


Sandro Ronaldo Bezerra Oliveira
Alexandre Marcos Lins de Vasconcelos

Abordagens para Apoio à Implementação da Melhoria do PROCESSO DE SOFTWARE


Editora
UFPE



Abordagens para Apoio à
Implementação da Melhoria do
PROCESSO DE SOFTWARE



Sandro Ronaldo Bezerra Oliveira
Alexandre Marcos Lins de Vasconcelos

Abordagens para Apoio à Implementação da Melhoria do PROCESSO DE SOFTWARE



Recife/2016

Catálogo na fonte:
Bibliotecária Joselly de Barros Gonçalves, CRB4-1748

A154 Abordagens para apoio à implementação da melhoria do processo de software [recurso eletrônico] / organizadores: Sandro Ronaldo Bezerra Oliveira, Alexandre Marcos Lins de Vasconcelos. – Recife : Editora UFPE, 2016.

Inclui referências bibliográficas.
ISBN 978-85-415-0754-7 (online)

1. Software – Desenvolvimento. 2. Engenharia de software. I. Oliveira, Sandro Ronaldo Bezerra (Org.). II. Vasconcelos, Alexandre Marcos Lins de (Org.).

005.1	CDD (23.ed.)	UFPE (BC2016-013)
-------	--------------	-------------------

Todos os direitos reservados aos organizadores: Proibida a reprodução total ou parcial, por qualquer meio ou processo, especialmente por sistemas gráficos, microfilmicos, fotográficos, reprográficos, fonográficos e videográficos. Vedada a memorização e/ou a recuperação total ou parcial em qualquer sistema de processamento de dados e a inclusão de qualquer parte da obra em qualquer programa juscibernético. Essas proibições aplicam-se também às características gráficas da obra e à sua editoração.



PRÓLOGO

O Objetivo e a Origem do Livro

O conteúdo deste livro aborda conceitos, teorias e, principalmente, as principais abordagens para a implementação de programas de melhoria do processo de software a partir do uso de ferramentas de software e demais mecanismos de apoio. Seu objetivo é apresentar e ilustrar como estas abordagens devem ser tratadas e usadas no cenário de desenvolvimento de software de modo a facilitar o caminho rumo à melhoria do processo e da qualidade de software. Às vezes, o livro pode atuar como ensino-aprendizagem, nunca como uma regra rígida, com as melhores práticas usadas para atingir um programa de melhoria organizacional.

Este livro é fruto dos resultados obtidos, na forma de artigos técnicos e científicos, entre os anos de 2009 a 2015 com as pesquisas e práticas realizadas pelo projeto SPIDER – *Software Process Improvement: DEvelopment and Research* (www.spider.ufpa.br), instituído no Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Pará (www.ufpa.br/pp-gcc), em parceria com o Centro de Informática da UFPE (www.cin.ufpe.br), contando com a participação de alunos de Graduação, Iniciação Científica e Pós-Graduação (Especialização, Mestrado e Doutorado) em Ciência da Computação e Sistemas de Informação, vinculados a estes centros de pesquisa. O projeto SPIDER tem o objetivo de: apresentar um levantamento de

abordagens com características adequadas para possibilitar a criação de produtos de trabalhos (artefatos) derivados dos resultados esperados descritos nos objetivos dos processos do modelo MPS.BR - Melhoria do Processo de Software Brasileiro, mantido pela SOFTEX - Associação para Promoção da Excelência do Software Brasileiro, e das práticas específicas descritas nos objetivos das áreas de processo do modelo CMMI – *Capability Maturity Model Integration*, mantido pelo CMMI Institute.

As pesquisas realizadas nestes centros de pesquisa foram acompanhadas pelos professores autores deste livro, Prof. Sandro Ronaldo Bezerra Oliveira (Professor Doutor do Programa de Pós-Graduação em Ciência da Computação da UFPA) e Prof. Alexandre Marcos Lins de Vasconcelos (Professor Doutor do Centro de Informática da UFPE). Os artigos, obrigatoriamente, tinham como meta apresentar os resultados da experiência da aplicação e/ou da utilização na indústria de software das abordagens tratadas neste livro. O livro é direcionado a alunos, gestores, implementadores, avaliadores e outros interessados no tema qualidade e melhoria do processo de software. Vale enfatizar que estes artigos foram revisados pelos membros pertencentes aos comitês das seguintes conferências e periódicos: SBQS, WAMPS, WER, WGPS, ENACOMP, SEA, CBSOFT, WSL, SBES, ERIN, Computer on the Beach, SEMINF, COBENGE, SBGames, JPC, Revista FSMA, ICSEA e Journal JSEA.

Por último, este livro não é autossuficiente. Ele exige que o leitor utilize as referências bibliográficas que apresentamos. Assim, se algo destacamos deste livro é que a bibliografia vale a pena por si só.

As Vertentes do Livro

O título deste livro mistura duas ideias poderosas. Fala de melhoria do processo de software e de abordagens que tratam sobre a implementação desta melhoria. Qualquer uma das duas ideias merece um livro independente, de

maneira que escrever só um, e no curto prazo com que contaram os autores, implica uma escolha de conteúdos. Este é um livro sobre as abordagens que são utilizadas em consultorias de melhoria de processos.

Não é um livro de consultoria, estes existem e são muito bons, escritos por consultores melhores que nós. No entanto, há muitos conselhos sobre como realizar as práticas importantes, as que levam a mudanças sérias, que estão contidas nos temas tratados. A escolha destes temas deu-se por se tratarem de abordagens técnicas que costumamos introduzir, de um modo ou de outro, em nossas consultorias na indústria de software e em ensino nas Universidades.

Embora favoreça a melhoria do processo organizacional, não é um livro sobre modelos de melhoria do processo, preferimos que o leitor aprenda sobre esta robusta área nos seus próprios guias e nos cursos autorizados que são oferecidos. No entanto, não há nada no livro que não tenha sido escrito com os conceitos de melhoria do processo em mente.

Relevância do Livro

Devido à globalização, cada vez mais os produtos e serviços estão sujeitos a novas exigências de mercado, à alta competitividade e à concorrência internacional. Neste cenário, a qualidade torna-se uma arma competitiva, tendo em vista a equiparação com padrões internacionais, conformidade com a especificação e a satisfação do cliente.

A qualidade pode ser definida como a totalidade das características de uma entidade que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas. Entre os benefícios da qualidade estão a redução de defeitos, o aumento da confiabilidade do produto, a redução do esforço de retrabalho, a redução de custo de desenvolvimento e manutenção, e um maior índice de satisfação dos clientes.

A literatura tem mostrado que a qualidade do produto ou serviço de uma organização só é plenamente obtida se os processos das organizações

aderirem a normas de qualidade. Visando a qualidade do processo, o modelo MPS.BR e o modelo CMMI propõem-se a serem guias evolucionários para o aperfeiçoamento e adequação de processos organizacionais.

De uma forma geral, o MPS.BR e o CMMI descrevem o quê deve ser feito para melhoria gradual de processos, definindo níveis de maturidade que são organizados por processos/áreas de processo que possuem objetivos alcançados por resultado esperados/práticas específicas os quais são evidenciados em produtos de trabalho. Para tanto, faz-se necessária a utilização de ferramentas de software para possibilitar a implantação dos modelos MPS.BR e CMMI nas organizações.

Atualmente, existe um movimento crescente onde esforços individuais e coletivos, impulsionados pela massificação da Internet, derivam a criação de softwares não proprietários com o objetivo de atender às necessidades comuns das organizações e de indivíduos. Estes softwares denominados software livre, devido a liberdade de qualquer um poder utilizá-los e modificá-los, estão cada vez mais difundidos nas organizações e no uso doméstico. Até governos, como o brasileiro, criam políticas de incentivo para difusão destes softwares.

Por definição o software livre (*free software*) deve estar disponível com a permissão para qualquer um usá-lo, copiá-lo e distribuí-lo, seja na forma original ou com modificações, seja gratuitamente ou com custo. A possibilidade de modificações implica que o código fonte esteja disponível (*open source*) e que o software possa ser executado em um sistema operacional também livre.

Assim, a relevância deste livro está no fato de apresentar alternativas viáveis com relação a ferramentas de software para auxiliar a implementação dos modelos MPS.BR e CMMI nas organizações, sem a necessidade de aquisição de softwares proprietários e com a possibilidade da ferramenta ser customizada para atender as especificidades da organização, podendo diminuir os custos e o tempo ao longo da implementação destes modelos de maturidade.

AUTORES

- **Sandro Ronaldo Bezerra Oliveira**

Mestre, Doutor e Pós-Doutor em Ciência da Computação pela UFPE. Professor Adjunto do Programa de Pós-Graduação em Ciência da Computação e da Faculdade de Computação da Universidade Federal do Pará. Implementador e Avaliador Líder MPS. Pesquisador-líder do Projeto SPIDER-UFPA.

- **Alexandre Marcos Lins de Vasconcelos**

Mestre em Ciência da Computação pela UFPE, Doutor em Ciência da Computação pela University of York e Pós-Doutor em Engenharia de Software pela Universidad Politécnica de Valencia. Professor Associado do Centro de Informática da Universidade Federal de Pernambuco. Implementador e Avaliador Líder MPS. Membro do Grupo de Pesquisa do Projeto SPIDER-UFPA.

COAUTORES:

Os créditos deste livro vão para todos os ex-alunos de Graduação, de Iniciação Científica, de Especialização, de Mestrado e de Doutorado que participaram ativamente em suas pesquisas do projeto SPIDER - *Software Process Improvement: DEvelopment and Research* da UFPA, a saber:

- Alexandre Brito Cardias Júnior
- Antônio André Cunha da Silva
- Arthur Ivson Xavier de Moraes Batista
- Bernardo José da Silva Estácio
- Bleno Wilson Franklin Vale da Silva
- Carlos dos Santos Portela
- Dan Jhonatan da Silva Teixeira
- Diego Oliveira Abreu
- Diogo Adriel Lima Ferreira
- Ehilton Yoshidome
- Elder de Jesus Ferreira da Silva
- Elói Luiz Favero
- Emanuel M. da C. Tavares
- Ewelton Yoshio Chiba Yoshidome

- Fabrício Medeiros Alho
- Fledison Barbosa
- Géssica Pinheiro da Silva
- Gleise Pinheiro Baldez
- Heresson João Pampolha de Siqueira Mendes
- Iuri Igonez Silva Raiol
- Júlio Cezar Costa Furtado
- Jñane Neiva Sampaio de Souza
- José Brício M. C. Neto
- Kleoson Bruno C. Santos
- Kleverton Oliveira Macedo
- Leonardo Possamai Mezzomo
- Luciana Neves Bentes
- Luiz Otávio Danin Lima
- Manoel Victor Rodrigues Leite
- Marcos Senna Benaion Leal
- Mariano Gomes
- Marília Paulo Teles
- Maurício Ronny de Almeida Souza
- Olavo Nylander Brito Neto
- Paulo Junior S. de Souza
- Paulo Robson Campelo Malcher
- Paulo Vitor R. Cardoso
- Pedro Afonso Aviz
- Rafael Oliveira Chaves
- Renan Sales Barros
- Rodrigo Araujo Barbalho
- Simone Nayara Costa Carneiro
- Sivaldinho de Oliveira Barbosa
- Tales C. Miranda
- Thiago Sylas Antunes da Costa

- Vitor de Souza Castro
- Wallace Michel Pinto Lira
- Walter A. da L. Lobato



Capítulo 1

Introdução

Sandro Ronaldo Bezerra Oliveira
Alexandre Marcos Lins de Vasconcelos

Devido à globalização, cada vez mais os produtos e serviços estão sujeitos a novas exigências de mercado, à alta competitividade e à concorrência internacional. Neste cenário, a qualidade torna-se uma arma competitiva, tendo em vista a equiparação com padrões internacionais, conformidade com a especificação e a satisfação do cliente.

A qualidade pode ser definida como a totalidade das características de uma entidade que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas [ABNT, 1994]. Entre os benefícios da qualidade estão a redução de defeitos, o aumento da confiabilidade do produto, a redução do esforço de retrabalho, a redução de custo de desenvolvimento e manutenção, e um maior índice de satisfação dos clientes.

A literatura tem mostrado que a qualidade do produto ou serviço de uma organização, só é plenamente obtida se os processos das organizações aderirem a modelos e normas de qualidade. Visando a qualidade do processo, o modelo MR-MPS – Modelo de Referência de Melhoria do Processo de Software [SOFTEX, 2012a] propõe-se a ser um guia evolucionário para o aperfeiçoamento e adequação de processos organizacionais.

Em 2003, a Qualidade tornou-se uma das prioridades da SOFTEX – Associação para Promoção da Excelência do Software Brasileiro, elencada

como um dos seus Projetos Estruturantes. Desde dezembro de 2003, sete renomadas instituições brasileiras, com competências complementares na melhoria de processos de software em empresas, participam do programa MPS.BR – Melhoria do Processo de Software Brasileiro: a Sociedade SOFTEX; três instituições de ensino, pesquisa e centros tecnológicos (COPPE/UFRJ, CESAR, CenPRA); uma sociedade de economia mista (CELEPAR); e duas organizações não-governamentais integrantes do Programa SOFTEX (RIOSOFT e Sociedade Núcleo SOFTEX 2000 de Campinas).

O programa MPS.BR visa a melhoria de processos de software em empresas brasileiras, a um custo acessível, especialmente na grande massa de micro, pequenas e médias empresas [Weber *et al.*, 2004a]. Não é objetivo do programa definir algo novo no que se refere a Normas e Modelos de Maturidade. A novidade do projeto está na estratégia adotada para sua implementação, criada para a realidade brasileira [Weber *et al.*, 2004a]. Além disto, o Modelo de Negócio definido para o programa tem grande potencial de replicabilidade no Brasil e em outros países de características semelhantes, como, por exemplo, os países latino-americanos. Desta forma, modelos, normas e métodos já disponíveis foram ponto de partida para a definição do Modelo de Referência, a análise da realidade das empresas brasileiras, a norma ISO/IEC 12207 [ISO/IEC, 1995, ISO/IEC, 2001, ISO/IEC, 2004b], a série de normas ISO/IEC 15504 (SPICE) [ISO/IEC, 2004a] e o modelo CMMI – *Capability Maturity Model Integration* [SEI, 2010a].

De uma forma geral, o MPS.BR descreve o quê deve ser feito para melhoria gradual de processos, definindo níveis de maturidade que são organizados por processos que possuem objetivos alcançados por resultados esperados, os quais são evidenciados em produtos de trabalho. Para tanto, de forma geral, faz-se necessária a utilização de ferramentas de software para possibilitar a implantação do programa MPS.BR nas organizações.

Neste contexto, atualmente, existe um movimento crescente onde esforços individuais e coletivos, impulsionados pela massificação da Internet, derivam a criação de softwares não proprietários com o objetivo de atender

às necessidades comuns das organizações e de indivíduos [GNU, 2015]. Estes softwares, denominados software livre, devido a liberdade de qualquer um poder utilizá-los e modificá-los, estão cada vez mais difundidos nas organizações e no uso doméstico. Até governos, como o brasileiro [GOV, 2015], criam políticas de incentivo para difusão destes softwares.

Por definição o software livre (*free software*) deve estar disponível com a permissão para qualquer um usá-lo, copiá-lo e distribuí-lo, seja na forma original ou com modificações, seja gratuitamente ou com custo [Hexsel, 2002]. A possibilidade de modificações implica que o código fonte esteja disponível (*open source*) e o software possa ser executado em um sistema operacional também livre.

Como melhor caracterização do cenário descrito anteriormente, podemos listar as seguintes justificativas que enfatizam a importância deste livro:

- a demanda por profissionais e empresas qualificadas tem levado a instalação de fábricas de software para exportação em diversos países do mundo, visando a redução de custos e, principalmente, a falta de mão-de-obra qualificada nos países de origem;
- diversas iniciativas do Governo e do Mercado marcam a transição para uma postura mais agressiva voltada à exportação de software: o próprio MPS.BR, apoiado pelo Governo; e as recentes políticas de investimento na qualificação e especialização de profissionais relacionadas com a Secretaria de Política de Informática – SEPIN;
- crescente número de acordos de terceirização envolvendo empresas nacionais e estrangeiras;
- notória capacidade das universidades em formar profissionais em quantidade e qualidade no setor de desenvolvimento de software, cita-se a grande quantidade de cursos de graduação e pós-graduação existentes no mercado nacional;
- de acordo com os relatórios de “Resultados de Desempenho das organizações que adotaram o modelo MPS” [SOFTEX, 2008b] e

de “Lições Aprendidas do MPS.BR” [SOFTEX, 2008a], e partir de implementações dos resultados esperados do MR-MPS em organizações, pelo Grupo do Projeto de Pesquisa SPIDER, a duração de uma implementação é reduzida substancialmente ao se adotar como prática ferramentas automatizadas, o que torna o processo de implementação mais ágil e menos custoso para as organizações. Adicionalmente, a partir do grande quantitativo de ferramentas de software livre de apoio às disciplinas da Engenharia de Software existentes na comunidade, pode-se caracterizar uma minimização dos custos ao longo do programa de melhoria da qualidade, quando desta adoção.

Neste contexto, este livro tem, como um dos focos principais, apresentar um levantamento das ferramentas de software livre com características adequadas para possibilitar a criação de produtos de trabalhos (artefatos que evidenciam a implementação do programa da qualidade organizacional) derivados dos resultados esperados descritos nos objetivos dos processos constantes em alguns níveis de maturidade do modelo MR-MPS-SW – Modelo de Referência de MPS para Software.

O objetivo deste livro visa apresentar alternativas viáveis com relação a ferramentas de software para auxiliar a implementação do modelo MR-MPS-SW e CMMI-DEV nas organizações, sem a necessidade de aquisição de softwares proprietários e com a possibilidade da ferramenta ser customizada para atender as especificidades da organização, diminuindo os custos e o tempo ao longo da implementação deste programa de maturidade.

Vale mencionar, ainda, os inúmeros resultados obtidos a partir das pesquisas em parcerias, que são consolidados com a elaboração deste livro, entre núcleos de renome na área de Engenharia de Software, como a UFPA e UFPE, através do projeto de pesquisa SPIDER – *Software Process Improvement: DEvelopment and Research* (www.spider.ufpa.br), com seus inúmeros artigos nacionais e internacionais publicados em conferências e periódicos,

sua obtenção em 2011 do 2º lugar no Prêmio Dorgival Brandão Júnior da Qualidade e Produtividade de Software (PBQP-SW) mantido pela Secretaria de Política de Informática do Ministério da Ciência, Tecnologia e Inovação, e outros tantos prêmios em trabalhos apresentados em eventos (WSL 2012, SBQS 2015, SBGames 2010), e sua formação de mais de 50 alunos em programas de mestrado e doutorado.

Assim, este livro está dirigido a:

- gerentes de projeto e de processo interessados em entender melhor algumas práticas para a melhoria do processo de software;
- responsáveis pela área de qualidade das organizações de software que queiram aprimorar seus conhecimentos sobre alguns dos modelos de qualidade para o processo e produto de software existentes na comunidade;
- responsáveis pela concepção, elaboração, construção e avaliação do processo de software organizacional que desejam melhor entender algumas técnicas e métodos existentes para apoiar nos seus trabalhos;
- engenheiros de software;
- professores de graduação e pós-graduação em Engenharia de Software;
- alunos de graduação e pós-graduação em Engenharia de Software.

Assim, nos próximos capítulos este livro fornecerá:

- os conceitos relacionados à melhoria de processos de software e nos modelos de qualidade MR-MPS-SW e CMMI-DEV (Parte I);
- conhecimentos importantes para a implementação dos modelos de qualidade, abordando os processos considerados para o nível de maturidade gerenciado (Parte II);
- abordagens para a implementação dos processos considerados para o nível de maturidade definido dos modelos de qualidade (Parte III);

- conhecimentos sobre algumas abordagens de apoio aplicadas indiretamente à implementação dos processos discutidos nas Partes II e III (Parte IV).



PARTE I

FUNDAMENTAÇÃO TEÓRICA

Sandro Ronaldo Bezerra Oliveira
Alexandre Marcos Lins de Vasconcelos

A definição, utilização e melhoria contínua de um processo de software correspondem aos principais objetivos de uma organização de software. Esse esforço geralmente considera apenas métodos e práticas da engenharia de software, sem contemplar suficientemente as restrições do ambiente de trabalho ou o conhecimento e a experiência das equipes de software. Ao definir um processo de software adequado a uma organização, é também importante ponderar as características peculiares da própria empresa e de seus grupos de trabalho. A alta rotatividade de pessoal, as equipes geograficamente dispersas, a falta de experiência sobre o domínio do conhecimento da aplicação não correspondem a dificuldades técnicas, mas podem determinar o sucesso ou o fracasso de um projeto de software.

Ainda neste contexto, os processos organizacionais do ciclo de vida são utilizados pelas organizações para estabelecer e implementar uma estrutura subjacente, constituída de processo de ciclo de vida e pessoal associados e para melhorar continuamente a estrutura e os processos. São empregados fora do domínio de projetos e contratos específicos.

Neste contexto, Humphrey (1989) define processo de software como o conjunto de tarefas de engenharia de software necessárias para transformar os requisitos dos usuários em software. Na definição de um processo

de software devem ser consideradas as seguintes informações: atividades a serem realizadas, recursos utilizados, artefatos consumidos e gerados, procedimentos adotados, paradigma e tecnologia adotados, e o modelo de ciclo de vida utilizado [Falbo, 1998; Pfleeger, 1998]. Travassos (1994) e Falbo (1998) apresentam alguns conceitos relacionados à definição de processos de software, a saber:

- **Atividades:** são as tarefas a serem realizadas. Uma atividade requer recursos e pode consumir ou produzir artefatos. Para sua realização, uma atividade pode adotar um procedimento. Uma atividade pode ser decomposta em outras atividades. Além disso, atividades, em qualquer nível, podem depender da finalização de outras atividades, denominadas pré-atividades;
- **Artefatos:** são produtos de software produzidos ou consumidos por atividades durante a sua realização. São exemplos de artefatos: manuais de qualidade, manuais de revisão, diagramas de fluxos de dados, diagramas de objetos, código fonte, etc. Um artefato pode ser decomposto em outros artefatos;
- **Procedimentos:** são condutas bem estabelecidas e ordenadas para a realização de atividades. Alguns procedimentos podem ser parcialmente automatizados por ferramentas de software. São exemplos de procedimentos: métodos de construção de sistemas, tal como o método de Booch para o desenvolvimento orientado a objetos; técnicas de avaliação da qualidade, tais como inspeções e *walkthroughs*; roteiros diversos para a produção de documentos; normas de programação; etc.;
- **Recursos:** são as pessoas, as ferramentas de software, os equipamentos, ou quaisquer outros recursos necessários à execução de uma atividade. Um recurso humano, especificamente, desempenha um papel na execução das atividades do processo;

- **Processos:** são coleções de atividades relacionadas que têm lugar durante o desenvolvimento de um produto.

Os processos de software podem apresentar grande complexidade e possibilitar diversas alternativas de execução de suas atividades. Desta forma, um processo de software definido permite que profissionais de engenharia de software possam trabalhar de forma ordenada, possibilitando um melhor entendimento do seu trabalho, bem como de outras atividades executadas por outros membros da mesma equipe [Humphrey, 1989].

No entanto, não existe um processo de software que possa ser genericamente aplicado a diversos projetos, visto que nenhum projeto é idêntico ao outro. Variações nas políticas e procedimentos organizacionais, métodos e estratégias de aquisição, tamanho e complexidade do projeto, requisitos e métodos de desenvolvimento do sistema, entre outros fatores, influenciam na forma como um produto de software é adquirido, desenvolvido, operado e mantido [ISO/IEC, 1995]. Assim, na definição de um processo deve-se considerar a sua adequação às tecnologias envolvidas, ao tipo de software em questão, ao domínio de aplicação, ao grau de maturidade (ou capacitação) da equipe em engenharia de software, às características próprias da organização, às características do projeto e da equipe [Humphrey, 1989; Rocha, 2001].

Assim, esta parte do livro fornecerá:

- conhecimentos sobre a melhoria contínua do processo (Capítulo 2);
- os conceitos relacionados ao modelo MR-MPS-SW (Capítulo 3);
- conhecimentos sobre o modelo CMMI-DEV (Capítulo 4).



Capítulo 2

Melhoria Contínua do Processo

Sandro Ronaldo Bezerra Oliveira
Alexandre Marcos Lins de Vasconcelos

Ao longo das últimas décadas, os produtos de software sofreram um considerável crescimento de tamanho e complexidade e o seu papel na vida moderna ganhou tamanha importância que hoje já são considerados vitais para diferentes ramos da sociedade.

Contudo, à medida que este crescimento ocorre, o número de problemas enfrentados durante o desenvolvimento também aumenta. Segundo Humphrey (1989), a produtividade das equipes diminui à medida que crescem o tamanho e a complexidade dos sistemas, e isso ocorre devido à baixa qualidade dos sub-produtos que compõem o produto final. Assim, problemas de orçamento e cronograma podem ser constatados na grande maioria dos projetos e o grande número de defeitos encontrados nos produtos liberados para o uso indica a necessidade da criação de técnicas que aumentem e garantam a qualidade dos sistemas. Além disso, com a competitividade entre as empresas produtoras de software, a qualidade passou a não ser mais um diferencial competitivo, mas, sim, um requisito básico para a sobrevivência no mercado.

Com o intuito de aperfeiçoar o desenvolvimento de software e obter produtos com os níveis desejáveis de qualidade, a última década assistiu a uma mudança de enfoque com relação ao processo de software [Rocha, 2001].

Tem-se, então, uma nova abordagem na qual o enfoque principal das atenções está na garantia da qualidade do próprio processo produtivo, visto que este tem se mostrado o fator determinante para o alcance da qualidade do produto final.

A partir dessa mudança de enfoque, intensificou-se a pesquisa sobre o processo de desenvolvimento e várias normas e modelos de qualidade foram definidos, dentre os quais vale destacar a Norma ISO/IEC 12207 [ISO/IEC, 1995], o CMM - *Capability Maturity Model* [Paulk, 1994], o SPICE - *Software Process Improvement and Capability dEtermination* hoje a norma ISO/IEC 15504 [ISO/IEC, 1998], o CMMI - *Capability Maturity Model Integrarion* [SEI, 2010a], e o MPS.BR - *Melhoria do Processo de Software Brasileiro* [SOFTEX, 2012a], para auxiliar na definição e melhoria de processos de software. Constatou-se, também, que para alcançar níveis cada vez mais altos de qualidade, era necessário melhorar cada etapa do ciclo de desenvolvimento. Porém, para que isso se tornasse possível, dados quantitativos, que pudessem descrever a realidade do processo, precisavam ser obtidos e devidamente analisados. Dessa forma, a realização de medições foi reconhecida como um pré-requisito indispensável para se introduzir a engenharia ao desenvolvimento, manutenção e uso de produtos de software [Basili, 1985].

Muitas métricas foram, então, propostas e aplicadas em casos práticos a fim de alcançar os seguintes objetivos [Park, 1996]:

- Melhorar o entendimento sobre o processo, produto, recursos e ambiente de desenvolvimento e, assim, estabelecer bases para a comparação entre as medições;
- Avaliar o andamento do projeto comparando com dados planejados;
- Fazer previsões sobre o futuro andamento do projeto com base em comportamentos passados;
- Promover melhorias identificando falhas, ineficiências e outras oportunidades para melhorar a qualidade do produto e o desenvolvimento do processo.

Entretanto, definir, coletar e analisar um conjunto de métricas não é uma tarefa trivial. Na realidade, esta é uma tarefa custosa que demanda grande conhecimento para evitar que o seu uso não aumente ainda mais os problemas enfrentados durante o desenvolvimento de software. Medições para serem efetivas, segundo [Basili, 1994], devem:

- Concentrar-se em objetivos específicos;
- Ser realizadas sobre todos os produtos, processos e recursos do ciclo de vida;
- Ter seus resultados interpretados com base em características do contexto organizacional e do ambiente.

Além disso, propor um conjunto de modificações para o processo ou para a organização a fim de melhorar os resultados obtidos é uma tarefa ainda mais desafiadora. Assim, Rocha (2011) propôs uma abordagem para medição e melhoria de processos de software que se baseia nas seguintes etapas:

- Seleção/definição de métricas adequadas, para realizar as medições, com base em objetivos previamente identificados;
- Realização de medições como parte integrante do processo de desenvolvimento de software;
- Análise dos resultados do uso do processo em projetos, apoiada por um sistema com base em conhecimento;
- Realização de estudos empíricos envolvendo medição de processos de software.

Vale enfatizar que essa abordagem supõe a definição prévia de processos de ciclo de vida de software na organização.

2.1. Definição das Métricas

A seleção e definição de um conjunto de métricas para medição do processo de software não é uma tarefa trivial. Muitas métricas foram propostas na literatura técnica e a decisão sobre qual métrica incluir ou excluir não é simples. Para esta abordagem, escolheu-se para uso um conjunto de métricas extraído/adaptado da literatura. No entanto, este conjunto, segundo Rocha (2001), está longe de ser o melhor conjunto de métricas, mas é suficiente para permitir a avaliação de processos de software de acordo com os objetivos estabelecidos nesta abordagem.

As métricas são selecionadas com três objetivos: caracterizar o projeto e seu contexto; medir, avaliar e sugerir melhorias em um processo de software específico; e realizar estudos empíricos envolvendo medição de processos de software. O conjunto selecionado foi organizado da seguinte forma:

- **Tempo:** tempo total do projeto; tempo nas fases de análise, projeto, codificação, teste de unidades feitos por analistas, testes do sistema e testes de homologação; tempo em reuniões de revisão; e tempo em retrabalho;
- **Precisão de estimativa de cronograma:** estimativa de cronograma de todo o projeto, estimativas de cronograma para as fases de análise, projeto, codificação, teste de unidades feitos por analistas, teste do sistema e teste para homologação;
- **Esforço:** esforço total do projeto; esforço nas fases de análise, projeto, codificação, teste de unidades feitos por analistas, testes do sistema e testes de homologação, esforço em reuniões de revisão; esforço em revisões; e esforço em retrabalho;
- **Precisão da estimativa de esforço:** estimativa de esforço para todo o projeto; estimativa de esforço para as fases de análise, projeto, codificação, teste de unidades feitos por analistas, teste do sistema, teste de homologação e revisões;

- **Tamanho do sistema:** número de linhas de código; pontos por função; pontos por casos de uso;
- **Número de erros:** número de erros na especificação de requisitos e no projeto do sistema encontrados em reuniões de revisão, erros no código encontrados nos testes de unidade feitos por analistas;
- **Número de modificações:** número de modificações na especificação de requisitos, projeto ou código após a sua aprovação;
- **Densidade de defeitos:** número de erros somado ao número de modificações em relação ao tamanho do sistema;
- **Rotatividade do pessoal:** percentual de pessoas que saíram, entraram ou mudaram de função durante o desenvolvimento do projeto;
- **Produtividade:** número de linhas de código produzidas por unidade de esforço; pontos por função ou casos de uso por esforço;
- **Deterioração do software:** relação entre o esforço gasto para corrigir os problemas encontrados após a liberação do sistema para o usuário comparado ao esforço gasto da liberação do software para o usuário;
- **Experiência da equipe:** experiência na linguagem de programação, no domínio da aplicação, nas ferramentas, no método e no processo de desenvolvimento, tipo de treinamento em engenharia de software e tempo total de experiência profissional.

As métricas cuja coleta tem como objetivo caracterizar o projeto e seu contexto de desenvolvimento visam permitir a comparação dos resultados obtidos em projetos semelhantes. As métricas, cujo objetivo de coleta é medir, avaliar e sugerir melhorias em um processo de software, visam inicialmente avaliar e melhorar a precisão das estimativas, avaliar e melhorar a qualidade dos produtos, e medir e diminuir o retrabalho. Finalmente, tem-se métricas a serem coletadas com o objetivo de formar uma base de dados para futuros estudos empíricos com o objetivo e aumentar o entendimento sobre os processos do ciclo de vida de software.

Um aspecto fundamental a ser considerado após a seleção das métricas é a sua definição operacional. Para produzir essa definição operacional é necessário ter conhecimento sobre o que um determinado dado representa e como este deve ser coletado de forma a se garantir que pessoas diferentes irão implementar as métricas de forma correta e consistente e que os dados serão interpretados corretamente [Rocha, 2001].

2.2. Coleta das Métricas

Dado que um dos objetivos dessa abordagem é não sobrecarregar a organização desenvolvedora com atividades extras, afastando-a de sua atividade principal que é o desenvolvimento do produto de software. Assim, os dados para cálculos das métricas e posterior análise, segundo esta abordagem, devem ser obtidos de documentos simples e geralmente produzidos pelas organizações ao longo do processo de desenvolvimento, como o relatório histórico do projeto e uma planilha de atividades [Rocha, 2001].

O relatório histórico do projeto contém o registro de todas as atividades do processo de desenvolvimento com suas datas de início e fim, as reuniões de avaliação realizadas e seus resultados, os testes realizados e sua duração e o tempo para a homologação. As planilhas de atividades contêm o registro de todas as atividades realizadas por cada membro da equipe de desenvolvimento e gerência do projeto.

2.3. Análise das Métricas Coletadas

A análise das métricas coletadas é realizada com dois objetivos: por um lado pretende-se analisar os resultados visando sugerir melhorias para o processo de software; por outro realizar estudos empíricos comparando as métricas obtidas de medições realizadas em diversos projetos [Rocha, 2001].

Para identificar os aspectos do processo que necessitam de melhoria é fundamental a existência de mecanismos para análise dos resultados obtidos por meio das medições. Nesta abordagem, essa etapa pode ser apoiada por um sistema com base em conhecimento construído (algum ambiente de desenvolvimento de software descrito na literatura especializada) a partir do conhecimento de especialistas com relação à interpretação dos resultados obtidos nas medições e da relação entre estes resultados e os aspectos do processo de software e do contexto de desenvolvimento de projetos. O resultado é um conjunto de recomendações para a melhoria do processo. Esse conjunto de recomendações pode ser definido a partir do levantamento com especialistas de possíveis problemas existentes no processo de software ou relacionados ao contexto em que é realizado o desenvolvimento quando são obtidos resultados não aceitáveis para as métricas. Sempre que for obtido um valor não aceitável para uma métrica, haverá alguma inferência no sistema com base no conhecimento sobre as características do processo.

2.4. Modelo IDEAL para Melhoria Contínua do Processo

As organizações estão reconhecendo cada vez mais a necessidade de orientação específica para a execução de suas tarefas quando adotam novas ferramentas, processos e métodos da engenharia de software. Muitos esforços de melhoria, incluindo a melhoria do processo do software, a gerência de risco contínua, ou a introdução de um novo ambiente de desenvolvimento, são muito complexos, e seus efeitos a longo prazo, requerem uma abordagem especializada e sistemática para controlar o ciclo de vida de adoção da tecnologia. O SEI – *Software Engineering Institute*, desenvolveu e refinou o modelo IDEAL – *Initiating, Diagnosing, Establishing, Acting & Learning* [Mcfeeley, 1996] para ajudar e satisfazer esta necessidade.

O modelo IDEAL, como concebido originalmente, era um modelo de ciclo de vida para a melhoria do processo de software baseada no CMM, e por esta razão o modelo usou termos da melhoria do processo.

O Modelo IDEAL fornece um guia para a melhoria contínua das características de um processo organizacional, ou seja, o modelo fornece um guia disciplinado de engenharia para a melhoria contínua, foca no gerenciamento do programa de melhoria e estabelece a base para uma estratégia de melhoria a longo prazo. Ele é composto de fases, atividades e princípios. A abordagem tem forte relacionamento com o TQM – *Total Quality Management*: apoio e suporte; treinamento; uso de dados para a tomada de decisões; envolvimento dos participantes; associado com os objetivos e necessidades do negócio; continuidade e envolvimento do cliente. O modelo consiste de cinco fases conforme descrito a seguir e apresentado na Figura 1 [Mcfeeley, 1996]:

- **Initiating:** esta é a fase onde a infraestrutura inicial da melhoria é estabelecida, os papéis e as responsabilidades para esta infraestrutura são inicialmente definidos, e os recursos iniciais são atribuídos. Esta fase foca no estímulo para a melhoria do processo de software, na definição do contexto e do patrocinador e no estabelecimento da infraestrutura inicial para suporte da melhoria;
- **Diagnosing:** o foco nesta fase é desenvolver um entendimento mais completo do trabalho de melhoria. Durante esta fase duas estratégias para a melhoria do processo da organização são definidas: o estado atual e o estado futuro desejado. Estes estados organizacionais são usados para desenvolver uma abordagem para a prática de melhoria do negócio. Assim, o objetivo é caracterizar o estado atual e o estado futuro desejado da organização, e desenvolver recomendações de como proceder nas fases subsequentes;
- **Establishing:** o objetivo desta fase é desenvolver um plano de trabalho detalhado. As prioridades de quais práticas organizacionais serão melhoradas são ajustadas para refletir as recomendações feitas durante a fase *Diagnosing*. Este passo tem a finalidade de colocar prioridades para as alterações, desenvolver uma estratégia para realização do trabalho, identificar recursos disponíveis, e

desenvolver um plano de implementação detalhado (onde as ações, marcos de referência e as responsabilidades são incorporadas em um plano de ação);

- **Acting:** as atividades desta fase servem para ajudar uma organização a implementar o trabalho realizado nas três fases anteriores (*Initiating*, *Diagnosing* e *Establishing*). Assim, o foco é criar a melhor solução que atenda as necessidades organizacionais identificadas, testar a solução criada através de um projeto piloto, modificar a solução para refletir o conhecimento, experiências e lições obtidas no projeto piloto, e implementar a solução em toda a organização;
- **Learning:** nesta fase a experiência obtida com execução do modelo IDEAL é revista para determinar se os esforços conseguiram atingir os objetivos pretendidos, e como a organização pode executar a mudança mais eficazmente e/ou eficientemente no futuro. Para isso, as lições são coletadas, analisadas e documentadas, as necessidades identificadas na fase *Initiating* são reexaminadas para ver se foram atendidas, e propostas de alterações para melhoria futura devem ser fornecidas.

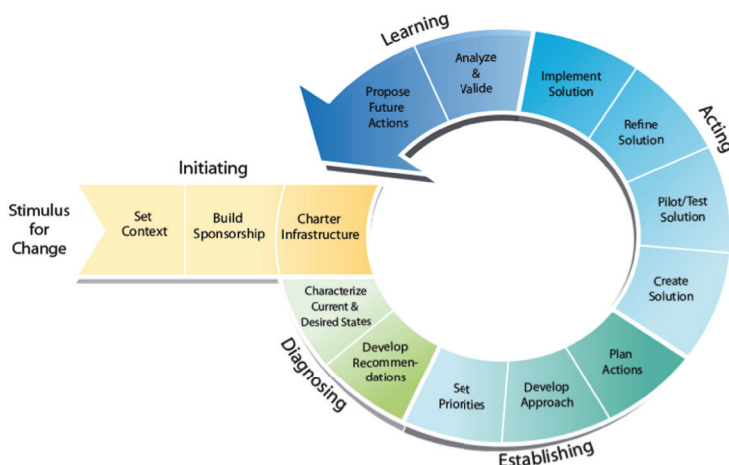


Figura 1. Estrutura do Modelo IDEAL [Mcfeeley, 1996]

Desta forma, pode-se notar que Programas de Melhoria da Qualidade Organizacional visam acompanhar o dia a dia da organização. Alguns envolvendo todas as funções e os níveis das organizações, objetivando a melhoria contínua da organização para que estas sejam capazes de satisfazer cada vez mais seus clientes.



Capítulo 3

O Modelo MR-MPS-SW

Sandro Ronaldo Bezerra Oliveira
Alexandre Marcos Lins de Vasconcelos

O MPS.BR é um programa mobilizador, de longo prazo, criado em dezembro de 2003, coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), que conta com apoio do Ministério da Ciência, Tecnologia e Inovação (MCTI), Financiadora de Estudos e Projetos (FINEP), Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE) e Banco Interamericano de Desenvolvimento (BID/FUMIN) [SOFTEX, 2012a].

O programa MPS.BR surgiu com o objetivo de buscar a Melhoria de Processo de Software e Serviços com duas metas a alcançar a médio e longo prazos: uma meta técnica e outra de negócio. A meta técnica visa a criação e melhoria do modelo MPS, e a meta de negócio visa disseminação do modelo, de modo que diversas organizações brasileiras, em todas as regiões do país, de pequeno, médio e grande porte, possam conhecer e adotar o modelo a um custo razoável, sejam estas públicas ou privadas [SOFTEX, 2012b]. O modelo vem ganhando repercussão não apenas no Brasil, mas também na América Latina, pois este modelo possui grande potencial de replicabilidade em outros países que possuem características similares no que se refere ao setor de software [Weber *et al.*, 2004a e 2004b].

O programa MPS.BR conta com uma unidade de execução do programa (UEP), responsável por executar o programa do modelo de qualidade

e duas estruturas de apoio para a execução de suas atividades: o Fórum de Credenciamento e Controle (FCC) e a Equipe Técnica do Modelo (ETM). O FCC emite parecer que subsidie decisão da SOFTEX sobre o credenciamento e monitoração de Instituições Implementadoras e Instituições Avaliadoras; podendo emitir parecer sobre descredenciamento se for o caso. O ETM apoia a SOFTEX sobre os aspectos técnicos, criação e aprimoração relacionados ao Modelo de Referência MPS para Software (MR-MPS-SW), Modelo de Referência MPS para Serviços (MR-MPS-SV) e Método de Avaliação (MA-MPS), para: além da capacitação de pessoas por meio de cursos, provas e workshops [SOFTEX, 2012b].

Através destas estruturas de apoio o MPS.BR pode contar com a participação de representantes de universidades, instituições governamentais, centros de pesquisa e de organizações privadas, os quais contribuem com suas visões complementares que agregam valor e qualidade ao programa [SOFTEX, 2012b]. A participação destes torna o processo mais robusto, pois agrega vários setores com sua visão de qualidade.

O MR-MPS-SW contém a definição dos níveis de maturidade, dos processos e dos atributos do processo relacionados a cada nível de maturidade. A base técnica para a construção e o aprimoramento do modelo é composta pelas normas ISO/IEC 12207 [ISO/IEC, 1995] e ISO/IEC 15504 [ISO/IEC, 2004a].

A iniciativa para elaboração deste modelo teve como objetivo aumentar a qualidade do processo das empresas de desenvolvimento de software, de forma que pequenas e médias empresas, possam melhorar seus processos com um custo inferior ao que seria necessário para implantar o CMMI-DEV, com possibilidades de conseguir incentivos financeiros do BID, BNDES e do SEBRAE para implantação do modelo MPS.BR. O recurso financeiro fornecido pelos Bancos visam o desenvolvimento do país e devem ser utilizados para obter a certificação, caso a empresa não alcance a certificação, o dinheiro deve ser devolvido. O SEBRAE aporta o recurso, mas a empresa tem que pagar 10% do valor. De modo que essas empresas, de forma acessível, possam

implantar seus modelos de qualidade e possuir certificado de qualidade em seus processos.

O progresso e o alcance de um determinado nível de maturidade do MR-MPS-SW obtêm-se quando são atendidos os resultados esperados dos processos e dos atributos de processos estabelecidos para aquele nível, como pode ser visto na Figura 2. Os atributos de processo (AP) são uma característica mensurável da capacidade do processo. Embora não seja detalhado no processo, para atender aos AP é necessário atender aos resultados esperados dos atributos do processo (RAP) para todos os processos correspondentes ao nível de maturidade [SOFTEX, 2012a].



Figura 2. Estrutura do MR-MPS-SW [SOFTEX 2012a]

A Figura 2 mostra a estrutura do modelo MR-MPS-SW, composta por níveis de maturidade, onde cada nível é uma combinação dos processos e da capacidade dos processos. Os processos são descritos segundo o propósito e os resultados esperados. O propósito descreve o objetivo a ser atingido com a execução do processo e os resultados esperados estabelecem os que devem ser obtidos com a efetiva implementação do processo. Já a capacidade do processo é representada por um conjunto de atributos descritos em termos de resultados esperados [SOFTEX, 2012a].

O níveis de maturidade do MR-MPS-SW são descritos na Tabela 1 conforme seus processos e atributos de processo. O modelo possui uma escala de maturidade que é dividida em sete níveis, começa no nível G e evolui até o nível A, quando a organização atinge a alta maturidade. Para cada processo ser atingido é necessário que os resultados esperados dos atributos do processo sejam evidenciados. Os atributos do processo do nível anterior são requeridos nos níveis posteriores. Então, para uma empresa estar no nível F é necessário possuir os atributos de processo do nível G e F. Os níveis que vão demandar maior esforço são os níveis G e F, pois requer uma adaptação no processo da empresa para atender os requisitos estabelecidos neste modelo.

Tabela 1. Níveis de Maturidade do MR-MPS-SW [SOFTEX, 2012a]

Nível de Maturidade	Processo	Atributos de Processo
A	-	AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Riscos – GRI	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Desenvolvimento para Reutilização – DRU	
	Gerência de Decisões – GDE	
D	Verificação – VER	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Validação – VAL	
	Projeto e Construção do Produto – PCP	
	Integração do Produto – ITP	
	Desenvolvimento de Requisitos – DRE	
E	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Gerência de Reutilização – GRU	
	Gerência de Recursos Humanos – GRH	
	Definição do Processo Organizacional – DFP	
	Avaliação e Melhoria do Processo Organizacional – AMP	

Nível de Maturidade	Processo	Atributos de Processo
F	Medição – MED	AP 1.1, AP 2.1 e AP 2.2
	Garantia da Qualidade – GQA	
	Gerência de Portfólio de Projetos – GPP	
	Gerência de Configuração – GCO	
	Aquisição – AQU	
G	Gerência de Requisitos – GRE	AP 1.1 e AP 2.1
	Gerência de Projetos – GPR	

O MR-MPS-SW [SOFTEX, 2012a] define sete níveis de maturidade, sequenciais e cumulativos, a seguir: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Gerenciado Parcialmente). Se comparado ao CMMI-DEV [SEI, 2010a], o modelo possui três níveis avaliáveis a mais, o que permite melhor atender às médias, pequenas e microempresas, que poderão alcançar os objetivos de melhoria em etapas intermediárias, fornecendo mais visibilidade do progresso [SOFTEX, 2012b]. Cada nível de maturidade possui atributos do processo a serem atingidos. Cada atributo do processo é requerido nos níveis do modelo e são descritos a seguir na Tabela 2.

Cada atributo do processo possui uma descrição e um propósito. O intuito da descrição é determinar o objetivo a ser atingido com a execução do processo e o propósito evidenciar os resultados esperados que devem ser obtidos com a implementação do processo.

Tabela 2. Atributos de Processo [SOFTEX, 2012a]

Atributo de Processo	Descrição	Propósito
AP 1.1	O processo é executado	Este atributo evidencia o quanto o propósito do processo é alcançado pela sua execução.
AP 2.1	O processo é gerenciado	Este atributo evidencia o quanto a execução do processo é gerenciada.
AP 2.2	Os produtos de trabalho do processo são gerenciados	Este atributo evidencia o quanto os produtos de trabalho produzidos pelo processo são gerenciados apropriadamente.

Atributo de Processo	Descrição	Propósito
AP 3.1	O processo é definido	Este atributo evidencia o quanto um processo padrão é mantido para apoiar a implementação do processo definido.
AP 3.2	O processo está implementado	Este atributo evidencia o quanto o processo padrão é efetivamente implementado como um processo definido para atingir seus resultados.
AP 4.1	O processo é medido	Este atributo evidencia o quanto os resultados de medição são usados para assegurar que a execução do processo atinge os seus objetivos de desempenho e apoia o alcance dos objetivos de negócio definidos.
AP 4.1	O processo é controlado	Este atributo evidencia o quanto o processo é controlado estatisticamente para produzir um processo estável, capaz e previsível dentro de limites estabelecidos.
AP 5.1	O processo é objeto de melhorias incrementais e inovações	Este atributo evidencia o quanto as mudanças no processo são identificadas a partir da análise de defeitos, problemas, causas comuns de variação do desempenho e da investigação de enfoques inovadores para a definição e implementação do processo.
AP 5.2	O processo é otimizado continuamente	Este atributo evidencia o quanto as mudanças na definição, gerência e desempenho do processo têm impacto efetivo para o alcance dos objetivos relevantes de melhoria do processo.



Capítulo 4

O Modelo CMMI-DEV

Sandro Ronaldo Bezerra Oliveira
Alexandre Marcos Lins de Vasconcelos

O CMMI-DEV [SEI 2010a] é um modelo de maturidade e capacidade de processos de software criado pelo SEI – *Software Engineering Institute* e que consiste de boas práticas de engenharia de software para o desenvolvimento e manutenção de produtos e serviços.

O CMMI possui cinco níveis de maturidade começa no nível 1 e evolui até o nível 5, quando a organização atinge a alta maturidade. Para cada área de processo ser atingida é necessário que os resultados esperados dos objetivos específicos e genéricos sejam evidenciados. As áreas de processo do nível anterior são requeridos nos níveis de posteriores. Então para uma empresa estar no nível 3 é necessário evidenciar as áreas de processo do nível 1 e 2, como pode ser visto na Tabela 3.

Tabela 3. Níveis de capacidade e de maturidade do CMMI-DEV [SEI 2010a]

Nível	Nível de Capacidade (representação contínua)	Nível de Maturidade (representação por estágios)
0	Incompleto	-
1	Realizado	Inicial
2	Gerenciado	Gerenciado
3	Definido	Definido
4	-	Gerenciado quantitativamente
5	-	Em otimização

O CMMI-DEV [SEI, 2010a] possui dois tipos de representação: contínua e por estágios. Na representação contínua, as áreas de processos são organizadas em categorias e a implementação da melhoria ocorre por níveis de capacidade, enquanto que na representação por estágios, as áreas de processos são organizadas em níveis de maturidade. Na primeira, as áreas de processos podem ser avaliadas individualmente, segundo a estratégia e objetivos de negócio da organização. Já na representação por estágios, a avaliação é realizada em todas as áreas de processos que compõem o nível de maturidade selecionado pela organização [SEI, 2010a].

O CMMI-DEV é formado por componentes agrupados em três categorias: componentes requeridos, componentes esperados e componentes informativos, que auxiliam na interpretação do modelo, conforme apresentado na Figura 3.

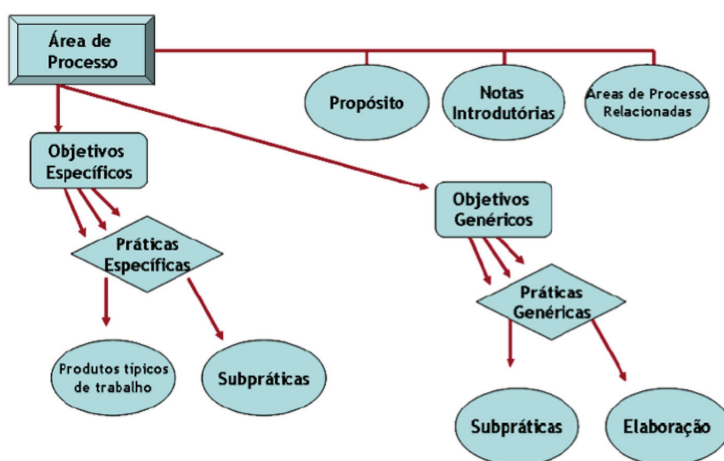


Figura 3. Componentes do CMMI-DEV [SEI, 2010a]

Cada área de processo possui objetivos específicos e objetivos genéricos a serem atingidos. Os objetivos específicos são atingidos a partir dos resultados esperados das práticas específicas e os objetivos genéricos são atingidos a partir dos resultados esperados das práticas genéricas.

O propósito, os objetivos específicos – relacionados ao processo – e os objetivos genéricos – relacionados a todos os processos e à organização – compõem as áreas de processo. A função dos objetivos específicos é definir as características únicas que devem estar presentes para que uma determinada área de processo seja satisfeita. Por sua vez, os objetivos genéricos estão associados a mais de uma área de processo e definem as características que devem estar presentes para institucionalizar os processos que implementam a área de processo [SEI, 2010a].

Em processos de avaliação da implementação do CMMI-DEV [SEI, 2010a], a equipe de avaliação visa buscar evidências de que os objetivos e práticas, específicos e genéricos, estão atendidos nos projetos, para as respectivas áreas de processos avaliadas.

O CMMI-DEV [SEI, 2010a] é composto por 22 áreas de processos, que podem ser observadas na Tabela 4, com os respectivos níveis de maturidade e categorias.

Tabela 4. Áreas de Processo do CMMI-DEV [SEI, 2010a]

Nível de Maturidade	Área de Processo	Categoria
2	Monitoração e Controle do Projeto (PMC)	Gerência de Projeto
2	Planejamento do Projeto (PP)	Gerência de Projeto
2	Gerência de Requisitos (REQM)	Gerência de Projeto
2	Análise e Medição (MA)	Apoio
2	Garantia da Qualidade do Processo e do Produto (PPQA)	Apoio
2	Gerência de Configuração (CM)	Apoio
2	Gerência de Contratos com Fornecedor (SAM)	Gerência de Projeto
3	Gerência de Projeto Integrada (IPM)	Gerência de Projeto
3	Gerência de Riscos (RSKM)	Gerência de Projeto
3	Definição do Processo Organizacional (OPD)	Gerência de Processo
3	Foco no Processo Organizacional (OPF)	Gerência de Processo
3	Treinamento Organizacional (OT)	Gerência de Processo
3	Desenvolvimento de Requisitos (RD)	Engenharia
3	Integração do Produto (PI)	Engenharia

Nível de Maturidade	Área de Processo	Categoria
3	Solução Técnica (TS)	Engenharia
3	Validação (VAL)	Engenharia
3	Verificação (VER)	Engenharia
3	Análise de Decisão e Resolução (DAR)	Apoio
4	Gerência Quantitativa do Projeto (QPM)	Gerência de Projeto
4	Desempenho do Processo Organizacional (OPP)	Gerência de Processo
5	Gerência do Desempenho Organizacional (OPM)	Gerência de Processo
5	Resolução e Análise Causal (CAR)	Apoio

4.1. Comparação do MR-MPS-SW com o CMMI-DEV

O MR-MPS-SW define níveis de maturidade que são uma combinação de processos e atributos de processo. Processos estão descritos através de seu propósito e resultados esperados do processo, enquanto que os atributos de processo estão descritos através de resultados de atributos de processos (RAP). Os componentes requeridos no MR-MPS-SW são os Resultados esperados de processos e os Resultados de atributos de processos.

O CMMI-DEV está organizado em áreas de processos, com objetivos e práticas específicos, e em objetivos e práticas genéricos. No CMMI-DEV, os componentes são agrupados em três categorias: requeridos, esperados e informativos. Em uma avaliação, os componentes requeridos e esperados do modelo CMMI-DEV têm o mesmo impacto dos resultados esperados do processo e dos resultados de atributos de processos do MR-MPS-SW, que obrigatoriamente devem estar atendidos pela organização avaliada.

A Figura 4 ilustra a árvore estrutural dos dois modelos, as similaridades entre os elementos avaliados nos níveis de maturidade do MR-MPS-SW e CMMI-DEV. Com relação à maturidade, os processos do MR-MPS-SW são compatíveis com as áreas de processo do CMMI-DEV. Os resultados

esperados dos processos do MR-MPS-SW, são comparados com as práticas específicas das áreas de processo do CMMI-DEV. Com relação à capacidade, os resultados de atributos de processos do MR-MPS-SW, são comparados com as práticas genéricas do CMMI-DEV.

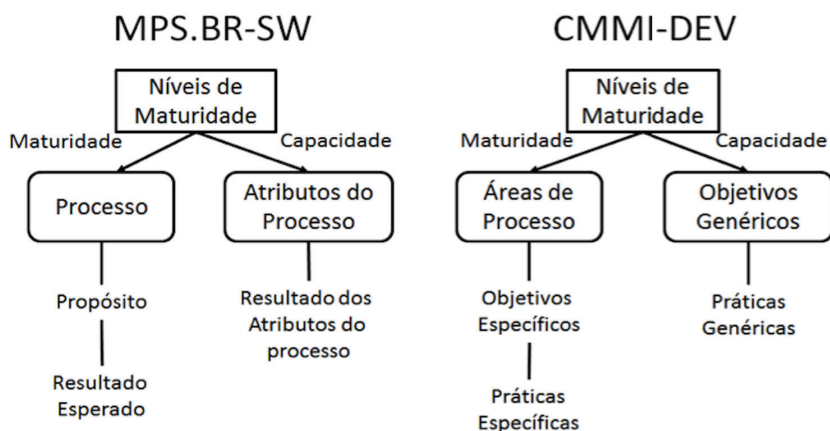


Figura 4. Comparação entre as Estruturas do MR-MPS-SW e CMMI-DEV

A Tabela 5 apresenta uma visão geral dos processos do MR-MPS-SW e seus correspondentes no modelo CMMI-DEV. A Gerência de Portfólio de Projetos, Gerência de Reutilização e Desenvolvimento para reutilização são processos que não existem no CMMI-DEV.

As áreas de processo do modelo CMMI-DEV que não têm um processo correspondente foram mapeadas por possuir correspondência com resultados de atributos de processos do modelo MPS, o que está refletido no mapeamento definido em [SOFTEX, 2012b].

No CMMI existe a representação por estágios e representação contínua. Segundo a documentação do MPS.BR, o modelo trabalha apenas com representação por estágios, podendo acontecer certificações por áreas de processo, nos casos onde a empresa estiver realizando algum nível de certificação. No MR-MPS-SW seria necessário implementar o nível inteiro e selecionar quais processos a empresa deseja adicionar. No CMMI é diferente, pois não é

necessário avaliar todo o nível, e sim apenas escolher quais áreas de processos seriam avaliados.

Tabela 5. Processos do MPS-SW e Áreas de Processos do CMMI-DEV [SOFTEX, 2012b]

Processos do MR-MPS-SW	Áreas de Processo do CMMI-DEV
Gerência de Projetos	Planejamento do Projeto
	Monitoração e Controle do Projeto
	Gerência Integrada de Projeto
	Gerência Quantitativa de Projeto
Gerência de Requisitos	Gerência de Requisitos
Aquisição	Gerência de Contratos com Fornecedor
Gerência de Configuração	Gerência de Configuração
Garantia da Qualidade	Garantia da Qualidade do Processo e do Produto
Gerência de Portfólio de Projetos	-
Medição	Medição e Análise
Avaliação e Melhoria do Processo Organizacional	Foco no Processo Organizacional
Definição do Processo Organizacional	Definição do Processo Organizacional
Gerência de Recursos Humanos	Treinamento Organizacional
Gerência de Reutilização	-
Desenvolvimento de Requisitos	Desenvolvimento de Requisitos
Integração do Produto	Integração do Produto
Projeto e Construção do Produto	Solução Técnica
Validação	Validação
Verificação	Verificação
Desenvolvimento para Reutilização	-
Gerência de Decisões	Análise e Tomada de Decisões
Gerência de Riscos	Gerência de Riscos
-	Análise e Resolução de Causas
-	Gerência de Desempenho Organizacional
-	Desempenho do Processo Organizacional

No MPS a empresa escolhe o nível a ser implementado, e se esta desejar pode determinar um ou mais processos de níveis posteriores. Então, esta pode receber a certificação do nível pretendido e do(s) processo(s) solicitados a mais.

O Quadro 1 expõe a compatibilidade entre os níveis de maturidade do MR-MPS-SW e do CMMI-DEV, que possibilita a migração do primeiro para o segundo, realizando adequações nos processos implantados. Resende (2009) apresenta um relato de experiência de uma empresa brasileira que implantou de forma simultânea o modelo MR-MPS-SW no nível F e o CMMI-DEV no nível 2. Segundo o autor, a migração entre os modelos ocorreu devido à compatibilidade já estabelecida e os recursos financeiros disponíveis.

Quadro 1. Mapeamento entre níveis de Maturidade MPS e CMMI [Resende, 2009]

Nível MPS	G	F	E	D	C	B	A
Nível CMMI	-	2	-	-	3	4	5

Para a realização das avaliações conjuntas e complementares do MR-MPS-SW e do CMMI-DEV foram geradas planilhas específicas (como visto na Figura 5), definidas a partir da planilha oficial de avaliação MPS, adaptadas para inclusão do componente correspondente do modelo CMMI-DEV, da classificação definida no mapeamento e das considerações relacionadas. Assim, foi mantida a mesma estrutura da planilha oficial MPS, com introdução de informações resultantes do mapeamento realizado [SOFTEX, 2012b].

	A	B	C	D	E	F	G	H	I	J	K	L
1	OPD	Definição do Processo										
2		PREENCHIDO PELA EMPRESA										
3	Práticas	Resultado esperado / evidências	Fonte da evidência	ORG	Projeto 1	Projeto 2	Projeto 3	Projeto 4	Projeto 5	Projeto 6	Projeto 7	Projeto 8
3		<p>O propósito do processo Definição do Processo Organizacional é estabelecer e manter um conjunto de ativos de processo organizacional e padrões do ambiente de trabalho usáveis e aplicáveis às necessidades de negócio da organização.</p> <p>DFP 1. Um conjunto definido de processos padrão é estabelecido e mantido, juntamente com a indicação da aplicabilidade de cada processo.</p> <p>As evidências apresentadas para este resultado permitem assegurar: (i) que os processos-padrão utilizados pela organização foram definidos, incluindo a identificação de quando são aplicáveis?; (ii) que estes processos-padrão e sua aplicação são atualizados, quando necessário?</p> <p>OPD SP 1.1 Estabelecer e manter o conjunto de processos padrão da organização.</p>	NEQ									
4	OPD SP 1.1											
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												

Figura 5. Planilha para Avaliação Conjunta MPS-SW/CMMI-DEV [SOFTEX, 2012b]

Nas áreas em destaque da planilha para avaliações conjuntas são apresentados: a área de processo (célula A1), a sigla e a declaração da prática específica do CMMI-DEV correspondente ao resultado esperado do processo no MR-MPS-SW (células A5 e A13 e células B5 e B13, respectivamente), o critério de classificação da comparação (células C5 e C13) e as considerações relacionadas, que foram incluídas na planilha por meio de comentários [SOFTEX, 2012b].

Inicialmente o modelo de qualidade é implementado, depois avaliado. Na implementação do modelo, em todos os processos que serão avaliados para o nível requerido, é necessário registrar as evidências de cada projeto na planilha, gerando assim um *hiperlink* para o(s) documento(s) que comprova(m) a evidência.

São escolhidos 4 projetos de acordo com o grau de relevância para organização, geralmente são 2 concluídos e 2 em andamento. O avaliador escolhe em conjunto com a organização.

O processo de avaliação consiste em uma avaliação inicial e outra final. Na avaliação inicial o avaliador fica em uma sala analisando as evidências registradas. A avaliação inicial no CMMI-DEV chama-se *headedness* e a final chama-se de avaliação oficial.

Na avaliação inicial estas evidências são verificadas. O avaliador marca a célula da planilha com uma das cores: verde (adequado), amarelo (precisa ser melhorado) ou vermelho (Não aceito). Todas as células marcadas com as cores: amarelo e vermelho são deixados como pontos requeridos. A empresa tem de 15 dias a 6 meses para ajustar os pontos nos projetos e corrigir todos os pontos fracos.

Na avaliação final o avaliador vai verificar se todos os pontos amarelos e vermelhos precisam ser revistos pois eles precisam tornar-se verdes, precisam estar adequados. Se na avaliação final houver uma célula vermelha, o processo de avaliação final é extinto, pois não pode existir pontos fracos na avaliação final. Caso todas as células estejam verdes, o avaliador deve definir o grau de implementação.

A avaliação final é realizada com entrevistas para evitar que as evidências tenham sido fraudadas. Todos os funcionários são reunidos para que os avaliadores os entrevistem sobre os processos da empresa, identificando assim que as evidências foram realmente produzidas pelo grupo. Se algo divergir na entrevista a avaliação pode ser impugnada.

O objetivo do avaliador nas entrevistas vai identificar a veracidade das evidências geradas e recomendar à alta administração o que o nível operacional está solicitando.

A implantação do modelo MR-MPS-SW ao ser comparado com o CMMI-DEV torna-se bem mais simples de ser implementado, pois o MPS possui mais níveis minimizando o esforço no processo inicial de implementação do modelo.

Na verdade o MPS.BR não é um novo modelo, o foco dele é fazer com que os modelos que já existem no mercado possam ser implementados no Brasil de maneira mais acessível de forma que as empresas nacionais

possam implementar. O MPS.BR é um programa para fazer com que os modelos que já existentes sejam de fato utilizados no Brasil a custo bem inferior.



PARTE II

A IMPLEMENTAÇÃO DOS PROCESSOS DO NÍVEL GERENCIADO

Sandro Ronaldo Bezerra Oliveira
Alexandre Marcos Lins de Vasconcelos

Segundo o SEI (2010a), no nível de maturidade Gerenciado, os projetos da organização têm a garantia de que os processos são planejados e executados de acordo com uma política; os projetos empregam pessoas experientes que possuem recursos adequados para produzir saídas controladas; envolvem partes interessadas relevantes; são monitorados, controlados e revisados; e são avaliados para verificar sua aderência em relação à descrição de processo. A disciplina de processo refletida pelo nível de maturidade Gerenciado contribui para que as práticas existentes sejam mantidas durante períodos de estresse. Quando essas práticas estão em vigor, os projetos são executados e gerenciados de acordo com seus planos documentados.

No nível de maturidade Gerenciado, o *status* dos produtos de trabalho e a entrega dos serviços estão visíveis para a gestão em pontos definidos (por exemplo, nos principais marcos e no término das principais tarefas). Os compromissos com as partes interessadas relevantes são estabelecidos e revisados conforme necessário. Os produtos de trabalho são controlados adequadamente. Os produtos de trabalho e serviços satisfazem às descrições de processo, aos padrões e procedimentos especificados.

Desta forma, esta parte do livro fornecerá:

- orientações de implementação dos processos de Gerência de Projetos (Capítulo 5) e Gerência de Requisitos (Capítulo 6);
- recomendações para a implementação dos processos de Gerência de Configuração (Capítulo 7), Garantia da Qualidade (Capítulo 8), Aquisição (Capítulo 9) e Medição (Capítulo 10).



Capítulo 5

Implementação do Processo de Gerência de Projetos

em [Yoshidome et al., 2010]

Ewelton Yoshio Chiba Yoshidome
Maurício Ronny de Almeida Souza
Wallace Michel Pinto Lira
Sandro Ronaldo Bezerra Oliveira

O objetivo do processo de Gerência de Projetos (GPR) é estabelecer e manter planos que definem as atividades, recursos e responsabilidades do projeto, bem como prover informações sobre o andamento do projeto que permitam a realização de correções quando houver desvios significativos no desempenho do projeto [SOFTEX, 2012a]. Gerir projetos inclui um conjunto de práticas gerenciais e técnicas que permite à equipe de software definir um roteiro, enquanto ela se move em direção à sua meta estratégica e seus objetivos táticos [Pressman, 2006]. Ainda, Gerência de Projetos é a disciplina de planejar, organizar e gerenciar recursos com o intuito de atingir os objetivos e metas do projeto [SEI, 2010a].

O Projeto SPIDER [Oliveira, 2011] agrega subprojetos que analisam a aderência de cada processo do MPS.BR às ferramentas de software livre existentes, adaptando-as ou desenvolvendo novas ferramentas conforme necessidade. No contexto deste trabalho, a ferramenta OpenProj, solução de software livre amplamente difundida para Gerência de Projeto, teve

funcionalidades adicionadas para suportar as exigências do MR-MPS-SW. Ainda, foi implementada a ferramenta Spider-CL com o objetivo de sistematizar a criação e aplicação de critérios objetivos nas análises promovidas pela organização.

Adicionalmente, foram desenvolvidas as ferramentas Spider-UCP e Spider-APF com o intuito de sistematizar a coleta e armazenamento de estimativas/métricas utilizando técnicas de Pontos Por Caso de Uso (*Use Case Points*) e Pontos por Função (*Function Points*). A versão customizada do OpenProj, a Spider-CL, a Spider-APF e a Spider-UCP encontram-se disponíveis em <http://www.spider.ufpa.br/>, acessando o item de menu Resultados.

5.1. Trabalhos Relacionados

Na linha de trabalhos apresentando propostas de implementação do MPS.BR a partir da utilização de ferramentas, Araújo e Mello (2009) propõe uma metodologia de uso da ferramenta Team Foundation Server para implementar o nível F do MR-MPS, (processo de gerência de projetos incluso), porém o artigo aborda a experiência de uso de uma ferramenta proprietária. Com foco em software livre, Cardias Junior *et al.* (2010), Estácio e Oliveira (2010) e Souza *et al.* (2010) apresentam estratégias de utilização de ferramentas de software livre para a implementação dos processos de Gerência de Requisitos, Medição e Gerência de Configuração, respectivamente, analisando, em seguida, a aderência das ferramentas elencadas com os resultados esperados do MR-MPS-SW.

Este trabalho aborda uma estratégia para a implementação do processo de Gerência de Projetos do MPS.BR baseada em boas práticas de uso de ferramentas de software livre, adaptadas e desenvolvidas no contexto de um projeto acadêmico, e analisa a aderência destas ferramentas semelhantemente aos trabalhos supracitados. Conjuntos de ferramentas para apoio à implementação MPS.BR também são vistos nas pesquisas [Rocha, 2005], que apresenta a

estação TABA – um ambiente de desenvolvimento de software, e [Lima *et al.*, 2006], sobre o WEbapsee, um ambiente de automação de processo de software. O presente trabalho analisa uma estratégia de utilização de ferramentas já existentes de software livre (Redmine e OpenProj) em conjunto com ferramentas desenvolvidas no Projeto SPIDER (Spider-CL, SPIDER-UCP e Spider-APF) para apoio à implementação do MR-MPS-SW.

5.2. Metodologia de Implementação do Processo de Gerência de Projetos com o Apoio de Ferramentas

Esta seção apresenta a metodologia proposta para implementação sistematizada do processo de GPR do MR-MPS-SW, através do uso das ferramentas de software livre: (1) OpenProj (versão customizada no Projeto SPIDER), para definição e planejamento do projeto; (2) Redmine (disponível em <http://www.redmine.org/>), para o acompanhamento e revisão; (3) o Subversion (ou SVN, disponível em <http://subversion.tigris.org/>), como repositório de dados do projeto; (4) Spider-CL, para definição e aplicação de critérios objetivos, a partir de checklists; e (5) Spider-APF ou Spider-UCP, para coleta e armazenamento de estimativas/métricas. Neste trabalho entende-se metodologia como sendo a codificação de um conjunto de práticas recomendadas, às vezes acompanhada de material de treinamento, programas de educação formal, planilhas, diagramas, tomando como parte um método (processo com uma série de passos, para construir um software) [Pressman, 2006]. Esta metodologia procura agregar boas práticas para o uso de ferramentas livres de apoio ao processo de GPR.

As ferramentas Spider-APF e Spider-UCP foram desenvolvidas pelo Projeto SPIDER. A primeira apresenta uma abordagem de coleta de estimativas/métricas a partir de Ponto por Função [Pressman, 2006], enquanto a segunda apresenta uma abordagem através de Pontos por Caso de Uso [Pressman, 2006]. A organização deve escolher a abordagem mais adequada

às suas especificidades. Para mais detalhes acerca do uso destas ferramentas, consultar <http://www.spider.ufpa.br/index.php?id=resultados>.

5.2.1. Planejamento do Projeto

Considerando um ambiente onde estejam devidamente instaladas e configuradas ferramentas de apoio ao processo de GPR, a metodologia inicia com a definição do Plano de Projeto. Este plano deve informar: (1) o escopo do projeto; (2) o dimensionamento dos produtos de trabalho; (3) a definição do modelo e das fases do ciclo de vida; (4) estimativas de esforço e custo; (5) definição do cronograma e orçamento; (6) análise dos riscos do projeto; (7) planejamento de recursos necessários; (8) identificação dos dados relevantes do projeto. Estas informações devem estar contidas no relatório gerado pela ferramenta OpenProj. É necessário, portanto, registrar estas informações na ferramenta, conforme visto na Figura 6.

Inicialmente, devem ser registrados os recursos disponíveis para o projeto. O OpenProj possibilita planejar os recursos, utilizando a funcionalidade “Recursos”, permitindo identificar recursos humanos ou de infraestrutura. Vale ressaltar que os recursos humanos envolvidos incluem todos os interessados no projeto, ou *stakeholders*.

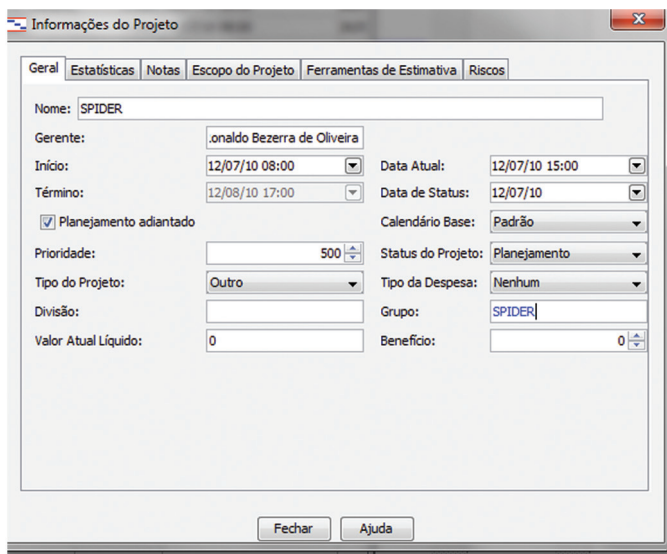


Figura 6. Informações do projeto: Descrição e detalhamento do projeto na ferramenta OpenProj

É necessário, posteriormente, registrar no OpenProj as tarefas que devem ser realizadas, alterando a tabela de tarefas em sua interface inicial. Recursos necessários para as tarefas devem ser alocados nesta etapa. No ato do registro das tarefas, informações importantes para cada tarefa são inseridas no Plano de Projeto, tais como: data de início; data de finalização; duração; e prioridade. Quando as tarefas forem registradas, a ferramenta OpenProj é capaz de gerar um diagrama de EAP (Estrutura Analítica do Projeto), o qual pode ser utilizado para visualizar o escopo da aplicação.

Para dimensionar as tarefas adequadamente, deve ser consultado dos projetos anteriores: (1) o histórico de Planos de Projeto gerados no OpenProj; (2) o acompanhamento de cada tarefa na ferramenta de Gerência de Mudança; e (3) o histórico de estimativas/métricas no Spider-APF/Spider-UCP. Com base nestas informações, as estimativas/métricas para o novo projeto devem ser registradas no Spider-APF/Spider-UCP ao passo que a duração das tarefas

deve ser feita no OpenProj, favorecendo, ainda, a definição do custo para realização de cada tarefa previamente especificada.

Após o registro dos recursos e as tarefas no OpenProj, deve ser preenchida a tabela referente à análise de riscos do projeto. Esta funcionalidade foi implementada pela equipe de desenvolvimento do Projeto SPIDER na ferramenta OpenProj. Para cada risco identificado, deve ser especificada uma descrição, probabilidade de ocorrência, impacto no projeto, as tarefas relacionadas ao risco e sua prioridade.

Finalmente, é necessário identificar os dados relevantes do projeto, que são as várias formas de documentação necessárias para a execução do mesmo, como atas de reuniões, relatórios, dados informais e demais artefatos importantes [SOFTEX, 2012a]. Para tal, deve ser utilizada a nova funcionalidade “Informações do Projeto”, implementada pela equipe do Projeto SPIDER, no guia “Dados Relevantes”. Devem ser informados o nome do artefato e a tarefa ao qual o mesmo está relacionado. O tratamento sobre estes dados importantes inclui definir formas de armazenamento, disponibilidade, responsáveis e nível de controle apropriado, tarefas normalmente planejadas pelo processo de Gerência de Configuração [Souza *et al.*, 2010]. A utilização de ferramentas de controle de versão e repositório, como o Subversion, contribui para a administração destes artefatos, de forma que todos os dados relevantes do projeto sejam colocados em um repositório e seu controle seja estabelecido através de metodologias de uso, definidas para a ferramenta durante o planejamento do projeto. Uma metodologia de uso da ferramenta Subversion é descrita em [Souza *et al.*, 2010].

Depois do registro destas informações, o Plano de Projeto está completo. No entanto, é necessário disponibilizá-lo. Para facilitar este procedimento, a equipe de desenvolvimento do Projeto SPIDER implementou a funcionalidade Gerar Relatório na ferramenta OpenProj. Esta funcionalidade gera um arquivo no formato PDF - *Portable Document Format* contendo os dados do Plano de Projeto.

5.2.2. Execução e Acompanhamento do Projeto

Antes de iniciar o projeto, o Plano de Projeto deve ser revisado. Em seguida, é necessário realizar um estudo de viabilidade considerando o escopo do projeto e aspectos técnicos, financeiros e humanos da organização. Estas revisões têm como objetivo verificar a viabilidade de inicialização do projeto. Em caso de aceitação, o Plano de Projeto deve, posteriormente, ser revisado com todos os *stakeholders* para assegurar seu comprometimento com o planejamento. Uma possível estratégia para documentar este comprometimento é divulgar o Plano de Projeto gerado na ferramenta OpenProj através de um fórum na ferramenta Redmine, evidenciando o comprometimento através da resposta dos interessados em mensagens no fórum. É importante observar que o Plano de Projeto deve ser previamente revisado em uma reunião com a equipe.

No contexto do processo de GPR, recomenda-se utilizar critérios objetivos para proceder a análise de viabilidade do projeto. Para tal, pode ser utilizada a ferramenta Spider-CL para registro de critérios objetivos, criação de *checklists* contendo os critérios definidos e posterior aplicação do *checklist*. Um *checklist* de Análise de viabilidade de projeto criado na ferramenta pode ser visualizado na Figura 7(b).

Durante a execução do projeto, este deve ser acompanhado de forma a garantir que todos os planos sejam seguidos, a aderência ao cronograma, analisar a resolução de cada tarefa e o esforço empregado. Para contemplar essas práticas, as tarefas definidas no cronograma da ferramenta OpenProj podem ser instanciadas através de *Issues* (registros na ferramenta Redmine para retratar tarefas a serem executadas ou problemas a serem resolvidos), e então acompanhar o que fora planejado com o que é realizado, como visualizado na Figura 7(a).

É necessário registrar, na ferramenta Redmine, os *stakeholders* (como usuários) e as tarefas (como *issues*), ambos previstos no Plano de Projeto. Durante a definição dos interessados, é possível atribuir papéis que irão

impactar nas permissões que os mesmos terão na ferramenta. Ao registrar *issues* de tarefas no Redmine é possível definir responsáveis, especificar datas, descrever passos da atividade, definir relações entre tarefas (ordem de execução ou dependência) e acompanhar a evolução das tarefas. Os *issues* contam com um ciclo de vida customizável na ferramenta, baseado em estados, e um histórico registrando qualquer atualização realizada. Os *issues* registrados podem ser visualizados em forma de gráfico de Gantt, atualizado automaticamente conforme o registro, execução ou término de tarefas. Para verificar se o planejamento está aderente à realidade, deve-se comparar o cronograma definido na ferramenta OpenProj (planejado) com o definido na ferramenta Redmine (instanciado). Finalmente, o registro de *stakeholders* como usuários, com sua consequente associação aos *issues* do Redmine, tem como objetivo gerenciar a atuação destes atores no projeto.

Tarefa #87 Update Log time Watch Duplicate

Elicitar Boas Práticas e Compor o Modelo de Processo

Added by Spider Admin about 9 hours ago. Updated about 9 hours ago.

Status:	Resolved	Start:	0
Priority:	Normal	Due date:	0
Assigned to:	Maurício Souza	% Done:	0
Category:	-	Spent time:	-
Target version:	-		

Description

Finalidade

- Apresentar o planejamento do projeto ao cliente.

Artefatos de Entrada:

- Cronograma Geral-CNG
- Descrição do Subprojeto-DSP
- Relatório de Aderência entre Programas de Melhoria-RAD

Artefatos de Saída:

- Descrição do Processo-DPR

Passos:

1. Elicitar as Boas Práticas que comporão o modelo de processo a partir do RAD;
2. Gerar a DPR a partir das boas práticas elicítadas no passo anterior;
3. Marcar a issue associada à tarefa como resolvida;
4. Atualizar DPR no SVN.

Subtasks

Related issues

blocks Gerência de Configuração - Tarefa #88: Validar Conjuntamente	In Progress	06 Au 2010
---	-------------	------------

Alteração dos Dados do Checklist

Dados do checklist:

Nome do checklist:

Descrição:

Lista de critérios do checklist:

Código do critério	Critério
80	O plano de projeto foi revisado, juntamente com...
81	Foi assumido o comprometimento entre os inter...
82	O modelo e as fases do ciclo de vida do projeto...
83	Foram identificados e documentados os riscos e...

Lista de usuário que podem acessar o checklist:

Código do usuário	Nome completo
-------------------	---------------

(a)
(b)

Figura 7. (a) Tarefa instanciada no Redmine. (b) Checklist para análise de viabilidade criado na Spider-CL

5.2.3. Revisar e Gerir Problemas

Ao longo do projeto, revisões devem ser feitas em marcos e conforme estabelecidas no Plano do Projeto, para verificar o andamento do projeto e gerar insumos para decisões relacionadas à viabilidade de continuidade do mesmo. Os marcos de projeto podem ser definidos como “Roadmaps” na ferramenta Redmine, e uma descrição pode ser associada a este marco explicitando informações pertinentes para a caracterização do marco. As revisões podem ser instanciadas como *issues* no Redmine, definindo no sistema um novo *tracker* (tipo de *issues*) “Revisão”, que pode ser relacionada a um marco específico. Os resultados das revisões devem ser adicionados no próprio *issue* criado, como notas ao atualizar seu estado para resolvido [Souza *et al.*, 2010].

Durante as revisões do projeto, problemas devem ser identificados, analisados e registrados para uma posterior ação corretiva e a devida gerência destas ações até a conclusão das mesmas. Estes problemas podem ser registrados como *issues* no Redmine para a devida gerência, e estes problemas registrados devem ser definidos como oriundos do *issue* de revisão ao qual estão relacionados.

Com base na análise dos problemas identificados, ações corretivas devem ser estabelecidas para a solução dos problemas que podem impedir que o projeto alcance seus objetivos, e estas ações devem ter sua execução monitorada até sua conclusão. O sistema de *issues* do Redmine fornece apoio a esta atividade, visto que as não-conformidades registradas têm um ciclo de vida que descreve a sua solução desde quando foi criada até ser concluída. À medida que o estado do *issue* sofre alterações, informações podem ser adicionadas, como: as ações corretivas; responsáveis; e relatos das atividades realizadas. Associado aos *issues*, o Redmine dispõe de históricos os quais permitem acompanhar a execução das atividades relacionadas aos problemas [Souza *et al.*, 2010].

5.3. Aderência da Metodologia Proposta ao MR-MPS

A análise de aderência entre a metodologia proposta ao modelo MR-MPS-SW é feita através do mapeamento das práticas sugeridas na Seção 5.2 com os resultados esperados do processo de Gerência de Projetos no Guia Geral do MR-MPS-SW [SOFTEX, 2012a], como visto na Tabela 6. É importante ressaltar que a metodologia restringe-se a apresentar boas práticas para que seja possível contemplar os resultados esperados do processo de GPR em conjunto com práticas definidas no processo de software da organização. Resultado Esperado (RE) é “um resultado observável do sucesso do alcance do propósito do processo” [ISO/IEC, 2004a]. No do processo de GPR do MR-MPS-SW (até o nível F), são 19 (dezenove) resultados esperados.

Tabela 6. Mapeamento da Metodologia com os Resultados Esperados de GPR do MPS
[Yoshidome *et al.*, 2010]

RE	Ferramentas	Funcionalidades/Práticas
GPR1	OpenProj	Diagrama EAP gerado automaticamente.
GPR2	OpenProj/ Spider-UCP/APF	Estimativas/Métricas registradas no Spider-UCP/APF; Duração das atividades registradas no OpenProj.
GPR3	OpenProj	Estrutura do Gráfico de Gantt gerado automaticamente.
GPR4	OpenProj	Consulta do histórico de Planos de Projeto.
GPR5	OpenProj/ Redmine	Registro de tarefas no OpenProj; Registro de <i>issues</i> do tipo Tarefa no Redmine.
GPR6	OpenProj	Tabela de riscos do projeto.
GPR7	OpenProj / Redmine	Registro de recursos humanos no OpenProj; Adição dos usuários (registrados como recursos no OpenProj) como membros do projeto no Redmine.
GPR8	OpenProj	Registro dos recursos de infra-estrutura.
GPR9	OpenProj / Redmine / Subversion	Tabela de dados relevantes do projeto no OpenProj; Acesso ao repositório no SVN pelo Redmine; Repositório de dados do projeto disponibilizado pelo SVN.
GPR10	OpenProj	Relatório gerado no formato <i>pdf</i> , contendo o plano do projeto.

RE	Ferramentas	Funcionalidades/Práticas
GPR11	OpenProj / Spider-CL	Revisão do relatório gerado; Aplicação do <i>checklist</i> pela Spider-CL.
GPR12	Redmine	Criar fórum para publicar plano do projeto e obter comprometimento dos interessados.
GPR13	Redmine	Instanciar e acompanhar tarefas planejadas no cronograma a partir de <i>issues</i> .
GPR14		
GPR15		
GPR16	OpenProj / Redmine	Registro de <i>stakeholders</i> no OpenProj; Registro de <i>stakeholders</i> no Redmine.
GPR17	Redmine	Registro de marcos como <i>Roadmaps</i> e revisões como <i>issues</i> .
GPR18	Redmine	Registrar problemas identificados nas revisões como <i>issues</i> , consequentes dos <i>issues</i> de revisão.
GPR19	Redmine	Registrar um plano de ação para a solução dos problemas no próprio <i>issue</i> , e acompanhar a sua resolução pelo ciclo de vida e histórico dos <i>issues</i> .

Vale enfatizar que a adequação das práticas propostas neste trabalho ao CMMI-DEV ocorre analisando os resultados esperados descritos na tabela acima com as compatibilidades descritas em [SOFTEX, 2012b] às práticas específicas deste modelo.



Capítulo 6

Implementação do Processo de Gerência de Requisitos

em [Cardias Junior et al., 2010]

Alexandre Brito Cardias Junior

Luciana Neves Bentes

Maurício Ronny de Almeida Souza

Ewelton Yoshio Chiba yoshidome

Sandro Ronaldo Bezerra Oliveira

A Gerência de Requisitos (GRE) é um conjunto de atividades que ajudam a equipe de projeto a identificar, controlar e rastrear requisitos e modificações de requisitos em qualquer época, à medida que o projeto prossegue [Pressman, 2006]. Essas características inerentes à GRE são melhor gerenciadas quando se automatiza/sistematiza o processo através da utilização de ferramentas, pois implicam na redução de esforço e tempo, devido a diminuição da necessidade de documentação agilizando o processo como um todo. A prática do uso de ferramentas para sistematizar/automatizar atividades de processo é um dos fatores que impacta diretamente na melhoria de processo de software [SOFTEX, 2008a].

O processo de Gerência de Requisitos no MR-MPS-SW tem como objetivo acompanhar a evolução dos requisitos e sua consistência com os outros produtos de trabalho da produção do software [SOFTEX, 2012a]. O processo de Gerência de Requisitos prevê a rastreabilidade bidirecional

entre requisitos e produtos de trabalho, que consiste em analisar impactos de possíveis mudanças.

O processo de Gerência de Requisitos não pretende coletar, desenvolver, detalhar requisitos, apenas acompanhar e administrar as inconsistências e mudanças que um requisito pode gerar, verificando até onde vai o impacto da mudança: planos do projeto, estimativas de tempo e custo, recursos humanos.

6.1. Ferramentas de Apoio

Para apoiar a implementação do processo de Gerência de Requisitos, foi utilizado um conjunto de ferramentas livres, de forma a alcançar uma metodologia sistematizada de implementação de um programa de melhoria da qualidade organizacional para atender aos resultados esperados do processo de GRE do MR-MPS-SW. Durante a pesquisa foi analisada a aderência de várias ferramentas aos resultados esperados do MR-MPS-SW, e destas foram escolhidas quatro ferramentas que melhor atendiam às necessidades da metodologia de implementação proposta: OSRMT, Spider-CL, DotProject e Mantis.

6.1.1. OSRMT

O OSRMT - *Open Source Requirements Management Tool* (isto é, ferramenta de código aberto para gerência de requisitos), disponível em <http://sourceforge.net/projects/osrmt>, é uma ferramenta desenvolvida na linguagem Java, projetada para apoiar o processo de gerência de requisitos e licenciada sob os termos da GPL (*General Public License*).

As principais características desta ferramenta focam em permitir uma completa rastreabilidade do ciclo de vida de desenvolvimento de software em relação aos requisitos.

Entre as funcionalidades da ferramenta, pode-se destacar: registro de autor, origem e motivo da necessidade de cada requisito; registro de casos de uso, *status* e origem de cada requisito (inclusive possibilitando atribuição de categorias aos requisitos); rastreabilidade (através de gráficos que identificam todas as dependências entre requisitos); definição e organização de artefatos e entrada de dados; e geração de relatórios padronizados em formato PDF.

6.1.2. Spider-CL

A Spider-CL, disponível em www.spider.ufpa.br, é uma ferramenta desenvolvida no projeto SPIDER da Universidade Federal do Pará, com propósito de criar *checklists* compostos por critérios objetivos para utilização em diversos contextos, provendo mecanismos para a aplicação destes *checklists*, mantendo histórico e registrando seus resultados.

Checklists são bastante utilizados para avaliações e inspeções objetivas de produtos de trabalhos diversos em organizações. Um *checklist* é uma lista de atributos ou qualidades que devem ser avaliados em um determinado produto de trabalho, onde cada um desses atributos possui uma lista de possíveis valores dos quais apenas um pode ser marcado. Um *checklist* nada mais é do que uma relação organizada de critérios objetivos.

A Spider-CL é uma ferramenta web, que pode ser executada através de servidor Tomcat, sendo acessível de qualquer navegador web, e seu banco de dados é estruturado em MySQL. Conta com serviço de controle de acesso através de cadastro de usuários e provê a sistematização do processo de definição e aplicação de *checklists* para avaliação, inspeção ou revisão através de critérios objetivos. A interface da Spider-CL foi desenvolvida utilizando componentes gráficos convencionais como caixas de textos, tabelas, listas e botões, para permitir fácil utilização.

A ferramenta Spider-CL é marcada pelas seguintes características:

- É uma ferramenta gratuita;
- É portátil, sendo desenvolvida como uma aplicação para o servidor Tomcat. A ferramenta pode ser executada em qualquer servidor capaz de executar o Tomcat 6.0 e o MySQL 5.1;
- Possui uma interface de fácil utilização;
- Pode ser utilizada para o desenvolvimento de qualquer tipo de *checklist* objetivo;
- Possui controle de acesso e mantém registro de todas as utilizações de cada *checklist*;
- Exporta os *checklists* preenchidos e seus resultados para o formato PDF.

6.1.3. DotProject

A DotProject, disponível em <http://www.dotproject.net>, é uma ferramenta desenvolvida em PHP com a finalidade de gerenciar projetos através de uma interface web. É software livre, com código aberto, distribuído sob a licença GPL, ou seja, seus usuários têm todo o poder de copiá-lo, modificá-lo ou distribuí-lo com ou sem modificações.

Sendo uma ferramenta de gerência de projetos, o DotProject é composto por funcionalidades para gerenciamento de tarefas, cronogramas, comunicação e compartilhamento.

6.1.4. Mantis

A Mantis, disponível em <http://www.mantisbt.org>, é uma ferramenta de *bugtracking*, sob licença GPL, desenvolvido para auxiliar o controle de modificações, no contexto do processo de gerência de configuração, através do gerenciamento das *issues*. *Issues* são relatos de problemas identificados nos

produtos de trabalho, que terão sua evolução acompanhada desde a solicitação da mudança até seu desfecho.

Por ser um software executado em *browser*, ele independe de sistema operacional e sua base de dados pode ser estruturada em MySQL, MS SQL e PostgreSQL.

Entre as principais funcionalidades desta ferramenta são identificados: criação de *issues*; gerenciamento do ciclo de vida das *issues*; registro do histórico das *issues*; e controle de *workflow* da ferramenta. Outros aspectos marcantes são: a possibilidade de customização; interface amigável, proporcionando fácil utilização; e a facilidade de extensão através de *plugins*.

6.2. Análise Avaliativa

A análise avaliativa das ferramentas propostas tem como objetivo tornar o processo Gerência de Requisitos do MR-MPS-SW totalmente implementado, de acordo com as exigências do modelo.

A análise isolada da ferramenta OSRMT não possibilitou essa implementação, devido restrições em suas funcionalidades quanto ao atendimento das recomendações do modelo MR-MPS-SW. Então, surgiu a necessidade de utilizá-la em conjunto com outras ferramentas para contemplar os resultados esperados.

A metodologia para o desenvolvimento da análise avaliativa iniciou-se com o estudo e entendimento do Guia Geral e Guia de Implementação do MPS.BR e posteriormente foi realizado o mapeamento dos resultados esperados de GRE com as funcionalidades das ferramentas. O resultado deste mapeamento pode ser visualizado na Tabela 7, que apresenta quais ferramentas são necessárias para a implementação de cada processo de GRE.

Tabela 7. Mapeamento dos Resultados Esperados com as Ferramentas Propostas [Cardias Junior *et al.*, 2010]

Resultados Esperados	OSRMT	Spider-CL	DotProject	Mantis
GRE1	Atende	Atende		
GRE2	Atende		Atende	
GRE3	Atende			
GRE4	Atende			Atende
GRE5				Atende

Vale enfatizar que adequação das práticas propostas neste trabalho ao CMMI-DEV ocorre analisando os resultados esperados descritos na tabela acima com as compatibilidades descritas em [SOFTEX, 2012b] às práticas específicas deste modelo.

O primeiro resultado esperado, GRE1, visa identificar um Fornecedor de Requisitos que é a pessoa responsável por fornecer e solicitar alterações em relação aos requisitos. Pretende-se evidenciar que o requisito foi Entendido, que não há discordância em relação ao que foi pedido; Avaliado, que o requisito não entra em conflito com os critérios da equipe técnica; e Aceito, de acordo com a expectativa do cliente; essas evidências devem ser apoiadas por critérios objetivos, ou seja, não podem mudar e devem ser claros.

Então na ferramenta OSRMT o Fornecedor de Requisitos poderá ser identificado e evidenciado na tela de usuário da ferramenta, como mostra a Figura 8.



Figura 8. Identificação do Fornecedor de Requisitos no OSRMT

Para a manutenção dos *status* de Entendido, Avaliado e Aceito, a ferramenta possibilita criar *status* novos para todos os requisitos que serão registrados. Antes de mudar o *status*, o fornecedor de requisitos deverá anexar o *checklist* aplicado na ferramenta Spider-CL, comprovando, dessa maneira, que o requisito foi avaliado segundo critérios objetivos.

O guia History na tela de Requisitos mostra a evolução de *status* que o requisito passou, como visto na Figura 9.

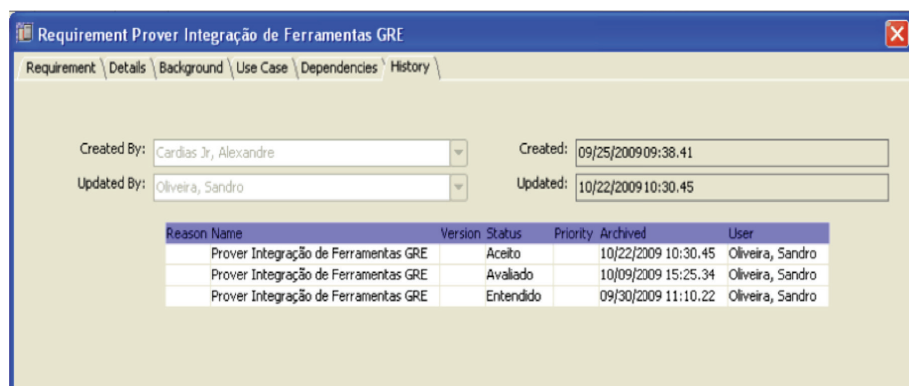
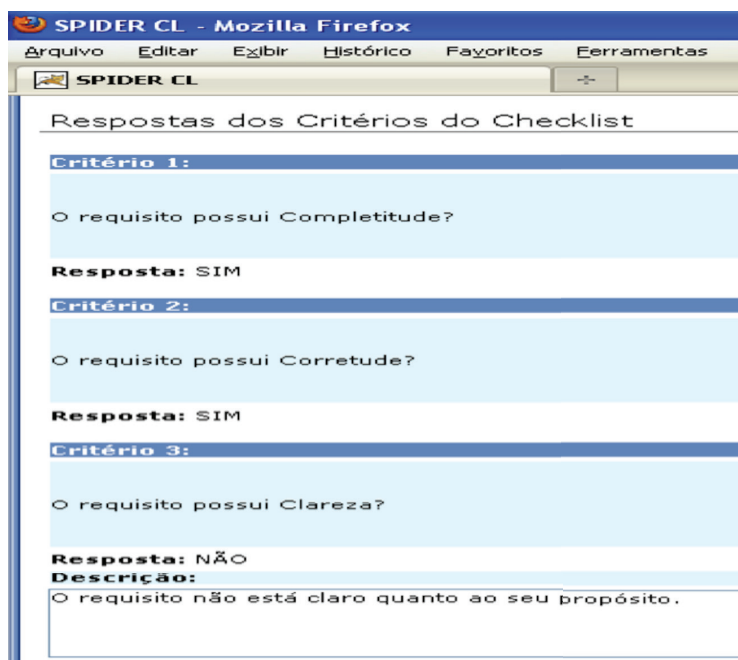


Figura 9. Histórico do Requisito no OSRMT

A ferramenta Spider-CL possibilita o registro e manutenção de critérios objetivos que serão adicionados no *checklist*, utilizado neste contexto para a avaliação dos requisitos. Possibilita, ainda, gerar um arquivo no formato PDF para evidenciar a aplicação dos critérios. Uma vez que o *checklist* foi definido, não será possível fazer modificações, atendendo dessa maneira a exigência do modelo MR-MPS-SW em se utilizar critérios objetivos, como visto na Figura 10.

O GRE2 trata sobre o comprometimento da equipe técnica com cada *status* referente ao GRE1. Assim, tanto quando o requisito é Entendido, Avaliado e Aceito o comprometimento deve ser obtido, assim quando uma mudança for solicitada e aprovada, devido o impacto que mudanças podem gerar nas estimativas, custos e cronograma do projeto.

Para atender o resultado esperado GRE2 o Gerente de Requisitos, Analista de Requisitos ou pessoa responsável, deve gerar um relatório na ferramenta OSRMT em que consta o nome, versão, *status*, prioridade e descrição dos requisitos a ser obtido o comprometimento. Este relatório servirá para a equipe técnica saber com quais requisitos estará se comprometendo.



The screenshot shows a web browser window titled "SPIDER CL - Mozilla Firefox". The address bar displays "SPIDER CL". The main content area is titled "Respostas dos Critérios do Checklist" and contains three criteria sections:

- Critério 1:**
O requisito possui Completitude?
Resposta: SIM
- Critério 2:**
O requisito possui Corretude?
Resposta: SIM
- Critério 3:**
O requisito possui Clareza?
Resposta: NÃO
Descrição:
O requisito não está claro quanto ao seu propósito.

Figura 10. Checklist Aplicado no Spider-CL

Como a ferramenta OSRMT não possui um mecanismo onde é possível registrar esse comprometimento, foi necessária a utilização da ferramenta DotProject, que possibilita a criação de fóruns. Parte-se do pressuposto que o projeto está registrado no DotProject, sendo necessário anexar o relatório gerado pela ferramenta OSRMT com os requisitos. Sendo assim, um fórum deve ser criado. Sugere-se o nome do fórum como sendo "Requisitos". Dentro desse fórum vários tópicos devem ser instanciados, para comprometimento de cada *status* (Entendido, Avaliado, Aceito), referenciando na mensagem do

tópico os identificadores dos requisitos funcionais registrados na ferramenta OSRMT, a ser obtido o comprometimento, como segue RF01, RF02, RF03. A sugestão para o nome do tópico tem o objetivo de facilitar a identificação dos arquivos e do *status* com o qual a equipe deve se comprometer. O nome do tópico pode seguir o formato:

[STATUS]#[DATA_CRIAÇÃO_TÓPICO]#[RELATORIO_DE_
ESPECIFICAÇÃO_DE_REQUISITOS]

Um exemplo para obtenção desse comprometimento e do formato do tópico do fórum, pode ser visualizado na Figura 11.

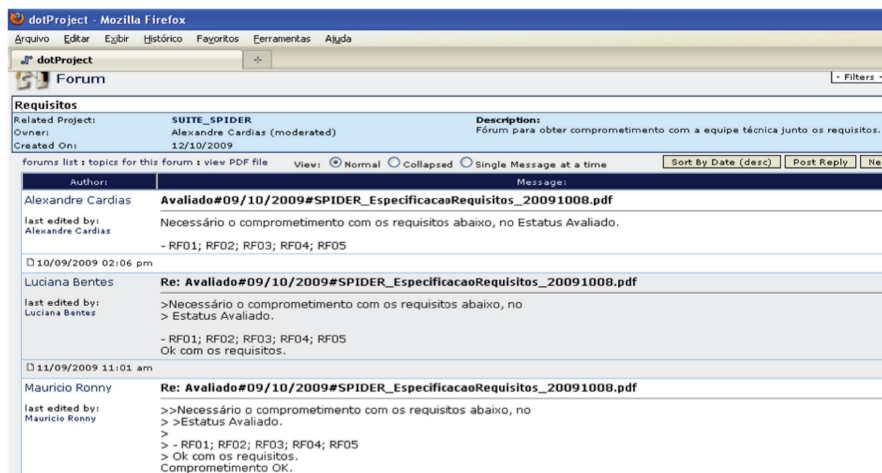


Figura 11. Comprometimento com a Equipe Técnica no DotProject

É no terceiro resultado esperado, GRE3, onde a rastreabilidade bidirecional vai ser estabelecida, que é um dos principais objetivos do processo de Gerência de Requisitos, servindo como base para a análise de impactos. Outro ponto fundamental nesse resultado esperado é a necessidade de manter a rastreabilidade para que seja sempre possível ter uma visão real da análise de impacto.

Este resultado esperado é alcançado através de algum mecanismo de rastreabilidade. A ferramenta OSRMT contempla de forma satisfatória este resultado através da Matriz de Rastreabilidade (horizontal ou vertical), como visto na Figura 12 e da Análise de impacto, importante mecanismo de visualização da(s) dependência(s) entre os artefatos, como visualizado na Figura 13.

No GRE4 faz-se necessária a realização de revisões em planos e produtos de trabalho do projeto, visando identificar e corrigir inconsistências em relação aos requisitos. Se inconsistências forem identificadas devem ser registradas [SOFTEX, 2012a].

Na ferramenta OSRMT as revisões serão registradas em *Feature*, funcionalidade que permite registrar características do projeto. Nessa funcionalidade deverá ser anexado o relatório gerado na ferramenta OSRMT contendo os requisitos. Para todas as revisões realizadas em relação a este relatório algumas informações referentes à data de realização da revisão, descrição e ocorrências (inconsistências), se ocorrer, deverão ser detalhadas, como mostra a Figura 14.

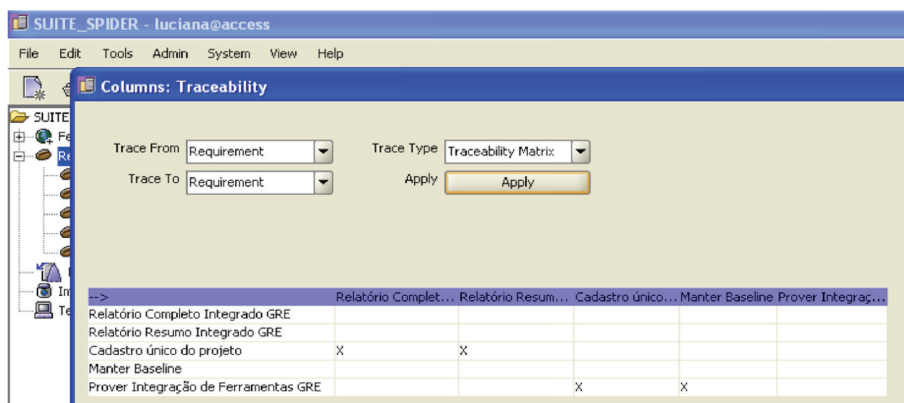


Figura 12. Matriz de Rastreabilidade no OSRMT

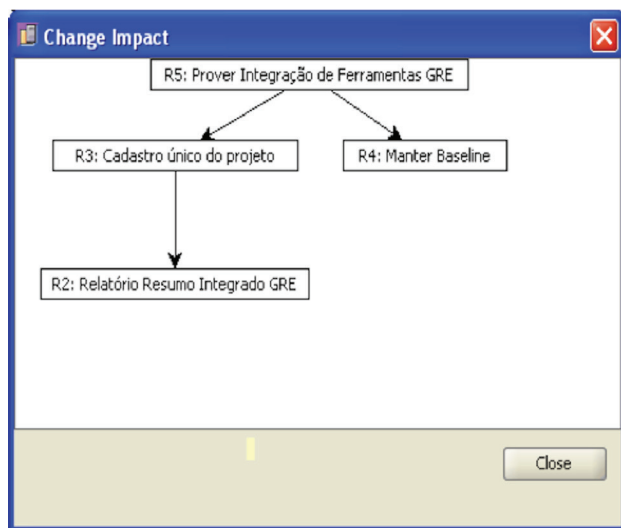


Figura 13. Análise de Impacto no OSRMT

Após a definição da revisão no OSRMT, as informações levantadas devem ser passadas para a ferramenta de controle de mudança, Mantis, a fim de realizar o devido tratamento e identificação dos problemas/melhorias.

Cada revisão constituirá de um conjunto de problemas/melhorias a ser tratado, e sendo assim cada revisão é registrada na ferramenta Mantis como uma *issue* (problema/melhoria), em que o campo “category” indicará que pertence ao processo de Gerência de Requisitos. Esta categoria deverá ser adicionada à ferramenta. A revisão registrada pode ser nomeada com um identificador no formato:

[IDENTIFICADOR_DA_REVISÃO]#[NOME_DO_DOCUMENTO_DE_REFERÊNCIA]

sendo um identificador único e de fácil reconhecimento.

Feature: SPIDER_EspecificReq_20091008.pdf

Version: Priority: Important

Feature Nbr: 1 Status: Submitted

Attachments:

Description	File name	Source
Especificação de Requisitos	SPIDER_EspecificReq_20091008.pdf	

Description:

```

Revisão: 001
Data: 08/10/2009
Descrição:
Ocorrências:
- Duplicidade de cadastro de usuário
- Mensagem de Confirmação de Alteração incorreta
- O id do requisito não está sendo replicado para as outras áreas de processo
- Histórico não atualiza
- Relatório Resumo não imprime
    
```

Figura 14. Revisões Registradas no OSRMT

No campo “summary”, onde é feita a descrição da *issue*, será informado o conteúdo da revisão, de forma idêntica a feita no OSRMT, contendo: número da revisão, data, descrição e as ocorrências, como mostra a Figura 15.

Viewing Issue Simple Details [[Jump to Notes](#)] [[Send a reminder](#)] [[Wiki](#)] [[cc](#)] [[View Advanced](#)] [[Issue History](#)] [[Print](#)]

ID	Category	Severity	Reproducibility	Date Submitted	Last Update
0000002	[SPIDER] Requisito	major	have not tried	2009-10-21 22:31	2009-10-21 23:41

Reporter: administrator View Status: public

Assigned To:

Priority: urgent Resolution: open

Status: resolved

Summary: 0000002: Revisao001SPIDER_EspecificReq20091008

Description:

```

Revisão: 001
Data: 08/10/2009
Descrição:
Ocorrências:
- Duplicidade de cadastro de usuário
- Mensagem de confirmação de alteração incorreta
- O ID do requisito não está sendo replicado para as outras áreas de processo
- Histórico não atualiza
- Relatório resumo não imprime
    
```

Additional Information:

Tags: No tags attached.

Attach Tags: (Separate by *) Existing tags

Attached Files:

Update Issue Assign To: [Myself] Change Status To: new Monitor Issue Create Clone Reopen Issue Move Issue Delete Issue

Figura 15. Identificação da Revisão no Mantis

Para cada ocorrência listada na revisão, deve ser criado um *issue* homônimo ao problema/melhoria, contendo descrição específica para aquela ocorrência, bem como a estratégia de resolução do problema. Os *issues* gerados a partir das ocorrências são conectados ao *issue* referente à sua revisão, através da funcionalidade de Relationships (relacionamento entre *issues*) do Mantis, de forma que o *issue* da revisão será “pai” dos *issues* de ocorrências, como pode ser visto na Figura 16.

Relationships				
New relationship			Current issue: parent of	<input type="text"/> <input type="button" value="Add"/>
parent of	0000003	resolved		Duplicidade de cadastro de usuário [Delete]
parent of	0000004	resolved	administrator	Mensagem de confirmação de alteração incorreta [Delete]
parent of	0000005	resolved	administrator	O ID de requisito não está sendo replicado para as outras áreas de processo [Delete]
parent of	0000006	resolved		Histórico não atualiza [Delete]
parent of	0000007	resolved		Relatório resumo não imprime [Delete]

Figura 16. Relacionamento entre *issues* de Ocorrências e *issue* de Revisão no Mantis

Esta metodologia de identificação dos problemas/melhorias encontrados nas revisões, permite satisfazer totalmente a implementação do GRE 4, e permite monitorar o avanço da resolução dos *issues*, rastreando-as a partir de suas respectivas revisões. O acompanhamento dos *issues* pode ser realizado no Mantis a partir do histórico dos *issues*.

O quinto resultado esperado, GRE5, tem como objetivo gerenciar as mudanças ocorridas ao longo do projeto. É importante para o gerenciamento, que um histórico seja mantido em relação aos requisitos, evidenciando dessa maneira a evolução das mudanças.

O resultado esperado GRE 5, que recomenda que mudanças nos requisitos são gerenciadas ao longo do projeto, requer identificar e acompanhar as mudanças nos requisitos, o que envolve controlar a evolução da resolução de ocorrências identificadas no resultado esperado anterior.

Os *issues* gerados na ferramenta Mantis, contam com um ciclo de vida próprio, baseado em uma série de estados em que um *issue* pode se enquadrar até ser fechada. Os estados são alterados conforme ações são tomadas para resolver o problema/melhoria. Os estados padrões de um *issue* no Mantis são:

new (novo), *assigned* (apontado para algum responsável), *resolved* (o *issue* foi provisoriamente resolvido) e *closed* (definitivamente resolvido).

Com base nos estados dos *issues*, pode-se definir um histórico de alterações que a ferramenta Mantis disponibiliza para cada *issue*. Além de apresentar estados no histórico, a ferramenta também exibe os detalhes sobre responsáveis (quem criou, a quem foi apontado), exibe notas que podem ser adicionadas explicando ações tomadas na resolução do problema/melhoria, datas de cada modificação no *issue*.

A metodologia aqui definida utiliza os históricos do Mantis como evidência do acompanhamento das ocorrências de problemas/melhorias identificadas nas revisões, como visualizado na Figura 17.

Issue History			
Date Modified	Username	Field	Change
2009-10-21 22:36	administrator	New Issue	
2009-10-21 22:43	administrator	Relationship added	child of 0000002
2009-10-21 23:38	administrator	Note Added: 0000006	
2009-10-21 23:38	administrator	Assigned To	=> administrator
2009-10-21 23:38	administrator	Status	new => confirmed
2009-10-21 23:39	administrator	Note Added: 0000007	
2009-10-21 23:39	administrator	Status	confirmed => resolved
2009-10-21 23:39	administrator	Resolution	open => fixed

Figura 17. Histórico de Mudanças dos *issues* no Mantis

Para facilitar o acompanhamento dos *issues*, recomenda-se o uso do campo “add note” para comunicar os procedimentos feitos para solucionar o problema, como visualizado na Figura 17.



Capítulo 7

Implementação do Processo de Gerência de Configuração

em [Souza et al., 2010]

Maurício Ronny de Almeida Souza
Wallace Michel Pinto Lira
Sandro Ronaldo Bezerra Oliveira

O processo de Gerência de Configuração (GCO) é responsável por aplicar procedimentos técnicos e administrativos para identificar e documentar as características físicas e funcionais de IC (Itens de Configuração), controlar suas alterações, armazenar e relatar o processamento das modificações e verificar a compatibilidade com os requisitos especificados, garantindo que tenha sido feito o que deveria ter sido feito [IEEE, 1990]. A Gerência de Configuração de Software é uma atividade de apoio que se desenvolve ao longo de todo o processo de software, comportando as ações de: identificar e controlar mudanças; garantir que as modificações sejam implementadas de forma adequada; e comunicar as modificações a todos os interessados [Pressman, 2006]. O foco é definir um sistema capaz de gerir produtos de trabalho eficientemente.

A metodologia proposta para implementação sistematizada do processo de GCO do MR-MPS-SW, fez uso das ferramentas livres: (1) Subversion (ou SVN, disponível em <http://subversion.tigris.org/>), para o controle de versão; (2) Redmine (disponível em <http://www.redmine.org/>), para a gestão de

mudanças; e (3) Spider-CL (disponível em <http://www.spider.ufpa.br>), para a definição e aplicação de critérios objetivos. Esta metodologia procura agregar boas práticas para o uso de ferramentas livres de apoio ao processo de GCO.

7.1. Definindo um Sistema de Gerência de Configuração

Considerando um ambiente onde estejam devidamente instaladas e configuradas as ferramentas de apoio ao processo de GCO, a metodologia inicia com o estabelecimento do sistema de GCO, ou seja, com a definição do meio de armazenamento, procedimentos e ferramentas para o registro e o acesso aos produtos de trabalho e suas solicitações de mudanças. Primeiramente, deve ser estabelecido o repositório para o projeto de acordo com as estratégias definidas pela organização, por intermédio de ferramenta de controle de versão, o SVN. Neste momento, também é estabelecido o banco de dados de solicitações de mudanças, isto é, os projetos são instanciados nas ferramentas de controle de mudança para o devido controle sistematizado. O repositório do projeto deve ser criado na ferramenta Subversion com a estrutura *trunk*, *branches* e *tags*, a qual: na pasta *trunk*, será o local de armazenamento da principal linha evolutiva dos itens de configuração; em *branches*, serão armazenados os ramos do projeto; e na pasta *tags*, serão guardadas as *baselines* (linhas base) do projeto.

Então, o projeto deve ser criado no Redmine e, durante a criação, devem ser definidos: nome do projeto; conexão com um banco de dados; tipo de repositório (SVN, neste caso); e caminho para o repositório. Com isto, será criado o sistema de controle de mudança já integrado ao sistema de controle de versão. Com os sistemas de controle de versão e de mudança devidamente estabelecidos para o projeto, deve ser iniciado um Plano de Gerência de Configuração, para a descrição dos procedimentos referentes a este processo através destas ferramentas. O plano é definido, nesta metodologia, em páginas *wiki* integradas ao Redmine, e nele devem ser descritos: os

responsáveis ou envolvidos; a identificação dos itens de configuração; planejamento e regras para criação de *baselines* e ramificações; a estrutura do repositório; as ferramentas utilizadas no processo de GCO e suas metodologias e políticas de uso; a descrição do processo de GCO; e o planejamento das auditorias de configuração.

7.2. Identificando Itens de Configuração

Tendo um sistema definido, é necessário definir os itens de configuração. Critérios objetivos devem ser elaborados para condução da identificação dos itens de configuração. Para avaliação dos itens de configuração, pode ser usada a ferramenta Spider-CL, a qual fornece serviços referentes à geração de *checklists*. Os resultados obtidos devem ser anexados ao Plano de Gerência de Configuração, encontrado na página *wiki* da ferramenta Redmine. Os itens de configuração definidos devem ser listados no Plano de Gerência de Configuração, além de ter informações sobre atributos como o nome, os responsáveis, a localização e o conteúdo. A regra de padronização da nomenclatura dos itens de configuração também deve ser incluída no Plano de Gerência de Configuração, juntamente com as regras de versionamento dos itens de configuração.

7.3. Manipulando e Controlando a Evolução dos Itens de Configuração

Os itens de configuração identificados são os artefatos ou produtos de trabalho que serão passíveis de GCO, logo, são os arquivos que deverão ser armazenados no repositório e, a partir de então, serem gerenciados quanto a sua versão e suas mudanças. O processo de registro e remoção de produtos de trabalhos do repositório é feito por intermédio de mecanismos de *check-in* e

check-out dos sistemas de controle de versão: *Check-in* refere-se ao ato de publicar alterações feitas em um produto de trabalho, enviando-as ao repositório para torná-las acessíveis a todos; *Check-out* é o ato de criação de uma cópia local de trabalho de um determinado item, a partir do repositório, ou seja, é a leitura de um item do repositório [Sussman, 2009].

Cada item armazenado nos sistemas de controle de versão passa a contar com um número de versão, ou revisão, automaticamente, e sempre que sofrer alguma alteração e for submetido (operação de *check-in*) ao repositório, este número é incrementado, permitindo que a equipe possa identificar e acessar os itens mais atuais do projeto. Além de controlar o número de revisão dos itens de configuração, toda a ação que envolva alteração no repositório registra: o usuário que fez a ação; o tipo de ação realizada (alteração, criação, remoção de elemento do repositório); data e hora; e permite a adição de um comentário pelo usuário.

Estes elementos constituem um histórico dos itens de configuração, armazenado na ferramenta. Os comentários adicionados durante operações de *check-in* são elementos imprescindíveis desta metodologia. Comentários padronizados devem ser institucionalizados dentro das organizações, de modo a prover uma leitura mais precisa do histórico do repositório. Detalhes importantes a serem colocados nestes são: o caráter da modificação; alguma explicação mais precisa sobre a modificação (qual seção de um documento foi alterada, por exemplo); e o motivo da alteração. Também é interessante colocar uma referência ao sistema de controle de mudança, o que será discutido posteriormente.

7.3.1. Gerando *Baselines*

Em algum momento do projeto, haverá necessidade de eleger configurações ao estado de *baselines*, e estas devem ser planejadas. Portanto, deve ser definido, no Plano de Gerência de Configuração, o padrão de identificação das

baselines, quando serão geradas, critérios de validação e questões relacionadas às auditorias sobre essas. O processo de criação de uma *baseline* é feito através da ferramenta SVN. Uma vez definida a configuração que será promovida a *baseline*, nela deverá ser aplicado um rótulo a partir da funcionalidade de *tags* do SVN (*svn copy*), que poderá ser recuperado a qualquer momento através de um *check-out*. Os rótulos criados para o projeto devem ser armazenados na pasta “*tags*” estabelecida no repositório e deve-se definir um padrão para eles, como, por exemplo: **projeto#release_x.y#aaaammdd**. O Redmine permite que versões do projeto sejam registradas no sistema, através da opção “*versions*” nas preferências do projeto, possibilitando que seja feito um acompanhamento mais preciso das mudanças sobre as *baselines*, liberações ou versões do projeto.

7.3.2. Acompanhando e Registrando a Situação dos Itens de Configuração e *Baselines*

Conforme os itens de configuração evoluam e as *baselines* forem definidas, surge a necessidade de identificar, diferenciar e recuperar o conteúdo de itens de configuração em diferentes etapas do projeto, ou seja, é necessário assegurar que todos os interessados tenham acesso e conhecimento sobre o histórico e situação específica de itens de configuração ou *baselines* ao longo do tempo.

Os históricos do SVN fornecem informações suficientes para um mapeamento preciso dos componentes de uma determinada *baseline*, apresentando a versão específica de cada item de configuração e permitindo a recuperação desta configuração. Com essas informações é possível fazer comparações entre *releases*, *baselines* ou versões do projeto, contabilizando o que foi feito ao longo do projeto. Aliado ao sistema de controle de mudanças é possível um controle ainda maior sobre o andamento do projeto.

O acompanhamento e o registro da situação dos itens de configuração são obtidos a partir da interpretação das informações dos relatórios fornecidos

pela funcionalidade *svn log* da ferramenta SVN, informando todas as alterações feitas e listando, para cada alteração: autor; data e hora; número de revisão; e o comentário da operação de *commit*. Mais detalhes podem ser obtidos através de variações do comando *svn log*, como a adição de opções para listar apenas mudanças de uma revisão em específico ou detalhamento de mudanças envolvendo criação, exclusão ou movimentação de arquivos. É importante realizar a comparação entre configurações, *baselines* ou até mesmo em itens de configuração, para a verificação de sua evolução. O SVN permite também a visualização de mudanças no próprio corpo do arquivo, através da funcionalidade *svn diff*, que permite a visualização das linhas alteradas em um dado arquivo de uma revisão para outra. Para realizar o mapeamento entre configurações/*baselines* e a versão de seus itens, é possível utilizar a funcionalidade *svn list*, que lista todos os itens de uma dada revisão.

7.4. Gerenciando Mudanças

Além de definir e controlar os itens de configuração e *baselines* deve ser mantido um sistema capaz de gerenciar as solicitações de modificações sobre os itens de configuração dos projetos. O controle de modificação consiste em gerenciar o ciclo de vida de uma mudança desde sua solicitação até a sua conclusão, com o objetivo de auxiliar na análise do impacto que a mudança causará, além de permitir a sua notificação às pessoas que serão afetadas.

7.4.1. Registrando Solicitações de Mudanças e Problemas

O Redmine possui um mecanismo de registro de problemas chamado de *issue*. Através desse mecanismo, os problemas são registrados e acompanhados através de um ciclo de vida baseado em estados. Deve ser registrado no

Plano de Gerência de Configuração o ciclo de vida e os critérios para a aprovação das solicitações de mudança. A resolução dos problemas e mudanças é acompanhada alimentando os *issues* com informações das ações tomadas, alterando os estados dos *issues*, conforme sua evolução, e atribuindo-os a responsáveis.

No preenchimento do formulário do *issue*, deve-se registrar: Nome/*Subject*, identificador claro e facilmente relacionável ao problema ou mudança em questão; Descrição da Solicitação de Modificação, onde é relatado o caráter da mudança, quais os itens de configuração relacionados, a origem do problema e demais informações que facilitem a solução da modificação; Tipo ou *Tracker*, classifica o *issue* de acordo com os possíveis tipos definidos pela organização, os padrões são “Problema”, “Tarefa” e “Melhoria”; Categoria, determina a área de origem da solicitação de mudança, podendo ser áreas de processo ou componente do sistema; Versão, especifica a qual versão/*baseline*/*release* do sistema que a mudança refere-se.

7.4.2. Acompanhando as Mudanças

O controle das modificações é obtido a partir do acompanhamento da evolução dos *issues* em harmonia com as alterações feitas sobre os itens de configuração, onde cada ação tomada e alteração feita devem ser registradas no *issue*, adicionando comentários e atualizando o estado da solicitação, estabelecendo um ciclo de vida para a mesma, desde sua criação até seu fim, tenha ela sido rejeitada ou resolvida.

O controle pode ser feito através dos históricos do sistema de controle de mudança, que armazena todas as mudanças feitas nas solicitações de modificação, fornecendo uma visão abrangente das ações tomadas para a conclusão das modificações. O acompanhamento das mudanças que estão ocorrendo no projeto é possível a partir do sistema de visualização de *issues*, onde o usuário pode obter uma lista dos *issues* do projeto ordenados e filtrados conforme as

necessidades da organização. A lista é útil para informar o andamento das várias solicitações de mudança registradas.

7.5. Definindo Políticas de Acesso e Segurança

É importante definir como tratar o acesso concorrente aos dados do repositório, para evitar problemas de perda de informação ou retrabalho. Para isto as ferramentas de controle de versão utilizam as metodologias de acesso concorrente ao repositório. Também é possível ramificar o projeto, isto é criar ramos (*branches*), utilizando variações das funcionalidades de criação de rótulos, já explicadas anteriormente. Além disso, as informações devem ser disponibilizadas através de um canal de comunicação seguro. Para tanto, deve ser configurado algum protocolo de segurança e autenticação para o acesso ao repositório de forma controlada, estabelecendo políticas de permissão de acesso, definindo usuários com poder de leitura e escrita.

Nas ferramentas utilizadas nesta metodologia foi configurado o protocolo SSL (*Secure Socket Layer*). O sistema em geral deve ser configurado de forma que apenas os responsáveis (estabelecidos no Plano de Gerência de Configuração) tenham acesso aos itens de configuração e informações relevantes do projeto, com permissões apropriadas de leitura e escrita.

7.6. Realizando Auditoria de Configuração

Para concluir o escopo do processo de GCO, ainda é necessário auditá-lo para verificar se as atividades de GCO foram realizadas de maneira apropriada. É necessária a geração de um roteiro para a realização desta auditoria, implementado através de um *checklist* na ferramenta Spider-CL, contendo os critérios objetivos para condução de uma auditoria de configuração. A auditoria de configuração tem o principal objetivo de garantir

que as *baselines* e os itens de configuração estejam íntegros, completos e consistentes.

Utilizando a Spider-CL, será gerado um *checklist* objetivo contendo critérios para a execução da auditoria de configuração, tanto funcional, avaliando a corretude da *baseline*, quanto física, verificando que a mesma está completa. O *checklist* deve ser dividido de forma a deixar claro estas duas categorias de auditoria. O *checklist* gerado deve ser aplicado utilizando o histórico de mudanças do SVN e Redmine como insumos para a verificação dos itens do *checklist*. Para apresentar e manter os resultados de auditorias ao longo do projeto deve ser criada uma página *wiki* apenas com este objetivo, onde serão listadas todas as auditorias realizadas com informações da data de execução e o auditor responsável. Cada uma das auditorias listadas deve possuir um *link* para uma página *wiki* própria daquela auditoria.

7.7. Aderência da Metodologia Proposta ao MR-MPS

A análise de aderência entre a metodologia proposta ao modelo MR-MPS-SW é feita através do mapeamento das práticas sugeridas nas seções anteriores com os resultados esperados do processo do processo GCO [SOFTEX, 2012a]. É importante ressaltar que a metodologia restringe-se a apresentar boas práticas para que seja possível contemplar os resultados esperados do processo de GCO, em conjunto com práticas definidas no processo de software da organização. Os resultados esperados são um conjunto de boas práticas de um determinado processo, que servem como diretrizes, durante uma avaliação, para determinar o grau de implementação do processo em questão em uma organização, ou “um resultado observável do sucesso do alcance do propósito do processo” [ISO/IEC, 2004a]. No caso do processo de GCO, são sete resultados esperados. As funcionalidades/políticas de uso das ferramentas citadas estão mapeadas com os resultados esperados do processo de GCO na Tabela 8.

Tabela 8. Mapeamento da Metodologia com os Resultados Esperados de GCO do MPS
[Souza *et al.*, 2010]

Resultados Esperados	Ferramentas	Funcionalidades/Práticas
GCO 1	Subversion/ Redmine	Definição do Plano de Gerência de Configuração em <i>Wiki</i> ; Definição do Repositório.
GCO 2	Redmine/ Spider-CL	Definição dos Itens de Configuração com base em Critérios Objetivos definidos através de <i>Checklist</i> .
GCO 3	Subversion/ Redmine	Criação de <i>Baselines</i> através de <i>tags</i> (rótulos) do Projeto no Subversion e Versões do Projeto no Redmine.
GCO 4	Subversion	Acompanhamento do Histórico dos Itens de Configuração no Sistema de Controle de Versão.
GCO 5	Redmine	Solicitação e Acompanhamento de Mudanças pelo Sistema de <i>Issues</i> do Redmine.
GCO 6	Subversion/ Redmine	Políticas de Permissão de Acesso, Concorrência e Garantia da Segurança dos Canais de Dados.
GCO 7	Redmine/ Spider-CL	Realização das Auditorias de Configuração Física e Funcional, guiadas por <i>Checklists</i> do Spider-CL; Publicar os Resultados; Reportar e Tratar as Não-Conformidades observadas.

Vale enfatizar que adequação das práticas propostas neste trabalho ao CMMI-DEV ocorre analisando os resultados esperados descritos na tabela acima com as compatibilidades descritas em [SOFTEX, 2012b] às práticas específicas deste modelo.



Capítulo 8

Implementação do Processo de Garantia da Qualidade

em [Teles e Oliveira, 2010], [Barbalho et al., 2011], [Raiol et al., 2014]

Iuri Igonez Silva Raiol

Luiz Otávio Danin de Lima

Marília Paulo Teles

Rodrigo Araújo Barbalho

Sandro Ronaldo Bezerra Oliveira

Conforme o IEEE (1990), a Engenharia de Software é “(1) a aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de um software. (2) O estudo das abordagens definidas em (1)” (tradução livre). A partir deste conceito observa-se que a engenharia de software é uma ciência exata que necessita de monitoramento e controle durante o processo de desenvolvimento de software para garantir a qualidade do produto e do processo, e como Pressman (2011) define: “A pedra fundamental que sustenta a engenharia de software é o foco na qualidade”.

Nesse contexto, a garantia de qualidade em um processo de desenvolvimento de software é fundamental para que se obtenha êxito em um projeto. A partir disso surgem modelos que definem padrões que auxiliam a criação do processo que uma equipe de desenvolvimento de software, um destes modelos faz parte do programa MPS.BR [SOFTEX, 2012a], o MR-MPS-SW.

É a partir deste modelo que se observa a garantia da qualidade do processo de software como um dos pilares para a construção de um desenvolvimento de software com qualidade, efetividade e eficiência.

No contexto das ferramentas de apoio aos processos, neste trabalho é apresentada a Spider-QA, uma ferramenta de apoio à gerência de qualidade que busca atender os resultados esperados do processo de Garantia da Qualidade (GQA) constante no MR-MPS-SW. Assim, este trabalho tem o objetivo geral de apresentar a ferramenta Spider-QA e descrever todo o apoio provido pela ferramenta ao atendimento das práticas constantes no processo de Garantia da Qualidade.

8.1. A Ferramenta Spider-QA

Visando garantir o desenvolvimento de projetos de software com qualidade, pautados nos modelos e normas de qualidade (CMMI, MPS.BR, ISO/IEC 12207, PMBOK, SWEBOK), pensou-se em um processo de Garantia da Qualidade genérico, que pudesse atender às diferentes realidades de desenvolvimento de software de empresas, ou seja, um processo de desenvolvimento que não estivesse voltado para a análise de uma organização específica, mas sim conseguisse envolver as demais normas e modelos pelos quais os projetos de desenvolvimento de software orientam-se.

Nesse intuito, orientado por esse referencial, foi elaborado o *framework* do processo de Garantia da Qualidade [Teles, 2011], que visa minimizar a adoção das práticas constantes nos diferentes modelos citados, valendo-se como base os resultados esperados do MR-MPS-SW, onde a partir dele outras normas e padrões foram comparados. Para facilitar a utilização deste *framework*, foi desenvolvida a ferramenta Spider-QA, auxiliando a implementação dos ativos de processo (atividade, artefatos, procedimentos, etc.) de Garantia da Qualidade definidos. É dentro deste contexto que o processo de desenvolvimento de um software relaciona-se com a qualidade do

produto, sendo assim a ferramenta Spider-QA auxilia na melhoria do processo de produção do software de uma organização, apoiando a gerência da qualidade por meio de subsídios necessários para as revisões dos produtos e processos.

Esta ferramenta pretende objetivamente auxiliar a execução da Garantia da Qualidade e gerar os indicadores necessários para que este processo esteja sendo corretamente executado, no que diz respeito aos resultados esperados definidos no MR-MPS-SW. Sendo assim, esta ferramenta pode ser utilizada por uma organização que pretenda alcançar uma avaliação neste modelo de maturidade.

A ferramenta é gratuita, com licença GPL - *General Public License*, tornando-se uma importante vantagem no quesito custos para a empresa que pretende adotar um modelo de qualidade de software. Aliás, a organização poderá utilizá-la em conjunto com o *framework* de processo, possibilitando uma solução mais completa e adequada às organizações. Contudo, é possível uma empresa utilizar somente o *framework* do processo ou a ferramenta. Vale ressaltar que a ferramenta pode ser utilizada em diferentes organizações, independente do seu porte, podendo, ainda, ser utilizada durante a realização de consultoria que envolva a implantação das práticas do processo de Garantia da Qualidade em empresas. A ferramenta está disponível em www.spider.ufpa.br.

Nesta seção serão apresentadas as principais funcionalidades da Spider-QA, que conta com módulos referentes à Gerência de *Checklists*, Gerência do Projeto de Software, Plano de Qualidade, Realização de Auditorias e Plano de Ação.

8.1.1. Gerência de *Checklists* de Produtos e Processos de Software

A Spider-QA possui uma estrutura baseada em *Checklists*, e a partir destes são criados itens que descrevem cada etapa ou parte de um Produto de

Trabalho ou um Processo/Disciplina, previamente cadastrados. Estes itens compõem o *Checklist*.

A Gerência do *Checklist* é realizada por meio das funcionalidades apresentadas na Figura 18, a saber: Adicionar *Checklist*, Detalhes, Itens, Editar e Excluir. O botão “Detalhes” apresenta as informações cadastradas do *Checklist* e itens pertencentes ao *Checklist*. O botão “Itens” representa as operações de adicionar, editar e excluir os itens que compõem o *Checklist* criado.

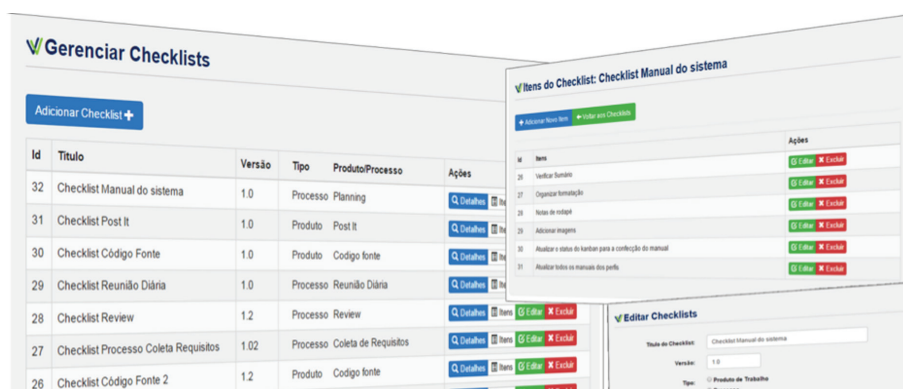


Figura 18. Gerência do *Checklist*

O *Checklist* é vinculado a um Produto de Trabalho ou Processo/Disciplina previamente cadastrado no sistema, a fim de representar um instrumento de avaliação objetiva deste produto ou processo.

8.1.2. Gerência do Projeto de Software

A Gerência do Projeto de Software é realizada por meio das ações apresentadas na Figura 19: Adicionar Projeto, Participantes, Fases, Disciplina/Produto, Editar e Excluir. Neste módulo, o projeto de software criado receberá a execução da Garantia da Qualidade, vinculando os elementos que o compõe. Na ação “Participantes” são vinculados o perfil e nome do participante,

por exemplo, João participa do projeto como Analista de Requisitos. Na ação “Fases” são vinculadas as fases e o período de duração no projeto, por exemplo, a fase de Concepção do RUP tem um período inicial e final. Na ação “Disciplina/Produto” são vinculados os Processos/Disciplinas e os Produtos de Trabalho do ciclo de vida do projeto, por exemplo, uma disciplina é a de Requisitos e um produto de trabalho seria o documento de Especificação de Requisitos. Além dessas ações é possível editar e excluir as informações do projeto registradas.

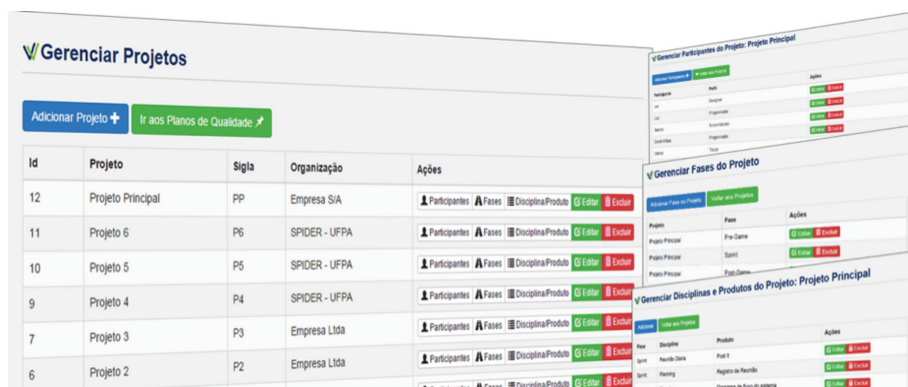


Figura 19. Gerência dos Projetos de Software

8.1.3. Plano de Qualidade

A gerência do Plano de Qualidade, vista na Figura 20, é realizada por meio da seleção de um projeto, previamente cadastrado, no qual é possível vincular a este projeto um Plano de Qualidade para orientar as ações de execução do processo de Garantia da Qualidade. O Plano de Qualidade é composto por descrição, tarefas, objetivos, detalhamento das auditorias de produto e do processo, métodos, contexto do projeto, resultados e um controle de versão. O Plano de Qualidade possibilita a inclusão de uma ou mais auditoria(s), que será descrita a seguir.

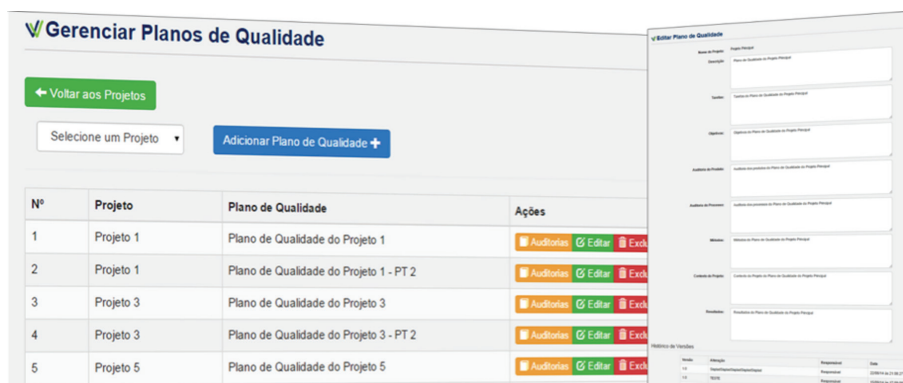


Figura 20. Gerência do Plano de Qualidade

8.1.4. Auditorias

A Gerência da Auditoria é realizada por meio da ação “Adicionar Auditoria”, vista na Figura 21. Esta ação trata da descrição de como, quando e onde irão ocorrer as auditorias registradas no Plano de Qualidade, e quais itens que compõem o *Checklist* serão usados. Estes itens dizem respeito aos Produtos de Trabalho, a exemplo o Código Fonte, e aos Processos, a exemplo o Teste de Software, vinculados anteriormente no projeto, pertencentes a esta auditoria. Neste caso, cada um destes será vinculado ao *Checklist* que servirá como base para a execução da auditoria registrada. É possível também comunicar a realização da auditoria planejada aos participantes do projeto por e-mail pela Spider-QA, possibilitando a comunicação entre todos os participantes do projeto auditado. A tela de Gerência da Auditoria pode ser observada na Figura 21.

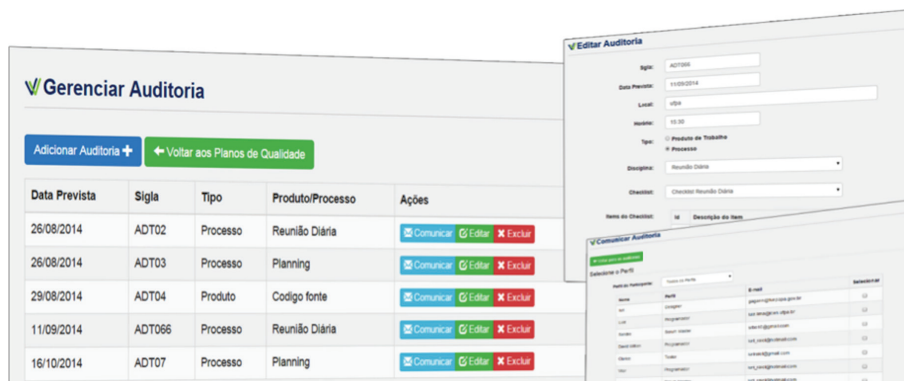


Figura 21. Gerência da Auditoria

Após a Gerência da Auditoria, pode-se realizar a execução da auditoria nos itens do *Checklist* vinculado, como visto na Figura 22. Esta execução ocorre com a classificação dos itens auditados conforme os seguintes critérios definidos: “Contempla”, “Contempla em Partes”, “Não Contempla” e “Não se Aplica”; sendo que cada item pode conter observações detalhadas em cada um dos critérios, itens que compõem o *Checklist*. A auditoria pode ser colocada em três estados diferentes: “Novo”, “Andamento” e “Finalizada”. Após finalizada, a auditoria não pode ser executada novamente, restando somente a realização do Plano de Ação, que será discutido adiante.

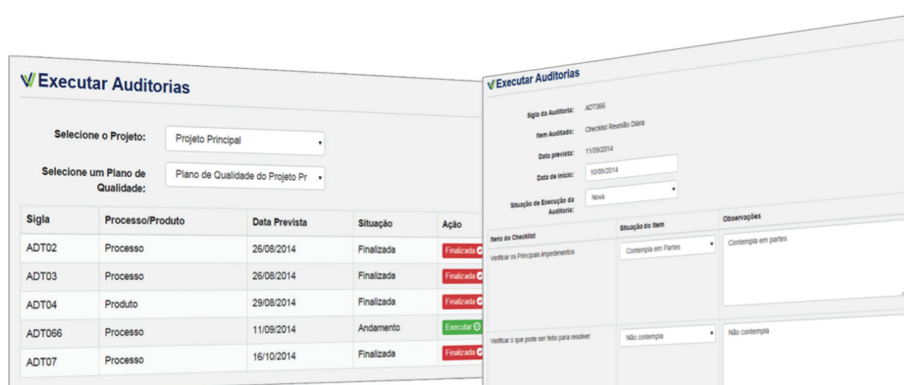


Figura 22. Execução da Auditoria

8.1.5. Plano de Ação

O Plano de Ação está diretamente relacionado a um Projeto de Software e um Plano de Qualidade, previamente definidos, podendo somente executar auditorias finalizadas. A gerência pode ser realizada por meio das operações: “Adicionar Plano de Ação”, “Itens”, “Executar”, “Ações” e “Comunicar”, conforme observado na Figura 23.

O plano pode ser adicionado após a vinculação deste a uma auditoria finalizada, definindo uma data de início e término da realização do plano e o controle das versões. Após este registro, pode-se “Executar” o Plano de Ação, sendo necessária a vinculação de um ou mais itens auditados a partir do registro de uma ação de resolução para as não conformidades coletadas na auditoria, definindo um responsável, prazo e o item da ação definido na ferramenta de *bugtracking*. A integração com a ferramenta de *bugtracking* independe da ferramenta utilizada, pois previamente é cadastrado o *link* de acesso à ferramenta no projeto o qual este plano de ação está vinculado. A necessidade do registro deste item dá-se para o acompanhamento da resolução das não conformidades a partir de uma máquina de estados.

Gerenciar Planos de Ação

Selecione o Projeto:

Selecione um Plano de Qualidade:

[Adicionar Plano de Ação +](#)

Plano de Ação	Data de Início	Data de Término	Situação	Ações
PAAD03	23/09/2014	26/09/2014	Novo	Editar Itens Executar Ações Comunicar
PACAOAD04	01/09/2014	03/09/2014	Andamento	Editar Itens Executar Ações Comunicar
Projeto Principal-07	16/10/2014	22/10/2014	Finalizada	Editar Itens Executar Ações Comunicar

Consultar Ações

Nº	Descrição da Ação	Responsável	Prazo	Itens do Checklist	Ações
1	teste NOVO	Fabiana	16/10/2014	Organizar formatação	Editar
2	Ação Nova	Luiz	27/09/2014	Adicionar imagens	Editar
3	01	lutf	21/10/2014	Notas de rodapé	Editar
4	02	Luiz	22/10/2014	Atualizar o status do banner para a confecção do manual	Editar

Executar Plano de Ação

Nome do Projeto:

Nome do Plano de Qualidade:

Nome do Plano de Ação:

Nome da Auditoria:

Nº	Item	Situação	Observação	Seleção
<input type="text"/>				

Figura 23. Gerência do Plano de Ação

8.1.6. Infraestrutura para Instalação e Uso

Para que houvesse maior facilidade de utilização da ferramenta, optou-se em realizar a implementação da mesma em uma plataforma Web, sendo seu desenvolvimento pautado no uso de tecnologias de software livre sob licenças *open source*, como a linguagem de programação PHP 5.4, facilitada pela utilização do *framework* de desenvolvimento CakePHP versão 2.4.6, o SGBD MySQL 5.6.12 e o servidor de aplicação Apache 2.4.4. Para a correta operação da ferramenta faz-se necessária a instalação do PHP com versão 5.4 ou superior, o SGBD MySQL 5.6.12 e o servidor de aplicação Apache 2.4. Também, é necessária a habilitação do módulo *mod_rewrite* do Apache, bem como a extensão do PHP *openssl* para envio de e-mail de forma segura.

8.2. Metodologia de Implementação do Processo de GQA com Apoio de Ferramentas

Nesta seção será apresentada a metodologia para implementação do processo de GQA do MR-MPS-SW e da área de processo PPQA do CMMI-DEV. A metodologia nada mais é que apresentar uma forma de utilização conjunta das ferramentas: (1) Spider-QA, para geração de planos de qualidade, realização de auditorias, registro e acompanhamento de não-conformidades e problemas encontrados; (2) Spider-CL, para definição e aplicação de critérios objetivos, a partir de *checklists*; e (3) Mantis, para registro das não-conformidades e problemas encontrados e acompanhamento de suas resoluções. As ferramentas Spider-QA e Spider-CL, bem como manuais de utilização das ferramentas, estão disponíveis em www.spider.ufpa.br. Ressalta-se que a escolha das ferramentas deu-se com base nos resultados obtidos e relatados em [Oliveira *et al.*, 2011].

Neste trabalho entende-se metodologia como sendo a codificação de um conjunto de práticas recomendadas (serviços de apoio/funcionalidades

oferecidos pelas ferramentas), às vezes acompanhada de material de treinamento, programas de educação formal, planilhas, diagramas, tomando como parte um método (processo com uma série de passos, para construir um software) [Pressman, 2006].

8.2.1. Planejando a Garantia da Qualidade

Considerando um ambiente onde estejam devidamente instaladas e configuradas ferramentas de apoio ao processo de GQA, a metodologia inicia com a definição de um Plano da Qualidade. Neste plano deverão conter: (1) quais produtos e processos serão avaliados; e (2) o agendamento de auditorias para realização das avaliações; estas atividades são registradas na Spider-QA. Tanto o MR-MPS-SW quanto o CMMI-DEV recomendam que critérios objetivos podem ser definidos e aplicados através de *checklists*, questionários, ferramentas automatizadas de verificação, entre outros, para a realização de auditorias. Nesta metodologia utilizam-se *checklists* e entrevistas para a realização de avaliações.

Os critérios objetivos que são utilizados para seleção de quais produtos de trabalho ou processos devem ser incluídos para avaliação no Plano da Qualidade. Estes critérios são registrados na ferramenta Spider-CL e em seguida inseridos em um *checklist* que pode ser aplicado através da própria ferramenta ou impresso para uma aplicação manual. Esta seleção deve-se ao fato de uma organização poder dispor de um conjunto grande de produtos de trabalho e processos para o desenvolvimento dos seus produtos de software ou serviços, assim a avaliação deve ser realizada por amostragem. A Figura 24 apresenta a aplicação de um *checklist* cadastrado a partir da ferramenta Spider-CL.

Uma vez definidos os produtos e os processos que serão submetidos à avaliação, gera-se um Plano da Qualidade através da Spider-QA, incluindo-se o agendamento de auditorias programadas para avaliação dos produtos,

a partir dos artefatos definidos no processo, e dos processos previamente selecionados, a partir dos produtos e da alocação dos membros alocados nos projetos de software em avaliação. Durante o registro de uma auditoria, devem-se definir: a data, o local e o horário previstos para realização da mesma; o tipo de auditoria (produto ou processo); a seleção de qual produto ou processo será avaliado; e o *checklist* que será utilizado para avaliação. O *checklist* que será utilizado para avaliação durante a auditoria é definido na ferramenta Spider-QA.

Membros alocados nos projetos de software que irão participar das auditorias, a partir de entrevistas, devem ser registrados como participantes na ferramenta Spider-QA e, posteriormente, devem ser comunicados sobre a realização das auditorias. A ferramenta Spider-QA possibilita o envio de e-mails contendo o nome do projeto a qual a auditoria está relacionada, a data, hora e o local da realização da mesma.

A imagem mostra uma janela de software intitulada "Preenchimento de Checklist". O formulário contém os seguintes campos:

- CheckTeste**: Um campo de texto para o nome do teste.
- Descrição do checklist:**: Um campo de texto grande para a descrição do checklist, com o rótulo "descricao" no canto superior esquerdo.
- Descrição do uso do checklist:**: Um campo de texto grande para a descrição do uso do checklist.
- Critério 1:**: Um campo de texto para o primeiro critério, com o rótulo "Critério 1" no canto superior esquerdo.
- alt1**: Um campo de texto pequeno para o primeiro alternativa, com o rótulo "alt1" no canto superior esquerdo.
- alt2**: Um campo de texto pequeno para o segundo alternativa, com o rótulo "alt2" no canto superior esquerdo.
- Descrição:**: Um campo de texto grande para a descrição do critério, com o rótulo "Descrição:" no canto superior esquerdo.

No canto inferior direito da janela, há dois botões: "Salvar" e "Cancelar".

Figura 24. Aplicação de um *Checklist* através da Ferramenta Spider-CL

8.2.2. Executando a Garantia da Qualidade

Uma vez definido o Plano da Qualidade, serão executadas as auditorias agendadas para avaliar a aderência dos produtos de trabalho e dos processos aos padrões de qualidade previamente estabelecidos. A execução das auditorias é realizada através da ferramenta Spider-QA, conforme definido na etapa Planejando a Garantia da Qualidade. Durante a execução informa-se a data de início da auditoria, a situação da mesma (sendo os possíveis estados da auditoria: Novo, Em Andamento ou Finalizada), e a avaliação do que está sendo auditado.

8.2.3. Registrando Não-Conformidades e Problemas

Durante a realização da auditoria não-conformidades e problemas podem ser detectados e, posteriormente, registrados. A ferramenta Spider-QA possibilita o registro das não-conformidades e dos problemas encontrados, conforme os critérios objetivos definidos previamente na criação dos *checklists* de auditoria. Cada item do *checklist* deve ser avaliado usando os seguintes graus: Contempla, Contempla Parcialmente, Não Contempla ou Não se Aplica. A ferramenta permite, ainda, o preenchimento de um campo de observação que pode ser utilizado para descrever a não-conformidade e/ou problema encontrado, ou fazer uma sugestão de melhoria/solução.

Após a finalização da auditoria as pessoas interessadas devem ser comunicadas sobre as não-conformidades e os problemas encontrados. A ferramenta Spider-QA permite que seja enviado um e-mail para as pessoas interessadas contendo um relatório informando os problemas encontrados.

8.2.4. Definindo o Plano de Ação

Após a finalização das auditorias, deverá ser definido um Plano de Ação para a resolução das não-conformidades e dos problemas encontrados, onde

deverão ser informados: a pessoa responsável pela resolução; e um prazo máximo para resolução.

A ferramenta Spider-QA permite que sejam criados Planos de Ação, no entanto, é necessário registrar um projeto na ferramenta de *bugtracking*, Mantis, e cadastrar casos que servirão para indicar não-conformidades e problemas a serem resolvidos. Cada caso pode possuir um estado que indica a situação da resolução do mesmo em um determinado momento, por exemplo, quando um novo caso é registrado ele inicia com status de “Novo” e quando este caso é encaminhado para resolução para a pessoa responsável, passa, então, para o estado de “Atribuído”. O uso do Mantis faz-se necessário já que este possui uma máquina de estados sistematizada para tratamento e relato dos casos provenientes das não-conformidades e dos problemas.

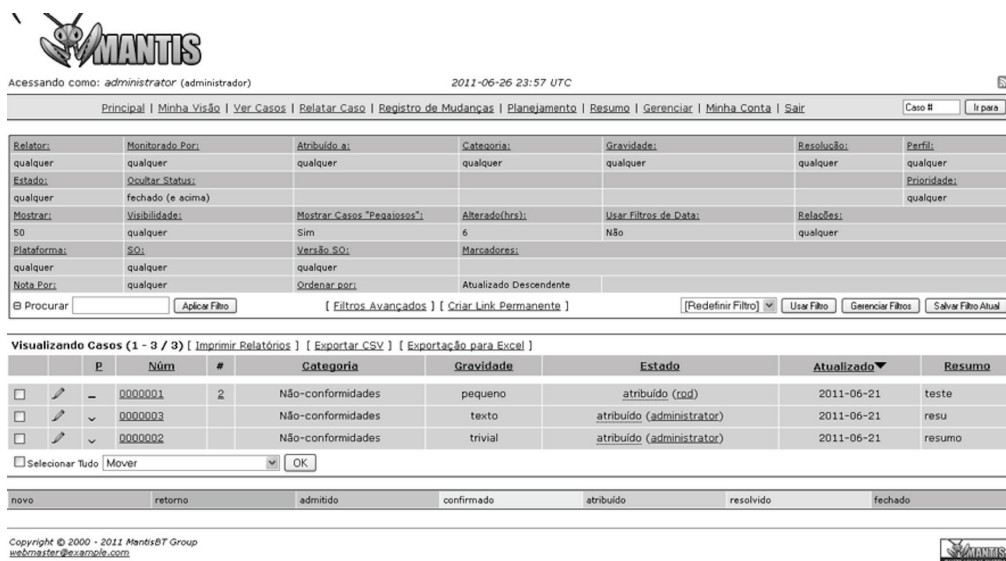
Com o projeto e os casos devidamente registrados no Mantis, o passo seguinte é, a partir da Spider-QA, definir quais itens entrarão no Plano de Ação. A Spider-QA realiza uma integração com a ferramenta Mantis onde os itens selecionados para resolução são incluídos nos casos criados a partir do seu identificador.

8.2.5. Acompanhando as Resoluções

Com a finalização da definição do Plano de Ação, o responsável pela garantia da qualidade deve fazer o acompanhamento das resoluções a partir da Spider-QA. A ferramenta realiza uma comunicação com o Mantis para obter informações de uma determinada não-conformidade e problema cadastrados previamente, apresentando o item, o responsável pela resolução, o prazo para resolução, o código do problema cadastrado no Mantis e um resumo do problema.

Caso uma não-conformidade ou um problema registrado não seja resolvido dentro do prazo definido no Plano de Ação, o responsável pela garantia da qualidade deve, através do Mantis, realizar o escalonamento deste

problema ou não-conformidade, ou seja, atribuí-lo a outra pessoa em um nível mais elevado na hierarquia da empresa, para tentar resolver o problema. A Figura 25 apresenta uma tela da ferramenta Mantis contendo os casos criados e a situação dos mesmos.



The screenshot shows the Mantis Bug Tracker interface. At the top, there is a navigation bar with links: Principal | Minha Visão | Ver Casos | Relatar Caso | Registro de Mudanças | Planejamento | Resumo | Gerenciar | Minha Conta | Sair. The date and time are 2011-06-26 23:57 UTC. Below the navigation bar, there is a table of filters for cases. The table has columns: Relator, Monitorado Por, Atribuído a, Categoria, Gravidade, Resolução, Perfil, Estado, Ocultar Status, fechoado (e acima), Prioridade, Visibilidade, Mostrar, Versão SO, Alterado(hrs), Usar Filtros de Data, Relações, Plataforma, SO, Marcadores, and Nota Por. Below the table, there are buttons for 'Procurar', 'Aplicar Filtro', 'Filtros Avançados', 'Criar Link Permanente', 'Redefinir Filtro', 'Usar Filtro', 'Gerenciar Filtros', and 'Salvar Filtro Atual'. Below the buttons, there is a section titled 'Visualizando Casos (1 - 3 / 3)' with links for 'Imprimir Relatórios', 'Exportar CSV', and 'Exportação para Excel'. Below this, there is a table of cases with columns: P, Núm, #, Categoria, Gravidade, Estado, Atualizado, and Resumo. The table contains three rows of data. Below the table, there is a button for 'Selecionar Tudo' and a 'Mover' button. At the bottom, there is a row of buttons: novo, retorno, admitido, confirmado, atribuído, resolvido, and fechado. At the very bottom, there is a copyright notice: Copyright © 2000 - 2011 MantisBT Group, webmaster@exarnole.com, and the Mantis logo.

P	Núm	#	Categoria	Gravidade	Estado	Atualizado	Resumo
<input type="checkbox"/>	0000001	2	Não-conformidades	pequeno	atribuído (rod)	2011-06-21	teste
<input type="checkbox"/>	0000003		Não-conformidades	texto	atribuído (administrador)	2011-06-21	resu
<input type="checkbox"/>	0000002		Não-conformidades	trivial	atribuído (administrador)	2011-06-21	resumo

Figura 25. Resumo dos Casos Existentes Cadastrados na Ferramenta Mantis.

8.3. Aderência da Metodologia Proposta ao MR-MPS-SW e CMMI-DEV

A análise de aderência entre a metodologia proposta ao modelo MR-MPS-SW e ao modelo CMMI-DEV é realizada através do mapeamento das práticas descritas na seção 8.2, com os resultados esperados do processo de Garantia da Qualidade constantes no Guia Geral do MPS.BR [SOFTEX, 2012a] e as práticas específicas da área de processo de Garantia da Qualidade do Produto e Processo constante no CMMI-DEV [SEI, 2010a], conforme visualizado na Tabela 9 [Teles, 2011]. É importante ressaltar que a metodologia restringe-se a apresentar boas práticas para que seja possível contemplar os resultados

esperados do MR-MPS-SW e as práticas específicas do CMMI-DEV em conjunto com as práticas definidas no processo de software da organização. Resultado Esperado é “um resultado observável do sucesso do alcance do propósito do processo” [ISO/IEC, 1995] e Prática Específica “é a descrição de uma atividade considerada importante para alcançar uma meta específica associada” [SEI, 2010a].

Tabela 9. Mapeamento da Metodologia com os Resultados Esperados do MR-MPS-SW e Práticas Específicas do CMMI-DEV [Teles, 2011]

MR-MPS-SW	CMMI-DEV	Ferramenta(s) de Apoio	Funcionalidade(s) das Ferramentas
GQA1	SP 1.2	Spider-QA Spider-CL	Definir o Plano da Qualidade Agendar Auditorias Aprovar o Plano da Qualidade Preparar Auditorias Selecionar Material para Auditoria Executar Auditorias
GQA2	SP 1.1	Spider-QA Spider-CL	Definir o Plano da Qualidade Agendar Auditorias Aprovar o Plano da Qualidade Preparar Auditorias Selecionar Material para Auditoria Selecionar Participantes para Entrevistas Executar Auditorias
GQA3	SP 2.1 SP 2.2	Spider-QA Mantis	Executar Auditorias Notificar Não-Conformidades ou Problemas
GQA4	SP 2.1	Spider-QA Mantis	Estabelecer Datas para Correção Corrigir Não-Conformidades ou Problemas Verificar Correções Reportar Não-conformidades ou Problemas encontrados à níveis mais altos na hierarquia da organização Finalizar Não-Conformidades ou Problemas Notificar Resultados



Capítulo 9

Implementação do Processo de Aquisição

em [Furtado e Oliveira, 2011a], [Furtado e Oliveira, 2011b]

Julio Cezar Costa Furtado

Sandro Ronaldo Bezerra Oliveira

As organizações são adquirentes de suas capacidades necessárias a partir da compra de Softwares e Serviços Correlatos (S&SC) de fornecedores, e desenvolvem cada vez menos estas necessidades de forma *in-house* [SEI, 2010b]. Muitas organizações têm necessidades específicas que só conseguem ser atendidas a partir do desenvolvimento direto de um produto que atenda a estas necessidades, sendo, muitas vezes, a solução mais viável a terceirização do desenvolvimento do produto [Guerra e Alves, 2004].

A terceirização de produtos e serviços de software é uma tarefa complexa e necessária para estas organizações. Considerando as dificuldades encontradas quanto ao atendimento das exigências para o desenvolvimento dos requisitos de sistemas e visando diminuir os riscos quanto ao cumprimento de estimativas de prazo e custos para seus projetos de aquisição, as organizações adquirentes têm optado pela adoção de processos de aquisição como forma de alcançar os objetivos traçados para seus projetos [SOFTEX, 2013]. Assim, o adquirente tem um maior acompanhamento sobre este contrato de compra, da preparação das necessidades da aquisição, passando pela seleção do fornecedor até a entrega do produto ou serviço.

Neste contexto, este trabalho apresenta a ferramenta Spider-ACQ, uma solução de ferramenta de software livre de apoio ao processo de aquisição como definido pelo Guia de Aquisição do MPS.BR [SOFTEX, 2013] e pelo CMMI-ACQ [SEI, 2010b].

9.1. A Ferramenta Spider-ACQ

A principal preocupação na definição da arquitetura utilizada pela ferramenta Spider-ACQ foi possibilitar uma integração com ferramentas externas da forma o mais transparente possível para o usuário. Para tal, a ferramenta integra-se a ferramentas externas, também livres: (a) utiliza a OSRMT, uma ferramenta para Gerência de Requisitos que possibilita também manter a rastreabilidade entre os requisitos elicitados; (b) para o planejamento de esforço e custo é utilizado a OpenProj, uma ferramenta de apoio à Gerência de Projetos; (c) para criação e aplicação de *checklist* é utilizada a ferramenta Spider-CL [Barros e Oliveira, 2010a]; (d) para elaboração do Plano de Medição é utilizada a ferramenta Spider-MPlan [Estácio e Oliveira, 2010]; e por fim, (e) para comunicar não-conformidades e solicitar mudanças é utilizado a Redmine, como ferramenta de *bugtracking*. Ressaltando que a escolha das ferramentas deu-se com base nos resultados obtidos em [Oliveira *et al.*, 2011], que avaliou a aderência de diversas ferramentas livres e de código aberto ao MR-MPS-SW.

Além de ser uma ferramenta de apoio para instituições que implementam o Guia de Aquisição do MPS.BR e/ou o CMMI-ACQ, a ferramenta também é voltada para empresas que já os adotam, mas desejam facilitar e aprimorar este processo. A ferramenta, em conjunto com as ferramentas externas, possibilita a coleta dos requisitos do S&SC: planejar o projeto de aquisição, selecionar fornecedores através de critérios objetivos, monitorar o processo de desenvolvimento do fornecedor e realizar os testes para aceitação do S&SC produzido.

A ferramenta Spider-ACQ utiliza o conceito de ator para especificar que área realiza determinada atividade no processo de Aquisição. Existindo quatro tipos de atores: a Área Gestora Interessada, responsável pela gestão do projeto de aquisição, como realizar a solicitação da aquisição, esclarecer a necessidade da aquisição; a Área Tecnologia da Informação, responsável pelo olhar técnico da aquisição, como o levantamento de requisitos, revisões técnicas, avaliação do S&SC; a Área de Administração e Pessoas, responsável pela contratação em si, onde irá negociar o contrato com o fornecedor; e a Área Jurídica, responsável pela revisão dos termos contratuais definidos entre as partes envolvidas.

Como forma de organizar a execução, as atividades foram agrupadas em 4 fases: (1) Preparação da aquisição; (2) Seleção do fornecedor; (3) Monitoração da aquisição; e (4) Aceitação pelo cliente. A escolha das fases tomou como base a divisão de atividades do Guia de Aquisição do MPS.BR, por se considerar que esta divisão reflete de forma adequada as fases de um projeto de aquisição, e a Figura 26 expõe uma visão macro do fluxo de atividades da ferramenta. A fase de Preparação da Aquisição e Seleção do Fornecedor são realizadas uma única vez no início do projeto de aquisição, já as fases de Monitoração da Aquisição e Aceitação pelo Cliente devem ser realizadas a cada remessa de produtos entregáveis do fornecedor, a cada iteração.

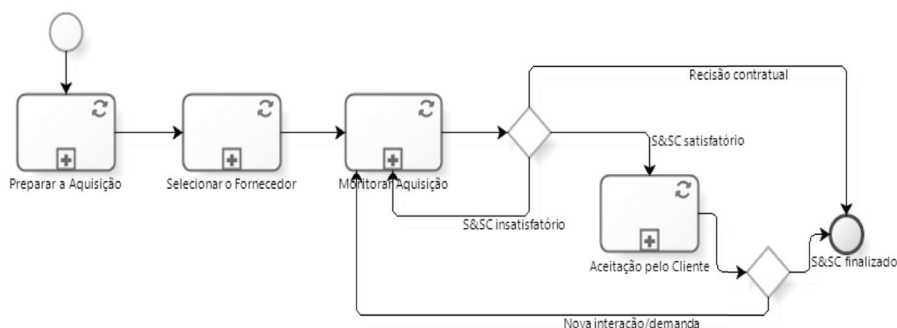


Figura 26. Fluxo de Atividades (Macro) do Processo de Negócio da Spider-ACQ

Considerando um ambiente onde estejam devidamente instaladas e configuradas as ferramentas de apoio ao processo de Aquisição, a metodologia inicia-se informando à ferramenta Spider-ACQ o endereço de acesso às demais ferramentas. O projeto, então, deve ser criado na ferramenta Spider-ACQ que automaticamente replica a informação para as demais ferramentas, assim como os usuários que forem criados na Spider-ACQ.

Neste trabalho entende-se metodologia como sendo a codificação de um conjunto de práticas recomendadas (serviços de apoio/funcionalidades oferecidos pelas ferramentas), às vezes acompanhada de material de treinamento, programas de educação formal, planilhas, diagramas, tomando como parte um método (processo com uma série de passos, para construir um software) [Pressman, 2006].

9.1.1. Preparação da Aquisição

A fase de Preparação da aquisição tem por objetivo o planejamento de tudo que diz respeito ao projeto de aquisição, realizando: a definição da necessidade e objetivos pretendidos com a aquisição, com foco em cadastrar as necessidades a serem atendidas por meio do projeto, assim como a motivação para a aquisição e os resultados pretendidos com a conclusão do projeto; a definição dos interessados no projeto, cadastrando a lista de *stakeholders* que serão considerados durante o projeto; a definição e priorização dos requisitos, elicitando os requisitos, considerando requisitos contratuais, restrições legais, financeiras, de prazo e priorizá-los, a partir do uso da ferramenta OSRMT; a revisão dos requisitos, garantindo que todos os interessados estejam contemplados, verificando conflitos, requisitos incompletos e atingindo um balanço entre a relação custo/benefício, utilizando-se da ferramenta Spider-CL; a análise de viabilidade técnica, analisando os requisitos coletados, principalmente a relação custo/benefício, utilizando-se da ferramenta Spider-CL; o desenvolvimento de uma estratégia de Aquisição,

compatível com as necessidades da aquisição, considerando as opções viáveis; a definição do cronograma, a partir de atividades contempladas durante o projeto e obtendo o acordo das datas e custos definidos, através da ferramenta OpenProj.

Além disso, realiza-se: o estabelecimento do plano de medição, definindo as métricas que serão consideradas durante a execução do contrato, utilizando-se da ferramenta Spider-MPlan; a definição dos critérios para aceitação do S&SC e plano de testes, configurando a aceitação pelo adquirente do produto, utilizando-se da ferramenta Spider-CL; a definição critérios de seleção do fornecedor: definir critérios objetivos que irão respaldar a seleção de um fornecedor, utilizando-se da ferramenta Spider-CL; a elaboração do plano de aquisição, que além de conter o cronograma, plano de medição, testes e critérios de seleção do fornecedor, irá contemplar os riscos envolvidos, assim como uma lista de produtos a serem fornecidos e responsabilidades; o estabelecimento do pacote de solicitação (vide Figura 27), definindo um documento que caracteriza o S&SC pretendido e as condições de entrega, sendo uma composição de especificação de requisitos e do Plano de Aquisição, assim o usuário tem apenas que executar a função e a ferramenta Spider-ACQ irá gerar um arquivo compactado contendo todas as informações necessárias, incluindo aquelas que foram geradas por ferramentas externas; a identificação dos potenciais fornecedores, baseando-se nos critérios para seleção de fornecedores, realizar uma seleção dos fornecedores mais aptos para o projeto, utilizando-se a ferramenta Spider-CL; e a distribuição da documentação da aquisição, enviando por e-mail, aos fornecedores selecionados, o documento de pedido da proposta.

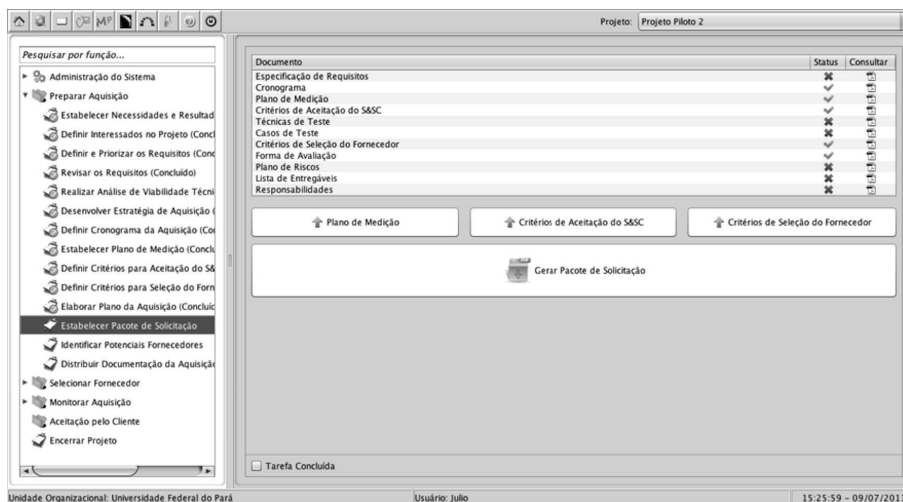


Figura 27. Função “Estabelecer Pacote de Solicitação”

9.1.2. Seleção do Fornecedor

A fase de Seleção do fornecedor contempla a avaliação das propostas recebidas, a negociação do contrato com o fornecedor, culminando com o contrato. Estas ações permeiam em: receber propostas de solução técnica solicitadas aos fornecedores selecionados e encaminhar aos avaliadores; emitir parecer operacional/técnico da proposta, utilizando-se a ferramenta Spider-CL; selecionar o fornecedor que melhor se alinhar às necessidades do adquirente de acordo com pareceres técnico e operacional; preparar e negociar um contrato, expressando as expectativas e responsabilidades das partes; emitir considerações técnicas/jurídicas sobre os termos, garantindo que as expectativas dos aspectos técnico e jurídicos estão sendo atendidas; garantir o bom entendimento dos termos contratuais acordados por ambas as partes, evitando possíveis impasses provenientes de um entendimento divergente; e emitir o contrato, realizando o upload do documento na ferramenta Spider-ACQ.

9.1.3. Monitoração da Aquisição

A fase de Monitoração da aquisição é onde o fornecedor já está desenvolvendo o produto ou serviço e o adquirente mantém o controle sobre o andamento do trabalho. Nesta fase ocorre: o desenvolvimento do plano de inserção, definindo a infraestrutura necessária, assim como, controle de acesso às informações e treinamento ao fornecedor sobre as políticas da organização adquirente; a realização da reunião inicial, realizando o upload da ata da reunião na ferramenta Spider-ACQ; o estabelecimento de um canal de comunicação, cadastrando quais pessoas serão responsáveis pelo canal comunicação e qual forma de comunicação será utilizada para tal; a seleção das interfaces do S&SC adquirido e gerenciá-las, em conjunto com a ferramenta Spider-CL; a troca das informações sobre o progresso técnico, selecionando artefatos para avaliar, além de avaliar aspectos de custos e possíveis riscos e comunicar não-conformidades, em conjunto com as ferramentas Spider-CL e Redmine.

Além disso, realiza-se: a revisão do desempenho do fornecedor, selecionando os produtos que passarão por verificação e validação e comunicar não-conformidades, usando as ferramentas Spider-CL e Redmine; a revisão dos termos contratuais estabelecidos e verificar se os mesmos ainda estão aderentes às necessidades e solicitar mudanças se necessário, usando a ferramenta Redmine; realizar o acordo com as alterações (vide Figura 28), avaliando o impacto das alterações e cadastrando observações/negociações realizadas entre os envolvidos, assim, a ferramenta Spider-ACQ exibe todas as *issues*, referentes a solicitação de mudança, cadastradas na Redmine para o projeto de aquisição selecionado e, para cada *issue*, o usuário deve realizar uma análise de impacto e aprová-la caso pertinente, esta aprovação é uma chamada à ferramenta Redmine, onde o usuário irá realizar a aprovação da *issue*, mudando seu status; o acompanhamento dos problemas identificados até o seu encerramento junto aos responsáveis, fazendo uso da ferramenta Redmine; o monitoramento dos processos do fornecedor, selecionando os processos cruciais para garantir que aspectos como custo, qualidade e prazo sejam atendidos,

com base nas medidas obtidas com a execução dos processos do fornecedor, usando a ferramenta Spider-CL; a tomada de decisão no projeto; e o encerramento do projeto, realizando o upload do termo de encerramento do projeto.

9.1.4. Aceitação pelo Cliente

A fase de Aceitação pelo cliente trata diretamente da aceitação do produto ou serviço desenvolvido pelo fornecedor. Assim, o produto é avaliado e seus resultados são comparados aos requisitos definidos e aos termos contratuais. As atividades permeiam em: revisar o plano de testes elaborado no planejamento da aquisição para garantir que o mesmo ainda está alinhado as necessidades; avaliar o S&SC, os produtos entregues irão passar por verificação/validação; manter conformidade com o contrato, os produtos entregues irão ser avaliados objetivamente ou subjetivamente em relação a conformidade com o que for definido no contrato, com o apoio da ferramenta Spider-CL; aceitar o S&SC, a partir da emissão do parecer de aceitação do S&SC; e gerenciar faturas do fornecedor, cadastrando as notas fiscais/faturas pagas ao fornecedor.

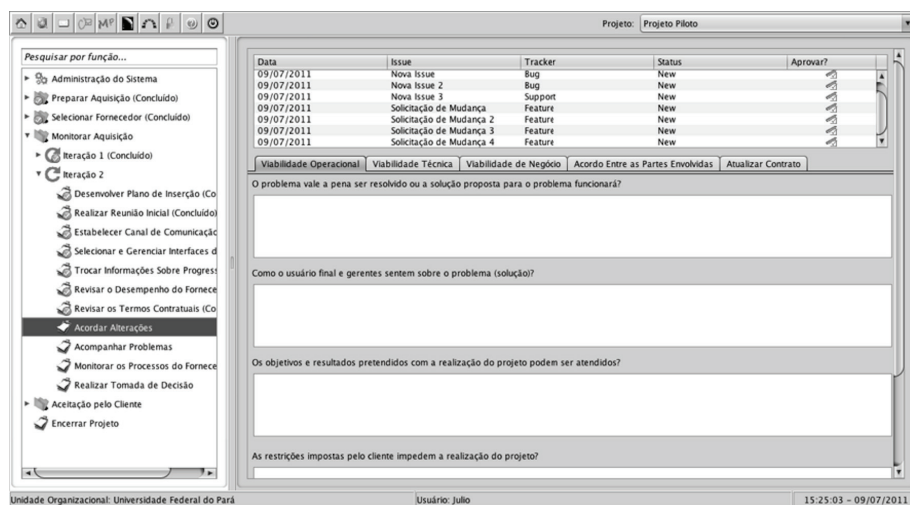


Figura 28. Função “Acordar Alterações”

9.2. Aderência da Ferramenta ao Processo de Aquisição

A análise de aderência da metodologia proposta é feita a partir do mapeamento das práticas sugeridas na Seção 9.1 com as tarefas previstas pelo Guia de Aquisição do MPS.BR. Como forma de ressaltar a abrangência do processo de Aquisição, a análise também evidenciará o atendimento aos resultados esperados (RE) do processo de Aquisição (ACQ) no Guia Geral do MPS.BR [SOFTEX, 2012a]. Resultado Esperado é “um resultado observável do sucesso do alcance do propósito do processo” (ISO/IEC, 1995). No do processo de ACQ do MPS.BR, são nove resultados esperados, mas apenas oito resultados esperados são cobertos pelo processo descrito no Guia de Aquisição do MPS.BR, como pode ser visto na Tabela 10.

Tabela 10. Mapeamento da Ferramenta ao Processo

RE	Tarefas Previstas	Ferramentas	Funcionalidades/Práticas
AQU1	Estabelecer a necessidade; Definir os requisitos; Revisar os requisitos; Desenvolver uma estratégia de aquisição.	Spider-ACQ/ OSRMT/ OpenProj/ Spider-CL/ Spider-MPlan	Definição da necessidade e objetivos; Definição dos interessados no projeto; Definição e priorização dos requisitos; Revisão dos requisitos; Análise de viabilidade técnica; Desenvolvimento de uma estratégia de aquisição; Definição do cronograma; Estabelecimento do plano de medição; Definição dos critérios para aceitação do S&SC e plano de testes; Elaboração do plano de aquisição.
AQU2	Definir os critérios de seleção dos fornecedores.	Spider-ACQ/ Spider-CL	Definição critérios de seleção do fornecedor; Estabelecimento do pacote de solicitação.

RE	Tarefas Previstas	Ferramentas	Funcionalidades/Práticas
AQU3	Avaliar a capacidade dos fornecedores; Selecionar o fornecedor.	Spider-ACQ/ Spider-CL	Distribuição da documentação da aquisição; Identificação dos potenciais fornecedores; Receber propostas.
AQU4	Preparar e negociar um contrato.	Spider-ACQ	Preparar e negociar um contrato; Emitir considerações técnicas/jurídicas sobre os termos; Garantir o bom entendimento dos termos; Emitir o contrato.
AQU5	Avaliar a capacidade dos fornecedores; Selecionar o fornecedor.	Spider-ACQ/ Spider-CL	Emitir parecer operacional/técnico da proposta.
AQU6	Estabelecer e manter comunicações; Monitorar a aquisição.	Spider-ACQ/ Redmine/ Spider-CL	Estabelecimento de um canal de comunicação; Monitoramento dos processos do fornecedor.
AQU7	Estabelecer e manter comunicações; Trocar informações sobre o progresso técnico; Revisar o desempenho do fornecedor; Obter acordo quanto às alterações; Monitorar a aquisição; Acompanhar problemas.	Spider-ACQ/ Redmine/ Spider-CL	Estabelecimento de um canal de comunicação; Desenvolvimento do plano de inserção; Realização da reunião inicial; Troca das informações sobre o progresso técnico; Revisão do desempenho do fornecedor; Revisão dos termos contratuais; Monitoramento dos processos do fornecedor. Acordo com as alterações; Acompanhamento dos problemas.

RE	Tarefas Previstas	Ferramentas	Funcionalidades/Práticas
AQU8	Preparar a aceitação; Avaliar o S&SC entregue; Manter conformidade com o contrato; Aceitar o S&SC.	Spider-ACQ/ Spider-CL	Revisar o plano de testes; Avaliar o S&SC; Manter conformidade com o contrato; Aceitar o S&SC.

Vale enfatizar que adequação das práticas propostas neste trabalho ao CMMI-DEV ocorre analisando os resultados esperados descritos na tabela acima com as compatibilidades descritas em [SOFTEX, 2012b] às práticas específicas deste modelo.



Capítulo 10

Implementação do Processo de Medição

em [Estácio e Oliveira, 2010], [Carneiro e Oliveira, 2011], [Costa et al., 2015]

Bernardo José da Silva Estácio
Bleno Wilson Franklin Vale da Silva
Dan Jhonatan da Silva Teixeira
Gessica Pinheiro da Silva
Paulo Junior S. de Souza
Simone Nayara Costa Carneiro
Thiago Sylas Antunes da Costa
Arthur Ivson Xavier de Moraes Batista
Sandro Ronaldo Bezerra Oliveira
Alexandre Marcos Lins de Vasconcelos

Alcançar um determinado grau de qualidade para poder competir no mercado é uma tarefa árdua e é preciso que se mantenha o controle, por isso a medição faz-se importante nesse contexto, pois de acordo com Tom Demarco “Não se pode controlar o que não se pode medir” [Demarco, 1982]. E para que se possa manter o controle do que se está fazendo, assim como prever o comportamento futuro dos produtos e processos de software, é necessária a utilização de um processo de medição [SOFTEX, 2012a], uma vez que “Não se pode prever o que não se pode medir” [Fenton e Pfleeger, 1997].

Neste contexto, as organizações são movidas por objetivos, os quais são estabelecidos e tidos como alvos a serem alcançados. Para isso, decisões precisam ser tomadas, entretanto por vezes são várias as opções e deve-se saber

qual dentre essas é a melhor. O processo de medição (MED) ajuda a atingir os objetivos das organizações gerando informações necessárias para dar suporte à tomada de decisão [Rocha *et al.*, 2012].

No entanto, um grande problema que afeta diretamente no tempo e consequentemente no custo financeiro está em como executar o processo de medição, pois muitas organizações precisam verificar e elaborar um conjunto de documentos e planilhas, os quais serão utilizados para armazenar as medidas e controlar o processo de medição, que por vezes dificulta o acompanhamento do mesmo [SOFTEX, 2012a]. Entretanto, dispondo de uma ferramenta sistematizada esses empecilhos tornam-se menos agravantes, uma vez que todo o processo vai estar centralizado e por conta disso mais organizado. Assim, a busca de uma informação poderá ser processada e encontrada mais facilmente, os gráficos e os dados serão gerados automaticamente, ao passo que as medidas sejam inseridas na ferramenta, entre outras vantagens. É válido mencionar que ainda são poucos os softwares disponíveis e eficazes para apoiar o processo de medição, como pode ser visto na Seção 6 deste trabalho.

Paralelamente, diversos modelos e guias de boas práticas para processos de software, como o MR-MPS-SW – Modelo de Referência do MPS para Software [SOFTEX, 2012a], a ISO/IEC 12207 [ISO/IEC, 1995] e o CMMI-DEV [SEI, 2010], possuem referência à implementação da área de medição, embasando a importância deste processo. Cada modelo possui um conjunto de metas a serem alcançadas neste processo e sugestões de como implementá-lo. Entretanto, organizações consideradas de micro e pequeno porte muitas vezes não possuem recursos suficientes para implementar um processo e avaliá-lo [SOFTEX, 2012a].

Este trabalho tem o objetivo de apresentar uma ferramenta de software capaz de auxiliar estas organizações a implementarem o processo de medição, dada a sua relevância para atingir os objetivos estratégicos organizacionais. A ferramenta de software proposta neste trabalho baseia-se nas práticas contidas na abordagem GQIM (*Goal-Question-Indicator-Metric*) e

nos modelos e normas de qualidade como o CMMI-DEV, MR-MPS-SW e ISO/IEC 12207.

10.1. Trabalhos Relacionados

Dentre as ferramentas encontradas que também oferecem apoio ao processo de Medição estão o software Remex [Greese e Rodrigues, 2000], a ferramenta Spider-Mplan [Estácio e Oliveira, 2010] e a aplicação MedPlan [Schneider *et al.*, 2004]. Entretanto, além de todas adotarem a abordagem GQM (*Goal Question Metric*) para definir como será realizado o processo de medição, ou seja, não contemplando indicadores, a última aplicação mencionada define que somente o Gerente do Projeto é quem poderá fazer as análises de medição, limitando a análise das causas e tomadas de decisão.

Outra ferramenta que também utiliza a abordagem GQIM é a WebAPSEE [Nascimento *et al.*, 2007], entretanto esta difere do software proposto por também modelar processos de software e só permite que sejam realizadas medições nas atividades constantes nestes processos, o que limita a definição dos indicadores para os projetos. Essa ferramenta também fixa que somente o Gerente é o responsável por realizar as coletas das medidas. Na ferramenta proposta neste trabalho, é permitido ao Administrador controlar o nível de acesso de cada perfil, para poder assim se adequar a diferentes ambientes organizacionais e projetos.

Assim, a Tabela 11 compara as funcionalidades entre as ferramentas Remex, Spider-Mplan, MedPlan, WebAPSEE e Spider-MsControl, tomando como base as funcionalidades a serem descritas na Seção 10.2. Cada ferramenta recebe a identificação: “S”, caso possua a funcionalidade; “P”, caso realize a funcionalidade parcialmente, ou seja, com restrições; “N”, caso não exista tal funcionalidade; e “D”, caso a referência analisada sobre a ferramenta não identifique esta informação.

Tabela 11. Comparativo das Funcionalidades da Ferramenta Spider-MsControl com os Trabalhos Relacionados [Costa *et al.*, 2015]

Funcionalidades	Remex	Spider-Mplan	MedPlan	WebAPSEE	Spider-MsControl
Utilização da metodologia GQIM	N	N	N	S	S
Login e Níveis de Acesso	N	S	N	S	S
Gerenciamento de vários Projetos Simultâneos	N	S	S	S	S
Definição de Objetivos, Necessidades de Informação, Indicadores e Medidas	P	P	P	P	S
Definição dos Procedimentos de Coleta e Análise	P	P	P	S	S
Coleta de Medidas e Análise de Indicadores	P	P	P	P	S
Resultados	D	S	P	P	S
Plano de Medição e Relatório	D	P	P	P	S

10.2. A Ferramenta Spider-MsControl

A ferramenta Spider-MsControl utiliza o conceito de ator para especificar quem realiza determinada atividade no processo de Medição. Existem quatro tipos de atores: o **Analista de Medição**, que detém a maior parte das atividades do processo como definição da abordagem GQIM, definição dos procedimentos de análise e coleta, coletar, analisar, elaborar/divulgar/revisar relatórios; a **Alta Administração**, responsável pelas atividades da definição dos objetivos de medição e, conjuntamente com o Gerente de Projetos, validam métricas, apreciam relatórios, analisam resultados e podem estabelecer tomadas de decisão; o **Usuário de Medição**, responsável pela atividade de fornecer a base de dados para a coleta de métricas; e o **Bibliotecário de Medição**, responsável por qualquer dado manipulado pela ferramenta. É importante que se ressalte que um usuário da ferramenta pode representar mais de um ator, dependendo do grau de envolvimento com o processo de medição.

Nesta seção são apresentadas as principais funcionalidades da ferramenta Spider-MsControl, que fazem dela um diferencial em relação a outras ferramentas disponíveis, descritas na Seção 10.1.

10.2.1. Utilização da Abordagem GQIM

Durante o processo de medição é importante que se use uma abordagem para ajudar na organização e na prática do como fazer a medição de software. Dentre as abordagens mais utilizadas está o GQM (*Goal-Question-Metric*) [Solingen e Berghout, 1999]. Nesta abordagem são definidos os objetivos organizacionais e de projeto, são estipuladas questões para que com suas respostas possam-se atingir os objetivos e responder como essas medidas são definidas, coletadas e analisadas.

A abordagem GQIM é uma variação do GQM. Essa abordagem está baseada no entendimento de que identificar questões e medidas sem visualizar um indicador muitas vezes não é suficiente. O GQIM é mais eficaz e eficiente que o GQM porque toma como base as metas de negócio da organização, que são mais abrangentes e geralmente mais conhecidas [Park *et al.*, 1996]. O GQIM inicia com a pergunta “O que se quer aprender?”. Além disso, no GQIM são introduzidos indicadores, que constituem um nível intermediário entre as perguntas e as métricas para auxiliar a identificação das métricas mais adequadas. A abordagem GQIM, devido sua eficiência e eficácia, se aplicada corretamente, vem ganhando espaço para ser utilizada nos processos de medição e por isso optou-se pela mesma para ser utilizada na concepção e desenvolvimento da ferramenta.

Assim, para manter a aderência à abordagem GQIM, foram definidas as seguintes funcionalidades para a ferramenta: gerenciamento dos objetivos da medição; gerenciamento das necessidades de informações associadas aos objetivos; definição e aprovação dos indicadores para responder as necessidades de informação; caracterização das medidas para

os indicadores. Cada uma destas funcionalidades será melhor apresentada a seguir.

10.2.2. Login e Níveis de Acesso

A ferramenta oferece suporte para que cada usuário possua um *login* e, desta forma, identificando o usuário, pode-se limitar o nível de acesso do mesmo de acordo com o seu perfil, como visto na Figura 29. A ferramenta possibilita com que o perfil de Administrador possa selecionar o que cada perfil poderá realizar durante a execução da ferramenta, o que se mostrou ser uma opção bastante interessante, uma vez que pode ser manipulada para se adequar à realidade das mais diversas organizações e projetos.

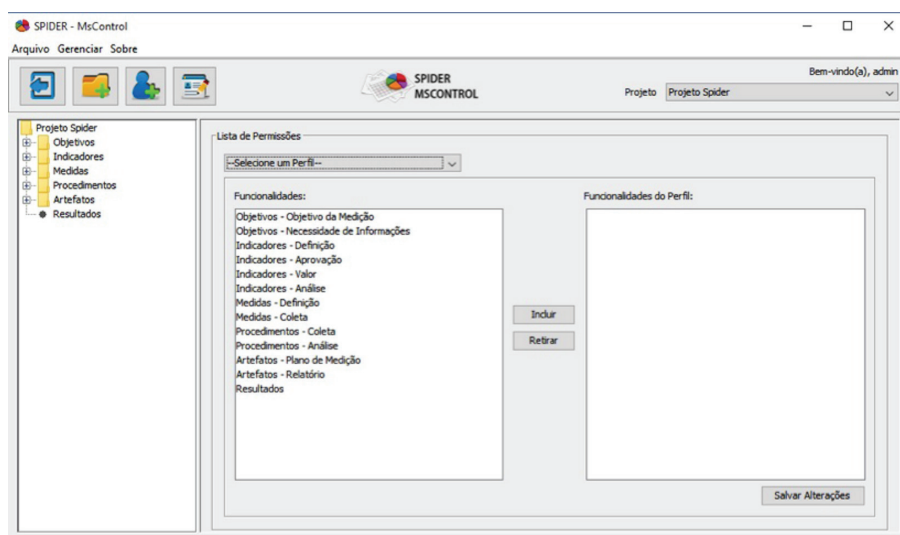


Figura 29. Função *Login* e Controle de Acesso

10.2.3. Gerenciamento Simultâneo de vários Projetos

A ferramenta foi desenvolvida para ser facilmente utilizada para gerenciar múltiplos projetos, visualizando informações de cada um de forma detalhada. Também é permitido visualizar o processo de medição de projetos concluídos e analisar os indicadores destes projetos. As avaliações são importantes insumos para o registro de uma base histórica organizacional e orientação em futuros projetos.

10.2.4. Definição de Objetivos, Necessidades de Informação, Indicadores e Medidas

Para este trabalho entende-se como [Barcellos, 2009]: necessidade de informação, necessária ao gerenciamento de objetivos, riscos e problemas; objetivo, é a meta pela qual ações de medição e estratégicas são planejadas/realizadas; indicador, é uma representação de forma simples ou intuitiva de uma medida para facilitar sua interpretação quando comparada a uma referência ou alvo; medida, é um instrumento de medição que é utilizado para associar um valor a um elemento mensurável ou pode ser a quantificação de dados em um padrão e qualidade aceitáveis (exatidão, completude, consistência, temporalidade).

Para o gerenciamento dos objetivos de medição, a ferramenta possibilita definir “o que é importante (necessário)” para a organização saber com o processo de Medição no sentido de atender os objetivos organizacionais. Posteriormente, o gerenciamento das necessidades de informações associadas aos objetivos é realizado a partir de uma folha de entrevista com a Alta Administração da organização com a finalidade de extrair e estruturar informações (como fatores de qualidade, fatores de variação) relevantes ao processo de Medição. Consiste, ainda, em levantar questões, as quais são perguntas e indagações que permitem facilitar o alcance dos objetivos de medição definidos.

A seguir, ocorre na ferramenta a definição e a aprovação dos indicadores para responder as necessidades de informação. Estes indicadores representam informações a partir das quais é possível avaliar uma situação e sua evolução histórica. Por fim, ocorre a caracterização das medidas para os indicadores, onde se define quais métricas serão usadas no processo de Medição e estas devem ser revisadas para verificar se estão de acordo com os objetivos de medição. Esta revisão pode ser realizada por um Gerente de Projetos, que tenha habilidades/experiência de coleta e análise de medidas de projetos e organizacionais.

Assim, a definição ou registro de novos objetivos, necessidades de informação, indicadores e medidas é realizada de forma detalhada, onde os campos de cadastro são definidos de acordo com os modelos estudados e armazenam as informações necessárias para que se tenha um bom processo de medição.

Os indicadores quando registrados devem indicar com qual necessidade de informação estão relacionados e por sua vez, cada necessidade de informação quando definida deverá indicar com qual objetivo está relacionada. As medidas são especificadas e coletadas, podendo ser usadas para responder o alcance dos limites propostos aos vários indicadores. Por sua vez, estes indicadores respondem à necessidade de informação, fornecendo assim dados para que se possa tomar a decisão mais assertiva e assim chegar ao objetivo estipulado.

10.2.5. Definição dos Procedimentos de Coleta e Análise

Esta funcionalidade detalha como será realizada a coleta sobre os dados das métricas fornecidas. O procedimento ajuda a assegurar que os dados corretos estão sendo coletados e a explicitar que as necessidades de informações e os objetivos das medições estão sendo atendidos, como pode ser visto na Figura 30.

O procedimento de coleta é definido para uma determinada medida, o qual armazena informações sobre como a coleta de medidas irá acontecer: como e quando a coleta será realizada; quem irá realizar a mesma; qual a frequência com que uma determinada coleta será realizada; qual o cálculo realizado para que uma medida seja gerada a partir dessa coleta; dentre outras informações.

Enquanto que o procedimento de análise é definido para um indicador, armazenando informações sobre como a análise de um indicador será realizada: como e quando a análise será realizada; qual o tipo de gráfico gerado que se adequa para uma análise mais clara do mesmo; quem irá analisar; quais as metas para cada estado do indicador (ok, alerta e crítico); como a análise deve ser feita; quais as ações devem ser tomadas, sabendo em qual estado encontra-se; dentre outros campos; como pode ser visto na Figura 31.

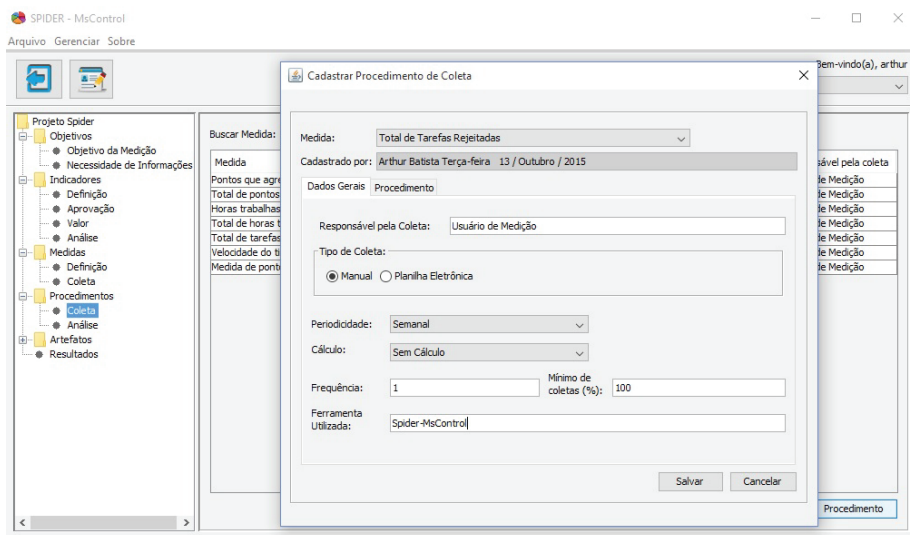


Figura 30. Função Procedimento de Coleta das Medidas

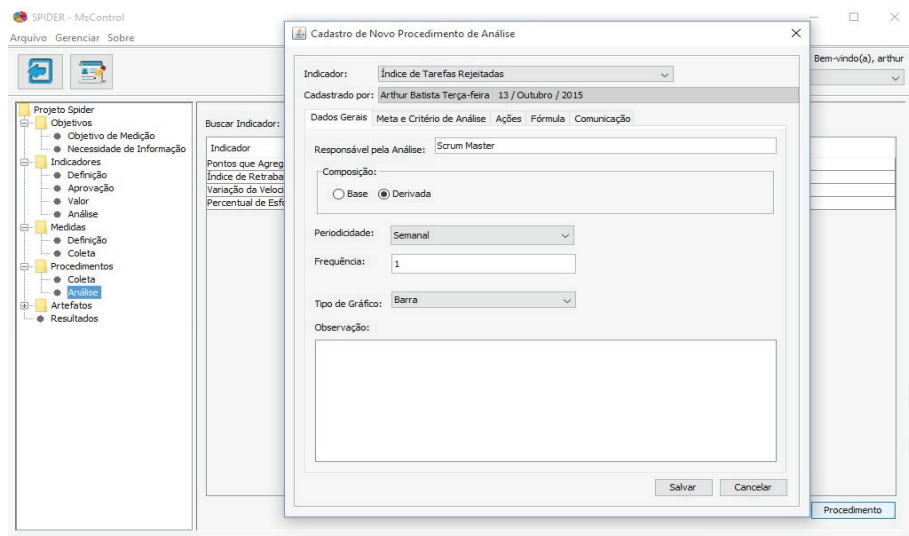


Figura 31. Função Procedimento de Análise dos Indicadores

Uma funcionalidade intrínseca a estes procedimentos diz respeito à definição do armazenamento das medidas, que consiste em especificar onde (o repositório) as métricas serão armazenadas. A definição de um repositório no contexto de medição ajuda a assegurar que os dados estarão disponíveis e acessíveis para uso futuro. O repositório deve ser definido em termos de localização, procedimentos de inserção e de acesso aos dados, incluindo permissões e responsabilidades. Assim, percebe-se que a própria ferramenta em si já sistematiza esta funcionalidade.

10.2.6. Coleta de Medidas e Análise de Indicadores

Esta funcionalidade é responsável pelo agrupamento e pela organização dos dados fornecidos pelos usuários, os quais serão usados nas medições, e pela análise dos dados coletados para que possíveis tomadas de decisões possam ser estabelecidas, como visto na Figura 32. A análise dos dados é acompanhada

por gráficos que facilitam a interpretação de uma coleção de dados, como visto na Figura 33.

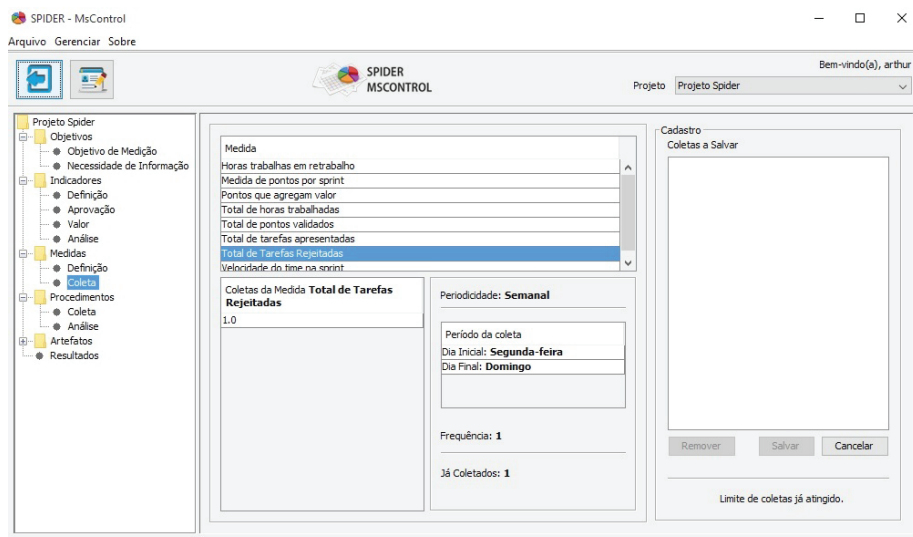


Figura 32. Função Coleta das Medidas

Importante enfatizar que a ferramenta Spider-MsControl possui integração a uma outra ferramenta de modelagem do processo chamada de Spider-PM (*Spider - Process Modeling*) [Barros, 2010], que provê um ferramental de apoio para a definição das atividades que serão medidas. Apesar desta integração, a Spider-MsControl pode ser usada de forma independente da Spider-PM.

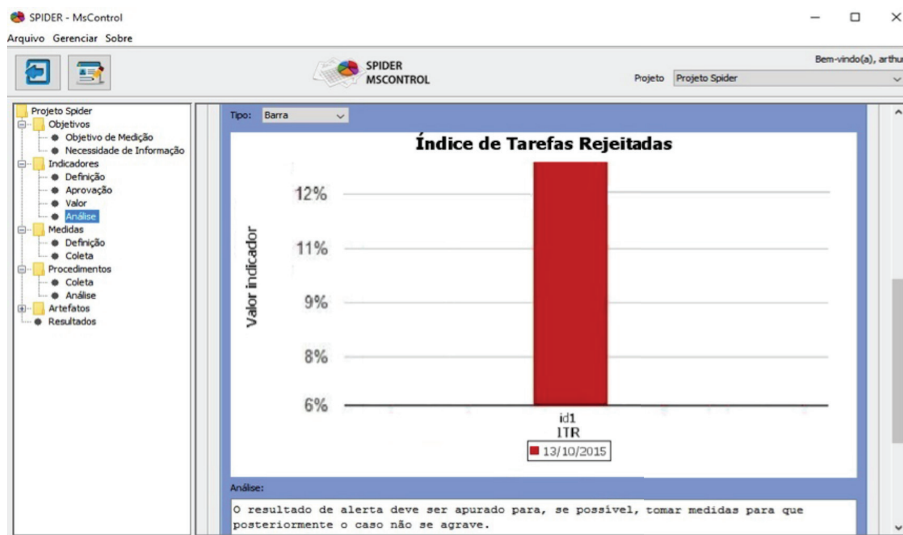


Figura 33. Função Análise dos Indicadores

Assim, a coleta das medidas e a análise dos indicadores são realizadas a partir do processo definido na ferramenta Spider-PM. Pelo fato deste processo não ser modelado dentro da ferramenta Spider-MsControl, esta característica permite uma flexibilidade para a organização na geração dos resultados de um processo de medição.

10.2.7. Resultados

Esta funcionalidade consiste em gerar relatórios a partir das análises realizadas sobre as métricas com o propósito de mostrar os resultados obtidos pelo processo de Medição. Depois de gerados, os relatórios de medição devem ser divulgados (comunicados) para os Gerentes de Projetos e/ou a Alta Administração, permitindo a avaliação deste relatório, a partir de críticas e sugestões.

Na etapa de resultado, a ferramenta disponibiliza um campo no qual o responsável irá analisar os gráficos gerados para interpretá-los e assim

definir uma tomada de decisão com base nos dados apresentados, como pode ser visto na Figura 34. Com isso chega-se ao objetivo de um processo de medição, que é obter informações suficientes para se tomar a decisão assertiva possível.

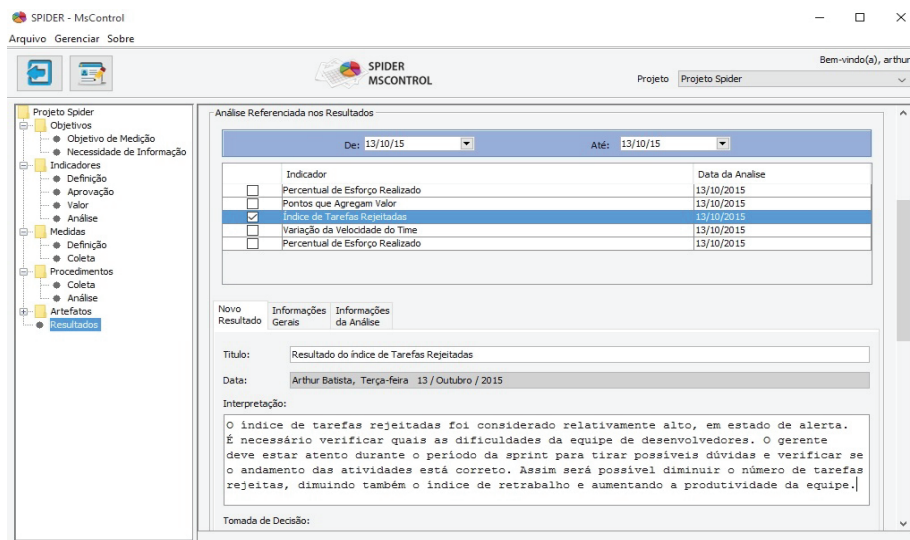


Figura 34. Função Resultados

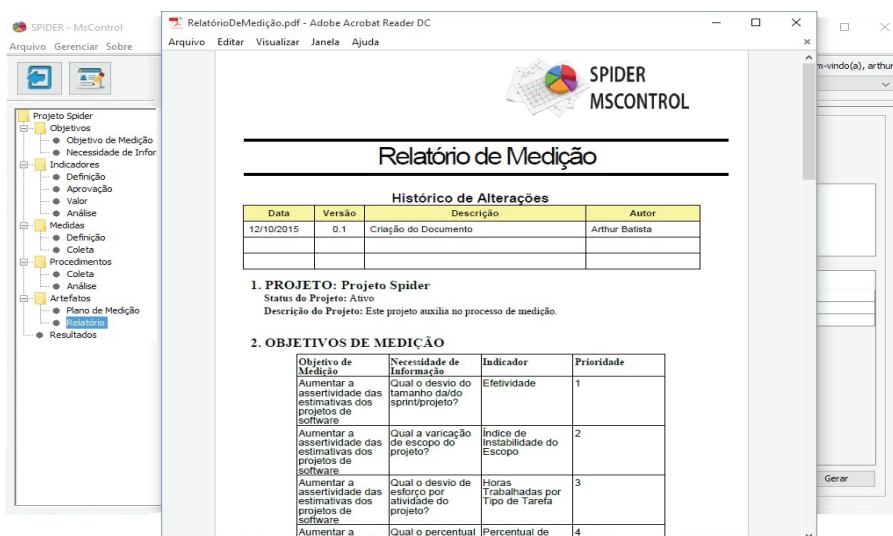
10.2.8. Plano de Medição e Relatório

A ferramenta gera dois artefatos ou documentos bem definidos na extensão PDF, sendo estes o Plano de Medição e o Relatório, como visto na Figura 35. O Plano de Medição contém basicamente os procedimentos de coleta e de análise, ou seja, de que forma o projeto foi planejado para executar o processo de medição. Já o Relatório contém os dados quantitativos em si, juntamente com os gráficos gerados em cada análise e a interpretação dos mesmos, assim como as decisões tomadas. A ferramenta Spider-MsControl possui a funcionalidade de permitir a análise dos indicadores gerados a partir da coleta dos dados, bem como o registro das ações tomadas como decisões,

quando de uma possível realização de uma Reunião de Análise Crítica com a Alta Administração.

Vale mencionar que a ferramenta permite com que artefatos com conteúdos parciais possam ser gerados, ou seja, os documentos podem ser gerados mesmo que nem todos os campos que o determinam tenham sido preenchidos.

Por fim, todos os registros devem ser armazenados, consistindo em mantê-los no repositório de medição com a finalidade de organizar todo o produto de trabalho produzido pelo processo de Medição.



The screenshot shows the Spider-MsControl application interface. On the left is a tree view of the project structure. The main window displays the 'Relatório de Medição' (Measurement Report) for 'Projeto Spider'. The report includes a table of changes, project status, and a table of measurement objectives.

Projeto Spider

- Objetivos
 - Objetivo de Medição
 - Necessidade de Informação
- Indicadores
 - Definição
 - Aprovação
 - Valor
 - Análise
- Medidas
 - Definição
 - Coleta
- Procedimentos
 - Coleta
 - Análise
- Artefatos
 - Plano de Medição
 - Relatório
 - Resultados

Relatório de Medição

Histórico de Alterações

Data	Versão	Descrição	Autor
12/10/2015	0.1	Criação do Documento	Arthur Batista

1. PROJETO: Projeto Spider
 Status do Projeto: Ativo
 Descrição do Projeto: Este projeto auxilia no processo de medição.

2. OBJETIVOS DE MEDIÇÃO

Objetivo de Medição	Necessidade de Informação	Indicador	Prioridade
Aumentar a assertividade das estimativas dos projetos de software	Qual o desvio do tamanho da/ds sprint/projeto?	Efetividade	1
Aumentar a assertividade das estimativas dos projetos de software	Qual a variação de escopo do projeto?	Índice de instabilidade do Escopo	2
Aumentar a assertividade das estimativas dos projetos de software	Qual o desvio de esforço por atividade do projeto?	Horas Trabalhadas por Tipo de Tarefa	3
Aumentar a assertividade das estimativas dos projetos de software	Qual o percentual de acerto do planejamento?	Percentual de Esforço Realizado	4

Figura 35. Função Artefatos de Medição

10.2.9. Ambiente de Implementação

A ferramenta Spider-MsControl foi implementada utilizando a linguagem de programação Java, sob a licença GPL – *General Public License*, voltada especificamente para o processo de Medição de Software, aderente às boas práticas recomendadas pelos padrões MR-MPS-SW, CMMI-DEV e ISO/IEC 12207, e

pela abordagem GQIM. Trata-se de um ambiente desktop que utiliza o sistema cliente-servidor e seu desenvolvimento foi pautado no uso de ferramentas de software livre, tais como: Sistema Operacional Ubuntu 14.10, Netbeans IDE 8.0.2, banco de dados MySQL 5.6.24 tanto para persistência de dados quanto para comunicação entre o servidor local e a ferramenta.

10.3. Metodologia de Implementação do Processo de Medição usando Ferramenta

Esta seção apresenta a metodologia proposta para implementação sistematizada do processo de MED do MR-MPS-SW e da área de processo MA do CMMI-DEV, por meio do uso da ferramenta Spider-MsControl (disponível em <http://www.spider.ufpa.br>, juntamente com o seu manual de usuário). Neste trabalho entende-se metodologia como sendo a codificação de um conjunto de práticas recomendadas, às vezes acompanhada de material de treinamento, programas de educação formal, planilhas, diagramas, tomando como parte um método (processo com uma série de passos, para construir um software) [Pressman, 2006]. Esta metodologia procura agregar boas práticas para o uso de ferramentas livres de apoio ao processo de MED do MR-MPS-SW e à área de process MA do CMMI-DEV.

10.3.1. Definindo Necessidades e Objetivos de Medição

Considerando um ambiente onde está devidamente instalada e configurada a ferramenta de apoio ao processo de Medição, a Spider-MsControl, a metodologia inicia com a definição do *abstraction sheet*, uma folha de entrevista com a Alta Administração da Organização com a finalidade de extrair, estruturar informações, como fatores de qualidade, fatores de variação, relevantes ao processo de Medição. Desta forma, informa-se: (1) o foco na qualidade; (2)

os fatores de variação; (3) a hipótese de *baseline*; e (4) o impacto na hipótese de *baseline*. Por fim, deve-se definir o objetivo da medição, que estabelece o(s) objetivo(s) bem definido(s) para o processo de Medição, feito a partir da análise das necessidades de informação e do *Abstraction Sheet* elicitados, devendo ser estabelecido: (1) o objetivo; (2) o propósito; (3) o foco; (4) o ambiente e o ponto de vista.

Tendo o objetivo organizacional definido, é necessário, posteriormente, a definição da necessidade de informação, que consiste em dizer “o que é importante/necessário” para a Organização saber com o processo de Medição, no sentido de atender os objetivos organizacionais. Assim, deve-se informar: (1) o perfil organizacional; (2) o usuário responsável pelo levantamento da necessidade de informação; (3) a necessidade de informação; (4) a prioridade e o tipo de derivação. Deste modo, será criada a meta de negócio da organização.

10.3.2. Identificando Medidas

Após a definição da necessidade de informação e do objetivo da medição, é necessário definir os indicadores, seguindo o paradigma GQIM proposto pelos modelos de qualidade, que estabelece as metas que permitem facilitar o alcance dos objetivos de medição definidos. Estes indicadores são compostos por medidas que serão coletadas e posteriormente analisados.

Criado o indicador, é necessário definir a medida a ser identificada, que retrata a métrica que será utilizada para conseguir responder as necessidades de informação levantadas, informando: (1) o nome e a descrição; (2) o mnemônico e as instruções; (3) a prioridade; (4) a escala; (5) a atomicidade; (6) a faixa e o valor base. Posteriormente, deve-se analisar a composição da medida, podendo ser básica ou derivada. Básica não deriva de outra medida, já a derivada depende de outra medida. Caso seja derivada, poderá ser configurada a fórmula para o cálculo da medida. Seguindo o fluxo, a medida

definida deverá ser aprovada, mediante a uma data e uma observação pela equipe responsável pela Medição, para estar disponível para uso ao longo de um processo de Medição.

10.3.3. Especificando Procedimentos de Coleta

Com a aprovação da medida a ser coletada, precisa-se especificar o procedimento de coleta e a forma de armazenamento, onde deve-se registrar: (1) a medida e a versão a ser usada; (2) o responsável pela coleta; (3) o momento e a frequência; (4) a periodicidade; (5) as instruções de coleta; (6) o tipo de coleta e a ferramenta utilizada. Com o procedimento definido, identifica-se a forma de armazenamento, caracterizando: (1) o local de armazenamento; e (2) o responsável pela preservação. Por fim, deve-se definir a verificação integridade dos dados, fornecendo: (1) a ferramenta de verificação; e (2) o nome do *checklist* a ser usado para esta verificação.

10.3.4. Definindo Procedimento de Análise

Tendo o procedimento de coleta definido, parte-se para a definição do procedimento de análise, identificando-se: (1) o indicador e a versão a ser analisado; (2) as instruções de análise; (3) o responsável; (4) a frequência e a periodicidade; (5) a forma de apresentação dos resultados; (6) os critérios da análise; e (7) a orientação para a tomada de decisão. Procura-se identificar a forma de comunicação dessa análise, ou seja, a forma como será informado aos interessados o resultado da análise, identificando-se a frequência, a periodicidade e o meio desta comunicação. Ao final desta prática, pode-se gerar o Plano de Medição que consolida informações organizacionais sobre a coleta das medidas e a análise dos indicadores. Esta evidência pode comprovar a aplicação prática do processo de Medição.

10.3.5. Coletando as Medidas e Analisando os Indicadores

A coleta inicia-se a partir da seleção da medida a qual será realizada a coleta, onde a ferramenta já registra a versão e a data/hora da coleta, e exibe: (a) a composição; e (b) o tipo de coleta; informações que foram definidas no procedimento de coleta, além de possibilitar um espaço para incluir observações sobre a coleta realizada. Caso o procedimento de coleta tenha sido definido como manual, a ferramenta desabilita a função importar dados, fazendo-se uma coleta manual por meio da opção de entradas de dados. Entretanto, se a forma de coleta definida for a automática, a opção importar dados é habilitada, bastando selecioná-la e importar os dados a partir de uma planilha eletrônica.

Realizada a coleta, pode-se analisar os dados gerados. A análise mostra: (1) a medida e a sua versão; (2) o indicador; (3) o critério de decisão; e (4) os dados coletados. Pode-se observar que fica registrada a data/hora da coleta, assim como o autor. Como instrumento para a análise a ferramenta gera gráficos, onde é necessário informar o período (data/hora) da geração da coleta, para que a ferramenta saiba quais dados capturar do repositório e consolidar na forma gráfica. Esse gráfico pode ser assumir várias formas, de acordo com o procedimento de análise.

10.3.6. Armazenando e Acompanhando Registro de Medição

A ferramenta Spider-MsControl registra em um banco de dados todas as informações da coleta, que podem ser acompanhadas pela operação histórico, que traz todas as informações já cadastradas, como a medida registrada, as coletas realizadas para cada medida e os valores destas coletas. Esta prática favorece a manutenção de um repositório de medidas organizacionais e a realização de uma auditoria contínua neste.

10.3.7. Gerando, Revisando e Apreciando Resultado de Medida

Ao final do processo de Medição, tem-se a necessidade de gerar e revisar os resultados das medições. Durante as revisões da medição, problemas podem ser identificados, analisados e registrados para uma posterior ação corretiva. A ferramenta dispõe de uma função “Resultado do Processo de Medição” que permite essa revisão e apreciação dos resultados pelos interessados no processo de medição.

Ao final pode-se gerar a evidência Relatório de Medição, que consolida informações gerais sobre o histórico de versão das medidas, as coletas e as análises realizadas. Esta prática pode favorecer a comprovação da aplicação prática do processo de Medição.

10.3.8. Aderência da metodologia Proposta ao MR-MPS-SW e CMMI-DEV

A análise de aderência entre a metodologia proposta aos modelos MR-MPS-SW e CMMI-DEV é feita pelo mapeamento das práticas sugeridas nas seções anteriores com os resultados esperados do processo MED constantes no guia geral do MR-MPS-SW [SOFTEX, 2012a] e as práticas específicas da área de processo MA sugeridas pelo CMMI-DEV [SEI, 2010a]. É importante ressaltar que a metodologia restringe-se a apresentar boas práticas para que seja possível contemplar os resultados esperados e práticas específicas do processo de Medição em conjunto com práticas definidas no processo de software da organização, como pode ser visto na Tabela 12. Resultado Esperado é “um resultado observável do sucesso do alcance do propósito do processo” [ISO/IEC, 1995] e Prática Específica “é a descrição de uma atividade considerada importante para alcançar uma meta específica associada” [SEI, 2010a].

Tabela 12. Mapeamento da Metodologia com os Resultados Esperados e Práticas Específicas de Medição do MR-MPS-SW e CMMI-DEV [Carneiro e Oliveira, 2011]

MR-MPS-SW	CMMI-DEV	Ferramenta(s) de Apoio	Funcionalidade(s) das Ferramentas
MED1	SP 1.1	Spider-MsControl	Definir Necessidade de Informações Elaborar <i>Abstraction Sheet</i> Definir Objetivos de Medição
MED2	SP 1.2	Spider-MsControl	Definir Indicadores Definir Medidas da Organização Alocar Objetivos de Medição ao Projeto Validar Medidas
MED3	SP 1.3	Spider-MsControl	Definir Procedimento de Coleta Definir Armazenamento de Medidas
MED4	SP 1.4	Spider-MsControl	Definir Procedimento de Análise
MED5	SP 2.1 SP 2.2	Spider-MsControl	Alimentar Repositório de Medidas Coletar Medidas Analisar Indicadores
MED6	SP 2.3	Spider-MsControl	Armazenar Registro, Ata de Reunião, Resoluções de Proposições
MED7	SP 2.4	Spider-MsControl	Gerar/Revisar Relatório com Resultados da Medição Divulgar Relatórios com Resultados da Medição Apreciar Relatórios com Resultados da Medição



PARTE III

A IMPLEMENTAÇÃO DOS PROCESSOS DO NÍVEL DEFINIDO

Sandro Ronaldo Bezerra Oliveira
Alexandre Marcos Lins de Vasconcelos

Segundo o SEI (2010a), no nível de maturidade Definido, os processos são bem caracterizados e entendidos, e são descritos em padrões, procedimentos, ferramentas e métodos. O conjunto de processos-padrão da organização, que é a base para o nível de maturidade Definido, é estabelecido e melhorado ao longo do tempo. Estes processos-padrão são utilizados para estabelecer uniformidade no contexto da organização. Os projetos estabelecem seus processos definidos ao adaptar o conjunto de processos-padrão da organização de acordo com as diretrizes para adaptação.

Uma distinção importante entre os níveis de maturidade Gerenciado e Definido é o escopo de padrões, descrições de processo e procedimentos. No nível de maturidade Gerenciado, os padrões, as descrições de processo e os procedimentos podem ser diferentes em cada instância específica do processo (por exemplo, em um projeto específico). No nível de maturidade Definido, os padrões, as descrições de processo e os procedimentos para um projeto são adaptados a partir do conjunto de processos-padrão da organização para se ajustar a um projeto específico ou uma unidade organizacional e, portanto, mais homogêneos, exceto por diferenças permitidas pelas diretrizes para adaptação.

Outra distinção importante é que no nível de maturidade Definido, os processos são geralmente descritos de forma mais rigorosa que no nível de maturidade Gerenciado. Um processo definido estabelece claramente o objetivo, as entradas, os critérios de entrada, as atividades, os papéis, as medidas, etapas de verificação, saídas e os critérios de saída. No nível de maturidade Definido, os processos são gerenciados mais pró-ativamente, com base na compreensão de como as atividades de processo relacionam-se e nas medidas detalhadas do processo, seus produtos de trabalho e serviços.

No nível de maturidade Definido, a organização deve amadurecer mais os processos do nível de maturidade Gerenciado. Assim, esta parte do livro fornecerá:

- orientações para a implementação dos processos de Definição do Processo Organizacional (Capítulo 11), Avaliação e Melhoria do Processo Organizacional (Capítulo 12) e Gerência de Reutilização (Capítulo 13);
- recomendações para a implementação dos processo de Desenvolvimento de Requisitos (Capítulo 14), Projeto e Construção do Produto (Capítulo 15) e Verificação e Validação (Capítulo 16);
- sugestões para a implementação do processo de Gerência de Riscos (Capítulo 17).



Capítulo 11

Implementação do Processo de Definição do Processo Organizacional

em [Barros e Oliveira, 2010b], [Barros e Oliveira, 2010c]

Renan Sales Barros

Sandro Ronaldo Bezerra Oliveira

Uma das áreas que compõem a Ciência da Computação é a Engenharia de Software. Essa área lida com os elementos relacionados com a produção de software objetivando a concepção de mecanismos que permitam o aprimoramento do processo de produção de software ou, simplesmente, processo de software. Esse processo, que é o principal objeto de estudo da Engenharia de Software, pode ser definido como o conjunto de atividades que buscam a construção de um software que atenda a um conjunto de requisitos [Oliveira, 2006].

Um processo de software pode ser representado por meio de um modelo. Esse tipo de representação pode ser definido a partir de diagramas que apresentam os principais elementos de um processo de software e as relações que existem entre esses elementos. Além disso, sabe-se que a utilização de modelos de processos traz para uma organização um conjunto de vantagens, dentre elas pode-se citar [Kellner e Hansen, 1988]:

- permitir que o processo seja entendido mais facilmente;

- propiciar a identificação de elementos do processo que podem ser melhorados;
- permitir a reutilização dos processos;
- apoiar as áreas de gerência de processos da organização.

Essas vantagens indicam que a modelagem de processos é uma área de conhecimento significativa dentro da Engenharia de Software. Contudo, a elaboração de diversos programas de melhoria do processo de software, juntamente com os esforços para que esses programas sejam adotados, e a crescente preocupação com a qualidade dos softwares desenvolvidos também são fortes indicadores da elevada importância de métodos que contribuam para o aperfeiçoamento do processo de software como a modelagem de processos. A maioria desses programas exige que os processos estejam representados de algum modo, o que, mais uma vez, evidencia a relevância da modelagem de processos dado que ela estabelece um mecanismo de apoio ao aperfeiçoamento dos processos de software e também uma maneira de se representar esse processos.

Para a construção de modelos de processos, a utilização de uma linguagem para modelagem é necessária. Além disso, uma maneira de facilitar a utilização de uma linguagem para modelagem consiste no uso de uma ferramenta sistematizada que apoie a utilização dessa linguagem de modelagem. Existem diversas linguagens e ferramentas para modelagem, no entanto, a partir de uma análise dos mecanismos de modelagem mais representativos encontrados na literatura especializada [Oliveira, 2006], observou-se que nenhum desses mecanismos foi elaborado objetivando tornar mais prática a modelagem de processos. Essa constatação justifica-se pelo fato desses mecanismos serem concebidos a partir de projetos acadêmicos, alguns não levando em consideração boas práticas obtidas com base na experiência em desenvolvimento de projetos pela indústria de software. Assim, a Spider-PM [Barros e Oliveira, 2010c] foi concebida com o foco de incorporar as principais práticas adotadas pelas organizações

para modelagem de processos e simultaneamente permitir a construção de modelos de processo consistentes com quaisquer processos adotados pela indústria.

Uma dessas práticas da indústria de software é a utilização das notações do *Software & Systems Process Engineering Meta-Model* (SPEM). O SPEM é o padrão da *Object Management Group* (OMG) para modelagem de processos e foi definido em 2002. O SPEM utiliza as notações da *Unified Modeling Language* (UML). O SPEM define um conjunto de estereótipos específicos para a modelagem de processos. Esses estereótipos são utilizados com as demais notações da UML de acordo com regras determinadas na especificação do SPEM [OMG, 2008].

No entanto, a partir de observações realizadas durante a implantação de programas de melhoria de processos organizacionais, observou-se que apenas um subconjunto dos estereótipos e diagramas disponibilizados pelo SPEM é utilizado pela maioria das organizações. Dessa forma, optou-se pela definição de um perfil do SPEM, que foi adotado como linguagem de modelagem da Spider-PM, denominado SPIDER_ML.

A SPIDER_ML [Barros e Oliveira, 2010b] foi elaborada para: incorporar e formalizar práticas de modelagem de processos; e para limitar a quantidade de estereótipos e diagramas utilizados na modelagem de processos e assim torná-la mais simples. Dessa forma, a utilização da Spider-PM e da SPIDER_ML tem como principal vantagem a simplificação da modelagem de processos, dado: o refinamento do conjunto de notações e estereótipos que compõem o SPEM; e a modelagem de processos de software aderente às recomendações de programas de melhoria da qualidade do processo de software, como o programa MPS.BR [SOFTEX, 2012a]. Além disso, a Spider-PM também deve possibilitar a produção de modelos totalmente compatíveis com o SPEM. Deve-se ressaltar que a Spider-PM encontra-se distribuída livremente sem nenhum custo e sem nenhuma limitação de uso, dessa forma, a utilização da Spider-PM deve ser menos onerosa para uma

organização se comparada com soluções que incluem a utilização de softwares proprietários.

Assim, o desenvolvimento da Spider-PM justifica-se pela carência de um mecanismo de modelagem de processos gratuito, mais prático, compatível com programas de melhoria de processos organizacionais e que produza modelos de processos fiéis aos processos adotados pela indústria de software.

11.1. A Linguagem SPIDER_ML

As abordagens para a modelagem baseadas na UML obtiveram uma maior aceitação pela indústria de software [Bendraou *et al.*, 2010]. Uma dessas abordagens é o SPEM. O SPEM foi utilizado nesse trabalho devido a um conjunto de fatores, como definido anteriormente. Por outro lado, frequentemente as organizações que adotam o SPEM para a construção de seus modelos de processos apenas utilizam um subconjunto das notações e diagramas disponibilizados pelo SPEM [Barros e Oliveira, 2010b e 2010c]. Além disso, muitas vezes, devido à complexidade da especificação do SPEM, algumas organizações utilizam essas notações e diagramas de maneira incompatível com essa especificação.

Visando solucionar esse problema, optou-se pela definição de um perfil para utilização do SPEM, esse perfil, denominado SPIDER_ML, foi utilizado como linguagem para modelagem de processos de software da Spider-PM, ou seja, a SPIDER_ML é uma linguagem para modelagem definida a partir das notações do SPEM e, consequentemente, da UML. As subseções a seguir apresentam as principais características e objetivos da SPIDER_ML. Além disso, também são mostrados os componentes, a estrutura e os diagramas presentes na SPIDER_ML.

11.1.1. Objetivos e Características da SPIDER_ML

Existem diversas linguagens para modelagem de processos de software. A maioria dessas linguagens foi concebida a partir de estudos acadêmicos que desconsideram as práticas de modelagem de processos adotadas pela indústria de software [Barros e Oliveira, 2010b]. Diante desse cenário, as organizações que desenvolvem software acabam tendo que criar ou adaptar soluções para modelagem de processos de software. A partir de um levantamento e análise de diversas linguagens para modelagem realizados em [Oliveira, 2006] e com base em observações realizadas durante a implementação de programas da melhoria dos processos organizacionais obteve-se um conjunto de práticas frequentemente utilizado pela indústria de software.

Com base nesse conjunto de práticas foi concebida a SPIDER_ML. Essa linguagem apresenta como principais objetivos:

- incorporar e formalizar as práticas de modelagem de processos utilizadas pela indústria de software;
- refinar e reutilizar o conjunto de elementos do SPEM;
- tornar a modelagem de processos mais simples de ser realizada;
- ser uma linguagem de fácil e rápida assimilação;
- permitir a modelagem de quaisquer processos de software que sejam utilizados nas organizações.

Ao modelar o processo de software segundo critérios próprios e práticas não formalizadas, as organizações podem não usufruir de todos os benefícios da modelagem de processos. Por exemplo, um processo modelado dessa forma pode dificultar a comunicação, uma vez que as notações estão sujeitas à interpretação de cada indivíduo, já que as mesmas não se encontram formalizadas. Além disso, caso se utilize uma linguagem em desacordo com sua especificação oficial um profissional pode ficar confuso em determinar qual a real semântica dessa notação. Assim, a SPIDER_ML procura formalizar um

conjunto de práticas organizacionais para que as organizações produzam modelos de acordo com a especificação da linguagem e não precise adaptá-la.

Uma das práticas levantadas nessa observação foi a utilização das notações do SPEM. No entanto, essas notações não eram utilizadas conforme a especificação do SPEM. Por exemplo, o SPEM apresenta um grande conjunto de estereótipos que devem ser utilizados para indicar o real significado de uma associação entre dois elementos. Como pode ser visto na Figura 36, uma grande quantidade de estereótipos é utilizada na modelagem de uma única tarefa. Por outro lado, nota-se que nenhuma semântica é atribuída ao direcionamento da seta que representa uma associação entre dois elementos. Aproveitando-se disso, as organizações acabam utilizando esse direcionamento para indicar quando um produto de trabalho é uma entrada ou saída da tarefa, por exemplo. Essa utilização do direcionamento das associações para modificar a semântica dos relacionamentos foi outra prática adotada pela SPIDER_ML. Assim, na Figura 37 pode-se observar a mesma definição de tarefa modelada na Figura 36 de acordo com a especificação da SPIDER_ML. Deve-se ressaltar que, apesar das diferenças, ambos os diagramas apresentam a mesma semântica.

Ao realizar esse refinamento nas notações do SPEM, com o objetivo de incorporar práticas da indústria de software, a SPIDER_ML permite que a modelagem de processos seja realizada com um número menor de elementos se comparada com o SPEM. Dessa forma, pode-se considerar que a modelagem por meio da SPIDER_ML é mais simples, prática e rápida de ser realizada se comparada com a modelagem feita de acordo com a especificação oficial do SPEM [Barros e Oliveira, 2010b].

Aliando essa redução de elementos necessários para a modelagem com a utilização de notações da SPEM e da UML mais o uso de práticas frequentemente adotadas pela indústria de software, a SPIDER_ML torna-se uma linguagem simples e rápida de ser aprendida devido ao reaproveitamento de conhecimentos comuns aos profissionais envolvidos com o processo de software.

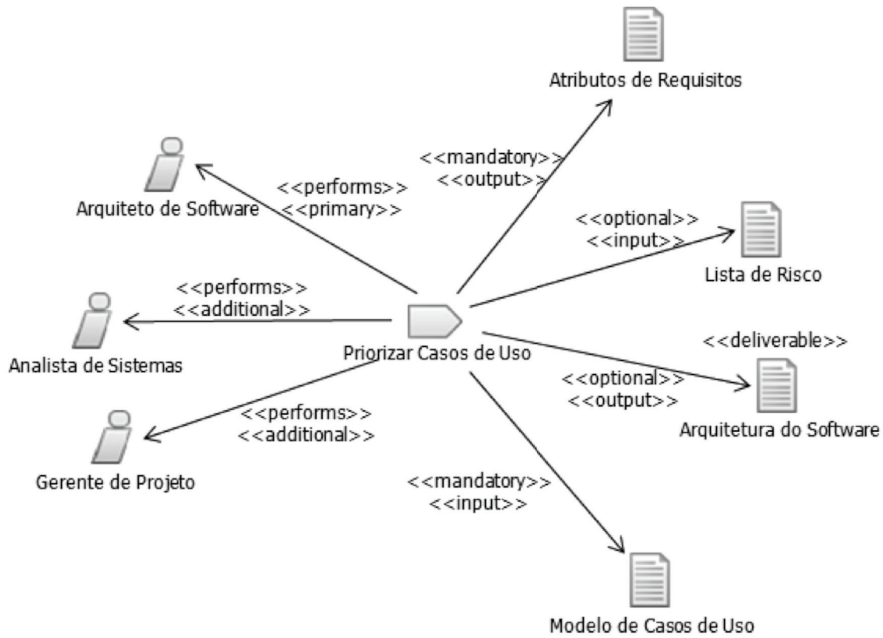


Figura 36. Exemplo de definição de tarefa de acordo com a especificação do SPERM [OMG, 2008].

Contudo, mesmo reduzindo o número de notações necessárias para a modelagem de processos, a SPIDER_ML ainda apresenta expressividade suficiente para modelar qualquer processo de software a um nível de detalhe suficiente para que o mesmo possa ser executado em um projeto. A SPIDER_ML foi concebida como uma linguagem com um pequeno conjunto de elementos básicos mais um mecanismo para extensão do mesmo. Dessa forma, mesmo que alguma singularidade de algum processo de software não tenha sido inicialmente prevista na especificação da SPIDER_ML, a mesma oferece mecanismos que permitem a modelagem desse tipo de processo. Basicamente, a SPIDER_ML apresenta um conjunto com os elementos básicos de um processo de software que podem ser estendidos por meio de estereótipos, ou seja, a SPIDER_ML encontra-se de acordo com a primeira abordagem para linguagens para modelagem, as linguagens com um pequeno número de elementos estendíveis.

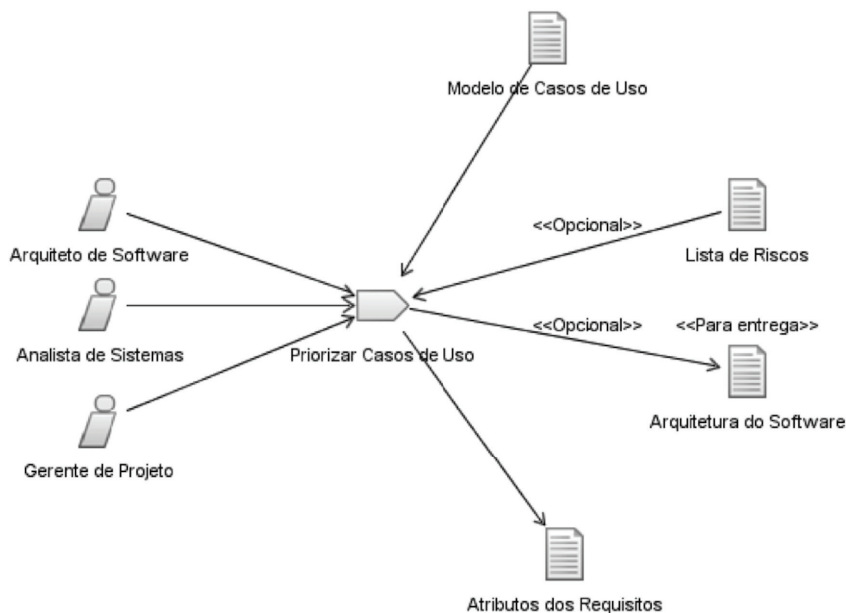












Figura 37. Exemplo de definição de tarefa de acordo com a especificação da SPIDER_ML.



Com relação à compatibilidade da SPIDER_ML com as fases do meta-processo definidas por Conradi e Liu (1994), tem-se que a SPIDER_ML atinge os objetivos descritos pelas fases 1, 3 e 4. Todas as demais fases estão relacionadas com simulação ou execução do processo de software. No entanto, assim como o SPEM, a SPIDER_ML não oferece mecanismos nativos para simulação e execução automatizada do processo de software. Por outro lado, os modelos de processos produzidos de acordo com a SPIDER_ML podem ser facilmente mapeados para uma ferramenta de gestão de projetos, por exemplo, onde o processo pode ser executado e acompanhado. Outra alternativa, também possível para o SPEM, é o mapeamento do processo modelado para um formalismo de execução como a *Business Process Execution Language* (BPEL). De qualquer forma, todas esses aspectos de execução estão mais relacionados com uma ferramenta que implemente a SPIDER_ML do que com as definições que a linguagem apresenta. De qualquer forma,

ressalta-se que a versão da Spider-PM referenciada nesse trabalho apoia apenas a modelagem do processo de software. No entanto, trabalhos são realizados visando a construção de mecanismos que possibilitem a simulação e a execução dos processos modelados de acordo com a SPIDER_ML na Spider-PM.

Apesar dessa limitação com relação à simulação e execução, ao se fazer uma avaliação da SPIDER_ML, considerando os critérios para comparação das linguagens para modelagem definidos por Bendraou *et al.* (2010), Oliveira (2006) e Zamli e Lee (2001), observa-se que a SPIDER_ML apresenta diversos outros recursos como pode ser observado na Tabela 13.

Tabela 13. Avaliação da SPIDER_ML











	SPIDER_ML
A linguagem é procedimental?	
A linguagem é orientada a redes?	
A linguagem é orientada a objetos?	
A linguagem é baseada em regras?	
A linguagem é formal?	
A linguagem apresenta notações visuais?	
A linguagem suporta meta-processos?	
A linguagem foca em processos de software?	
A linguagem é executável?	
A linguagem é suportada por ferramentas de modelagem?	








Legenda	
	Sim
	Não

11.1.2. Componentes da SPIDER_ML

Ao todo, a SPIDER_ML conta com 17 elementos para a modelagem de processos de software. Na Tabela 14 são listados todos esses elementos, seus respectivos ícones e a descrição de cada um desses elementos.

Tabela 14. Elementos da SPIDER_ML

Ícone	Nome	Descrição
	Processo Padrão	O conjunto de todos os ativos do processo de software que são reutilizáveis
	Disciplina	Um conjunto de definições de tarefa que contribuem para o atendimento de um mesmo objetivo do Processo Padrão.
	Definição de Tarefa	Um trabalho reutilizável e que não pode ser decomposto.
	Definição de Papel	Um conjunto de capacidades desejadas para um determinado papel que pode ser reutilizado em diferentes Processos Instanciados.
	Definição de Produto de Trabalho	Um conjunto de características desejadas para um produto de trabalho que pode ser reutilizado em diferentes Processos Instanciados.
	Definição de Ferramenta	Um conjunto de características desejadas para uma ferramenta que pode ser reutilizada em diferentes Processos Instanciados.
	Procedimento	Representa uma conduta que deve ser adotada durante a realização de uma tarefa do Processo Padrão ou de um Processo Instanciado.
	Conjunto de Processos Instanciados	O conjunto de Processos Instanciados a partir de um mesmo Processo Padrão.
	Processo Instanciado	Uma instância do Processo Padrão para um projeto específico.
	Fase	Um período significativo de um Processo Instanciado.

Ícone	Nome	Descrição
	Iteração	Um conjunto de atividades e tarefas que serão repetidas dentro de uma fase.
	Marco	Um evento significativo em um Processo Instanciado.
	Atividade	Um trabalho que deve ser realizado no decorrer de um Processo Instanciado e que pode ser decomposto em outras atividades ou tarefas.
	Tarefa Instanciada	Um trabalho que deve ser realizado no decorrer de um Processo Instanciado e que não pode ser decomposto.
	Papel Instanciado	Um papel desempenhado por um recurso no Processo Instanciado.
	Produto de Trabalho Instanciado	Um produto de trabalho consumido, gerado ou modificado em um Processo Instanciado.
	Ferramenta Instanciada	Uma ferramenta que deve ser utilizada durante a realização de uma tarefa em um Processo Instanciado.

O elemento Ferramenta Instanciada não é um elemento original do SPEM. No entanto, foi identificada a necessidade de um componente que apresentasse essa semântica para a construção de modelos de processos mais consistentes. Assim, a SPIDER_ML apresenta um único componente não oriundo da especificação do SPEM. Por outro lado, devido ao fato do SPEM possibilitar o uso de estereótipos UML para a extensão da semântica de seus elementos, a utilização do elemento Ferramenta Instanciada não consiste numa violação da especificação do SPEM.

Além dos elementos apresentados na Tabela 14, a SPIDER_ML conta com 7 relacionamentos da UML. Esses relacionamentos são: associação, transição, dependência, agregação, composição, generalização e âncora de nota. A semântica de cada um desses tipos de relacionamento permanece a mesma da

UML. Deve-se ressaltar que, para um melhor entendimento sobre a SPIDER_ML deve-se avaliar a especificação dessa linguagem.

11.1.3. Estrutura da SPIDER_ML

A SPIDER_ML apresenta uma estrutura básica de organização compatível o modelo para definição de processo descrito nesse trabalho. Dessa forma, existem duas estruturas de processos definidas na SPIDER_ML. Uma para o Processo Padrão e outra para os Processos Instanciados.

A abordagem utilizada pela SPIDER_ML define o Processo Padrão como um conjunto de Disciplinas. Por sua vez, cada Disciplina consiste de um conjunto de Definições de Tarefas. Na Figura 38 tem-se que o elemento raiz da árvore é o Processo Padrão. Em seguida, o Processo Padrão é decomposto em uma série de elementos do tipo Disciplina. Cada um dos elementos do tipo Disciplina é constituído por um conjunto de elementos do tipo Definição de Tarefa. Todo Processo Padrão modelado com a SPIDER_ML apresenta essa estrutura.

O Conjunto de Processos Instanciados provenientes de um mesmo Processo Padrão é representado na SPIDER_ML pelo componente denominado Conjunto de Processos Instanciados. Esse componente é mostrado na Figura 39 como o elemento que se encontra no topo da estrutura hierárquica. Os elementos imediatamente inferiores ao Conjunto de Processos Instanciados são elementos do tipo Processo Instanciado. Na abordagem utilizada pela SPIDER_ML, os elementos do tipo Processo Instanciado consistem de uma sequência Fases ordenadas. Essas Fases são representadas na Figura 39 como os elementos de menor hierarquia.

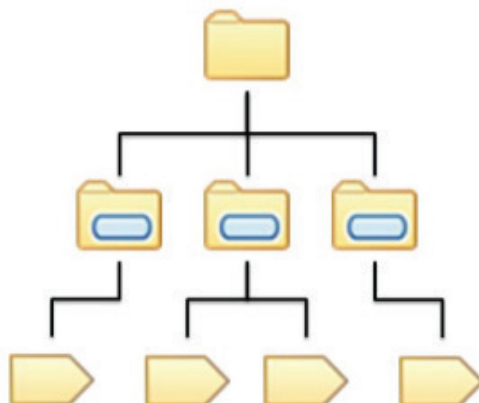


Figura 38. Estrutura do Processo Padrão

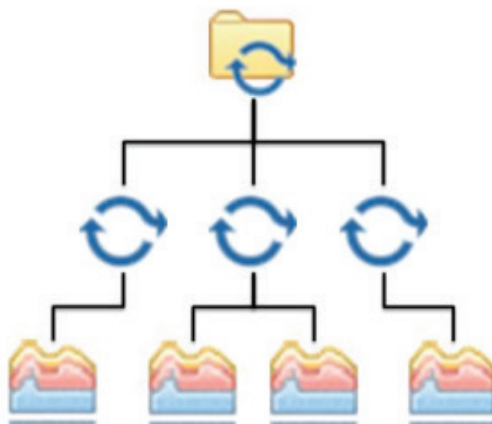


Figura 39. Estrutura do conjunto de Processos Instanciados

Uma fase pode apresentar dois tipos de ciclo de vida: sequencial e iterativo. Quando esse ciclo de vida é iterativo, uma Fase deve ser decomposta em um conjunto de componentes do tipo Iteração, como pode ser observado na Figura 40. Cada Iteração, por sua vez, pode ser decomposta em um conjunto que agrupa componentes do tipo Atividade e Tarefa Instanciada.

No caso de apresentar o ciclo de vida do tipo sequencial, a Fase deve ser definida da mesma maneira que o elemento Iteração, ou seja, ela é decomposta

em um conjunto de elementos do tipo Atividade e Tarefa Instanciada. A Figura 41 ilustra essa situação.

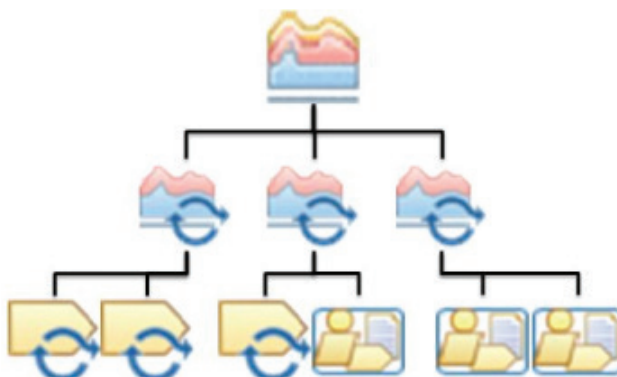


Figura 40. Estrutura de uma fase com ciclo de vida iterativo

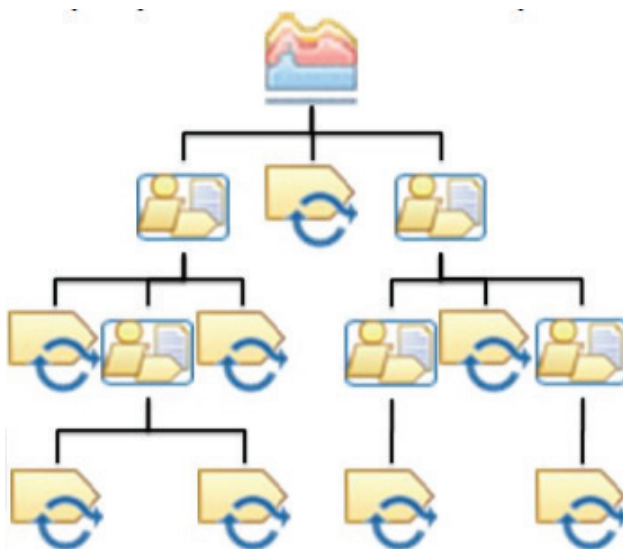


Figura 41. Estrutura de uma fase com ciclo de vida sequencial

A Tarefa Instanciada é o elemento que termina a estrutura hierárquica no contexto de Processos Instanciados uma vez que ela representa trabalhos mínimos que não podem ser decompostos. Os elementos do tipo

Atividade, por sua vez, representam trabalhos de um Processo Instanciado que podem ser decompostos. Assim, os elementos do tipo Atividade podem ser decompostos em um conjunto de outros elementos do tipo Atividade e elementos do tipo Tarefa Instanciada. Deste modo, uma Atividade é estruturada da mesma maneira que o elemento Iteração e o elemento Fase adotando o ciclo de vida sequencial.

11.1.4. Diagramas UML utilizados pela SPIDER_ML

Cada um dos elementos da SPIDER_ML é descrito por meio de um diagrama UML. Por exemplo, o elemento Processo Padrão é descrito por meio de um diagrama de pacotes que mostra quais as Disciplinas que o mesmo apresenta e quais as dependências entre essas Disciplinas. A Figura 42 apresenta um exemplo de diagrama que descreve o Processo Padrão.

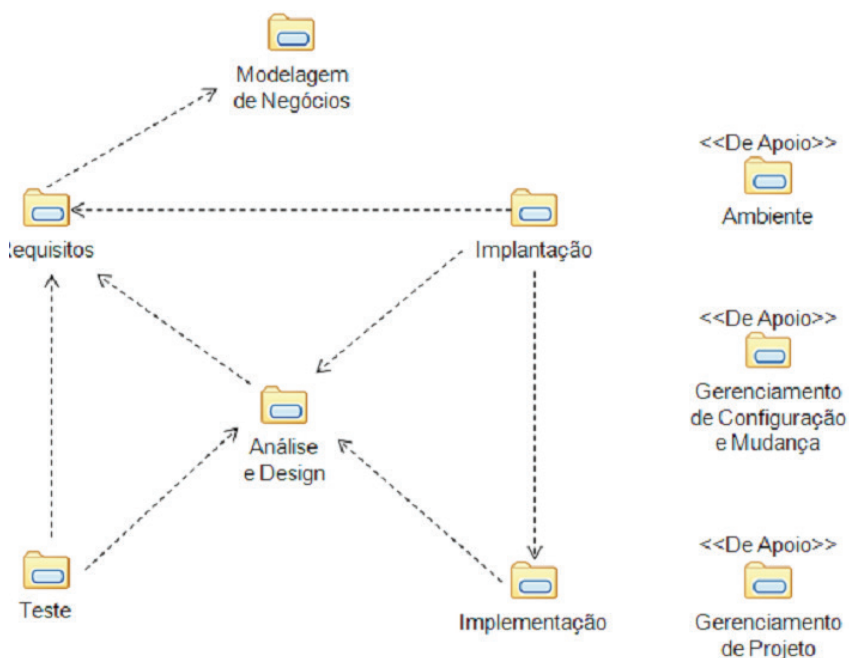


Figura 42. Exemplo de um diagrama de pacotes que descreve um Processo Padrão

Ao todo, a SPIDER_ML utiliza três diagramas da UML: diagrama de classes, diagrama de pacotes e diagrama de atividades.

O diagrama de pacotes expressa a organização interna de elementos que são definidos como um conjunto de outros elementos. Esse diagrama é utilizado para descrever o Processo Padrão as Disciplinas. A Figura 43 mostra uma Disciplina descrita por meio de um diagrama de pacotes.

O diagrama de classes mostra as associações existentes entre os elementos. O diagrama de classes é utilizado para descrever os seguintes componentes: Definição de Tarefa, Definição de Papel, Definição de Produto de Trabalho, Definição de Ferramenta, Procedimento, Tarefa Instanciada, Papel Instanciado Produto de Trabalho Instanciado e Ferramenta Instanciada. A Figura 44 mostra um diagrama de classes descrevendo uma Definição de Tarefa.

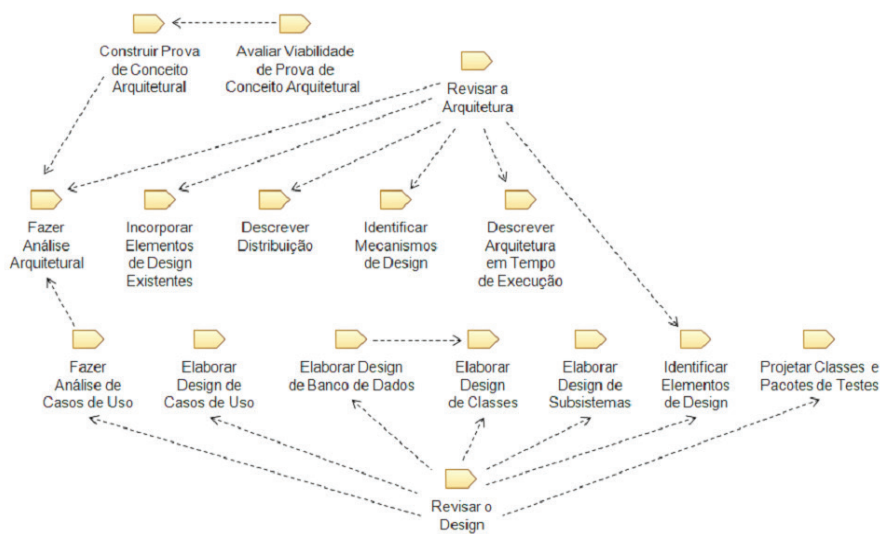


Figura 43. Exemplo de um diagrama de pacotes que descreve uma Disciplina

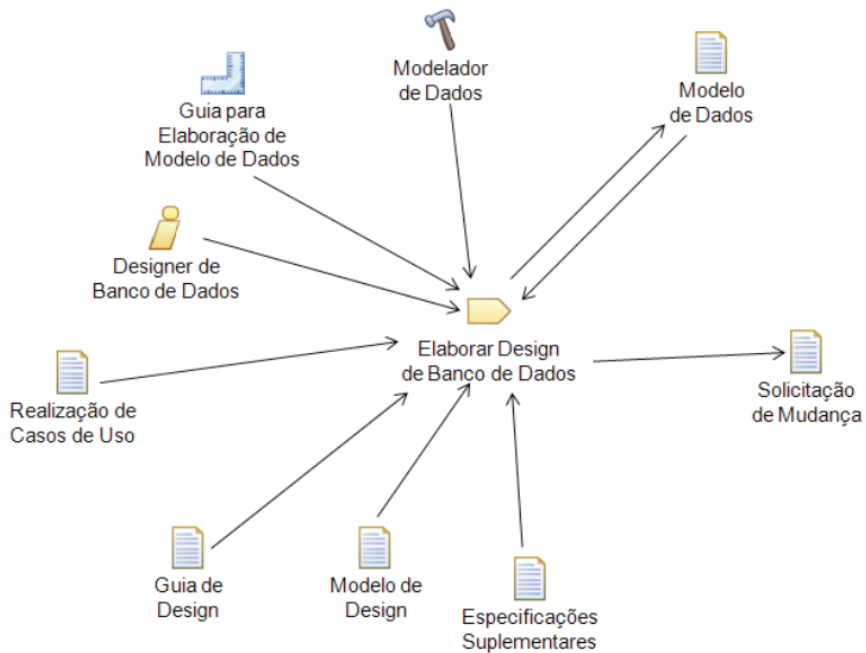


Figura 44. Exemplo de um diagrama de descrição de uma Definição de Tarefa

Por fim, o diagrama de atividades demonstra o fluxo de execução de um dado elemento. Devido esse fluxo estar diretamente relacionado com o ciclo de vida do projeto, esse diagrama apenas é utilizado no contexto de um Processo Instanciado. Os elementos que são descritos por meio de um diagrama de atividades são: Processo Instanciado, Fase, Interação, e Atividade. A Figura 45 mostra um diagrama de atividades que apresenta o fluxo de Atividades de uma Iteração.

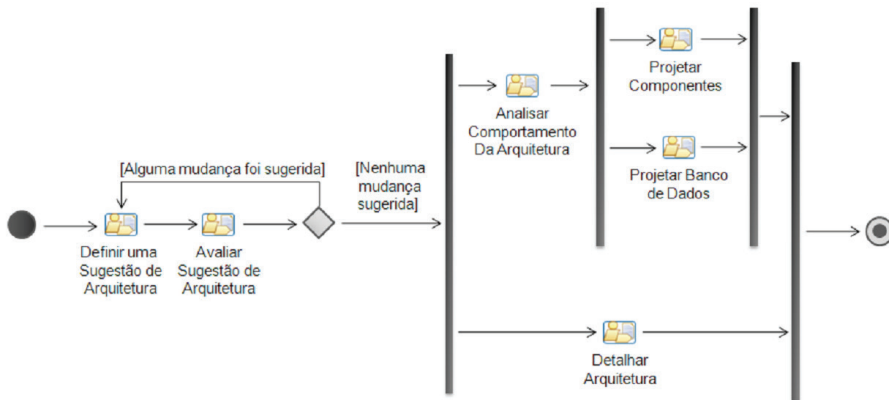


Figura 45. Exemplo de um diagrama de atividades que descreve uma Iteração

11.2. A Ferramenta Spider-PM

Existem diversas ferramentas que apoiam a modelagem de processos de software com base em uma das diversas linguagens para modelagem existentes. No entanto, a maioria dessas linguagens e ferramentas é concebida a partir de estudos acadêmicos que desconsideram aspectos práticos envolvidos na modelagem de processos de software. Nesse cenário, a Spider-PM foi concebida objetivando incorporar as principais práticas utilizadas pela indústria de software e, simultaneamente, permitir a construção de modelos de processos compatíveis com quaisquer processos de software que venham a ser utilizados pela indústria de software.

A Spider-PM é distribuída sob a licença *General Public License* (GPL) e foi desenvolvida utilizando-se apenas tecnologias livres. Dessa forma, não existe nenhuma restrição ou custo associado à utilização da Spider-PM. A Spider-PM e seu manual de utilização juntamente com a especificação da SPIDER_ML encontram-se disponíveis na página do Projeto SPIDER (www.spider.ufpa.br).

11.2.1. Características e Objetivos

A Spider-PM foi desenvolvida para apoiar a modelagem de processos de software suportando algumas das práticas adotadas pela indústria de software. Além disso, a Spider-PM também foi desenvolvida para permitir a modelagem de processos compatíveis com programas de melhoria como MPS.BR e o CMMI.

O principal objetivo da Spider-PM é permitir que as organizações se beneficiem mais facilmente das vantagens que podem ser obtidas com a modelagem de processos. Assim, para que de fato esses benefícios possam ser alcançados com facilidade, a Spider-PM, diferente da maioria das ferramentas de modelagem de processo disponíveis, foi produzida focando diretamente na indústria de software.

A partir de observações, constatou-se que muitas organizações utilizam as notações do SPEM para modelagem de processos pela vantagem de reutilizar as notações da UML, que são bastante familiares aos profissionais envolvidos com o processo de software. Por outro lado, a especificação da SPEM é relativamente complexa e, algumas vezes, exige um grande esforço caso se deseje produzir modelos de processo totalmente de acordo com tal especificação. Para contornar essa complexidade, muitas organizações acabam definindo perfis próprios de utilização da SPEM. Assim, para diminuir o esforço requerido para modelagem de processos, a linguagem para modelagem utilizada pela Spider-PM, que é a SPIDER_ML, incorpora algumas dessas características desses perfis. Por outro lado, a Spider-PM também permite a construção de diagramas totalmente de acordo com a especificação da SPEM 2.0, ou seja, a Spider-PM apesar de não focar diretamente na modelagem por meio do SPEM, pode ser utilizada como uma ferramenta para a construção de diagramas de acordo com o SPEM.

Dessa forma, visando atingir seu principal objetivo, que é facilitar a modelagem de processos de software, a Spider-PM apresenta um conjunto específico de características. A seguir tem-se uma lista com as principais características da Spider-PM:

- **Reutilização das notações da SPEM e da UML:** tal característica, como descrito, permite que muitos profissionais já familiarizados com essas notações possam aprender rapidamente como construir modelos de processo com a Spider-PM;
- **Interface gráfica consistente com a maioria dos softwares para modelagem UML:** muitos softwares para modelagem de processos apresentam uma interface complexa e pouco usual que muitas vezes exige que o usuário defina uma série de elementos para que posteriormente o mesmo utilize esses elementos em seus modelos. Diferente dessa abordagem, a Spider-PM apresenta uma interação semelhante às ferramentas para modelagem UML, onde os elementos são criados quando colocados em um diagrama;
- **Separação entre os elementos do processo que podem ser reutilizados e os que não podem:** a Spider-PM apresenta uma separação clara entre os elementos do processo de software que podem ser reutilizados em mais de um projeto e os que são únicos para um projeto. Essa separação é feita de acordo com a especificação do SPEM e de acordo com os programas de melhoria CMMI e MBS.BR;
- **Persistência dos processos modelados em arquivos XML:** muitas organizações utilizam sistemas automatizados de gerência de configuração, a maioria desses sistemas permite fazer comparações entre diferentes versões de arquivos XML. Assim, é possível manter e visualizar todas as mudanças feitas em um modelo de processo produzido pela Spider-PM;
- **Executa a partir da *Java Standard Runtime Environment 1.6* (JRE 1.6):** a Spider-PM pode ser facilmente inserida em diferentes ambientes de desenvolvimento, uma vez que ela pode ser executada em qualquer sistemas compatível com essa tecnologia;
- **Permite a extensão da semântica dos componentes para modelagem:** todo e qualquer elemento de modelagem da Spider-PM pode ter sua semântica estendida por meio dos estereótipos UML. Assim,

mesmo que um processo de software apresente características que não podem ser contempladas com elementos padrão da SPIDER_ML, é possível a definição de estereótipos UML de modo que essa característica seja contemplada;

- **Valida os diagramas enquanto eles estão sendo definidos:** essa característica impede que os usuários construam modelos de processos inconsistentes com a especificação da SPIDER_ML. A validação é realizada automaticamente durante a modelagem do processo e apenas reporta construções inválidas feitas pelos usuários.

A Tabela 15 mostra uma avaliação da Spider-PM com relação aos critérios utilizados na comparação de diferentes ferramentas de apoio à modelagem de processos de software com base no SPEM, apresentada em [Barros, 2010].

Tabela 15. Avaliação da Spider-PM

	Spider-PM
Plataforma:	Java 1.6
A plataforma é gratuita?	✓
A ferramenta é gratuita?	✓
Foca na modelagem por meio do SPEM?	✓
Utiliza as notações do SPEM 2.0?	✓
Permite o detalhamento de acordo com o tipo do elemento?	✓
Publica o processo modelado?	✓
Distingue os elementos reutilizáveis do processo?	✓
Valida o processo modelado?	✓
Permite a utilização de estereótipos?	✓

Legenda	
✓	Sim
✗	Não

A Spider-PM também deve ser integrada a um ambiente de software para simulação, execução, definição, avaliação e melhoria dos processos de software. Esse ambiente encontra-se em desenvolvimento pelo Projeto Spider e a Spider-PM é uma das ferramentas que fará parte desse ambiente. A Spider-PM tem um papel essencial nesse ambiente uma vez que a simulação, execução, definição, avaliação e melhoria dos processos de software é realizada a partir dos processos de software modelados na Spider-PM.

11.2.2. Aderência ao MPS.BR

A Spider-PM foi idealizada a partir de pesquisas realizadas pelo Projeto Spider (www.spider.ufpa.br) da Universidade Federal do Pará. Esse projeto visa o levantamento, customização e desenvolvimento de ferramentas de livres para apoiar a implementação do MPS.BR.

O MPS.BR é um programa que visa definir e aprimorar um modelo de melhoria e avaliação de processos de software mais adequado às características das organizações nacionais. Esse modelo apresenta sete níveis de maturidade para um processo de software. Os níveis de maturidade são estágios de melhoria e definem graus de evolução em um processo de software. Cada um desses níveis indica um grau de maturidade para o processo da organização. Em cada um desses níveis do MPS.BR podem existir conjuntos de atividades relacionadas entre si. No contexto do MPS.BR, esses conjuntos de atividades são chamados de processos. Cada um desses processos apresenta resultados esperados. Um resultado esperado consiste de um item que permite observar se uma meta do processo foi alcançada ou não.

A Spider-PM lida com os conhecimentos envolvidos nos seguintes processos do MPS.BR: Definição do Processo Organizacional (DFP) e Avaliação e Melhoria do Processo Organizacional (AMP). Com relação ao processo DFP, a Spider-PM é parcialmente aderente aos resultados esperados DFP1, DFP2, DFP3 e DFP4. Do processo AMP, a Spider-PM é parcialmente aderente aos

resultados esperados AMP1 e AMP2. Com relação ao DFP1, DFP2, DFP4, AMP1 e AMP2, a Spider-PM é parcialmente aderente porque ela não mantém um registro com as modificações feitas no processo de software. A tarefa de manter esse registro deve ser atribuída a um sistema de controle de versões que deveria manter um registro de todas as modificações realizadas no arquivo que armazena a processo de software modelado. Com relação ao DFP2, a Spider-PM é parcialmente aderente devido a não realizar uma simulação dos processos modelados, deste modo, a Spider-PM não determina automaticamente o desempenho esperado dos processos de softwares. Apesar dessa relação com o MPS.BR, a Spider-PM é uma ferramenta para modelagem de processos de software de um modo geral, ou seja, tanto processos de software baseados no MPS.BR quanto não baseados podem ser modelados na Spider-PM. Apesar dessa aderência parcial, a utilização em conjunto da Spider-PM com uma ferramenta de controle de versões, por exemplo, é totalmente aderente aos resultados esperados DFP1, DFP2, DFP4, AMP1 e AMP2. Dessa forma, pesquisas estão sendo realizadas pelo Projeto Spider com o intuito de alcançar uma aderência total aos resultados esperados dos processos DFP e AMP. Assim, seja a partir da utilização em conjunto com outras ferramentas ou a partir do desenvolvimento de novas funcionalidades, futuramente será possível apoiar totalmente a implementação dos processos DFP e AMP do MPS.BR por meio de ferramentas livres.

11.2.3. Principais Funcionalidades

Devido ao fato de apresentar uma iteração muito semelhante à iteração dos modeladores UML, a Spider-PM não apresenta suas funcionalidades rigidamente definidas e particionadas em sua interface gráfica. Assim, muitas vezes, um mesmo componente da interface gráfica pode ser utilizado para dar acesso a diferentes funcionalidades dependendo do contexto no qual a Spider-PM se apresenta. Dessa forma, a descrição das funcionalidades

detalhadamente seria complexa, uma vez que seria necessária a descrição dos diferentes estados que podem ser assumidos pela Spider-PM. Além disso, a Spider-PM também apresenta uma grande quantidade de funcionalidades, dessa forma, essa seção descreve as principais funcionalidades da Spider-PM de um modo mais geral, destacando os pontos mais significativos envolvidos em cada uma dessas funcionalidades. A Figura 46 mostra a tela principal dessa ferramenta.

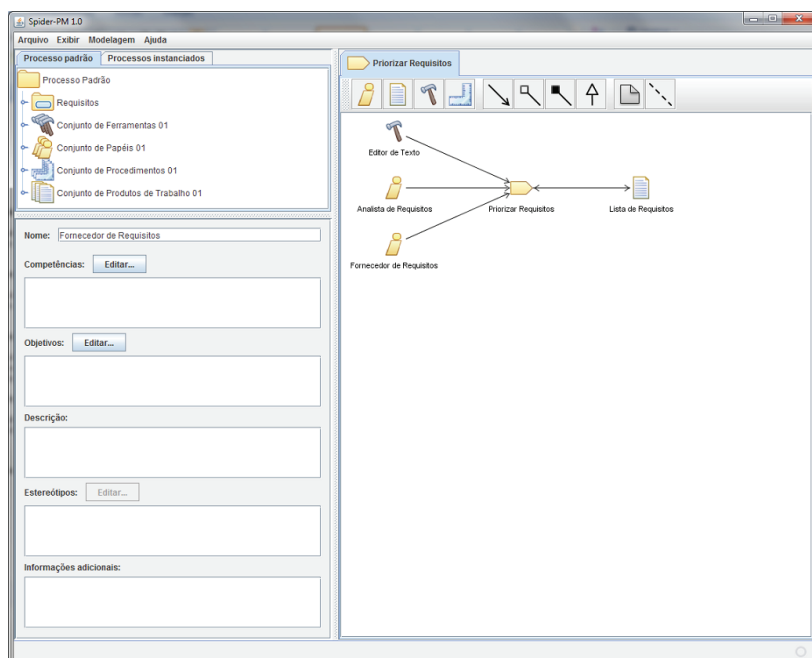


Figura 46. Tela principal da Spider-PM

Antes de descrever essas funcionalidades cabe uma breve explicação a respeito da estrutura geral da interface gráfica da ferramenta. A tela principal da Spider-PM está dividida em quatro áreas específicas. Uma dessas áreas apresenta a barra de menus; outra apresenta um diagrama que descreve um dos componentes da modelagem; em outra dessas áreas se encontram duas árvores que possibilitam a navegação entre os componentes da modelagem; e,

por fim, a área restante apresenta um conjunto de campos que permite visualizar e editar uma série de informações específicas sobre um dado elemento da modelagem.

A barra de menus permite acessar funcionalidades organizadas em quatro menus. O menu “Arquivo” reúne funcionalidades relacionadas com a manipulação do arquivo XML utilizado pela Spider-PM para salvar o processo de software modelado. O menu “Exibir” apresenta funcionalidades relacionadas com a exibição dos diagramas, como nível de zoom, por exemplo. O menu “Modelagem” apresenta funcionalidades relacionadas com a construção do modelo de processo de software. E o menu “Ajuda” apresenta funcionalidades que permitem o acesso à informações úteis para os usuários da ferramenta como a especificação da SPIDER_ML um guia de utilização da Spider-PM. A barra de menus está destacada na Figura 47.



Figura 47. Barra de menus da Spider-PM

A área da tela principal que contém as árvores de navegação serve para navegar entre os elementos da modelagem e para selecionar algum desses elementos para edição. Essa área está destacada na Figura 48. Existem duas estruturas em árvore nessa área da tela principal, mas apenas é possível visualizar uma dessas estruturas por vez. Para alternar entre uma árvore e outra, deve-se clicar nos guias rotulados com “Processo padrão” e “Processos instanciados”. O guia com rótulo “Processo padrão” permite o acesso à árvore para navegação entre os elementos do Processo Padrão. O guia com o rótulo “Processos instanciados” permite o acesso à árvore para navegação entre os elementos dos diversos Processos Instanciados a partir do Processo Padrão presente na modelagem.

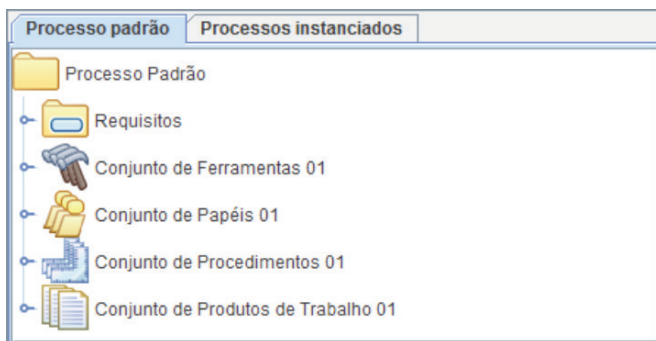


Figura 48. Árvores de navegação da Spider-PM

Alguns elementos podem apresentar diagramas que descrevem: ou sua estrutura interna por meio de um diagrama de pacotes; ou o fluxo de atividades e tarefas que o mesmo representa por meio de um diagrama de atividades; ou as associações desse elementos por meio de um diagrama de classes. A área da tela principal que mostra o diagrama que descreve um elemento selecionado e também possibilita a edição desse diagrama está destacada na Figura 49. Nessa figura, um diagrama de classes descreve uma Definição de Tarefa rotulada como “Priorizar Requisitos”.

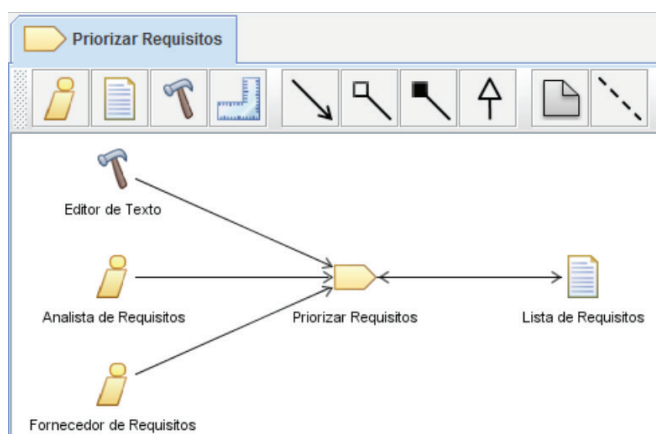


Figura 49. Área da tela principal para edição e visualizada dos diagramas da modelagem

Quando um elemento é selecionado num diagrama ou em uma das árvores de navegação, uma área da tela principal da ferramenta apresenta campos de acordo com o tipo do elemento selecionado. Esses campos podem ser utilizados para visualizar e editar detalhes específicos do elemento selecionado. Na Figura 50, essa área da tela principal está destacada mostrando os campos para detalhamento de uma Definição de Papel rotulada como “Fornecedor de Requisitos”.

The image shows a software interface for editing a 'Fornecedor de Requisitos' (Requirements Provider) element. The form is organized into several sections, each with a label and a corresponding input field or button:

- Nome:** A text input field containing the value 'Fornecedor de Requisitos'.
- Competências:** A label followed by an 'Editar...' button and a large empty text area for editing competencies.
- Objetivos:** A label followed by an 'Editar...' button and a large empty text area for editing objectives.
- Descrição:** A label followed by a large empty text area for editing the description.
- Estereótipos:** A label followed by an 'Editar...' button and a large empty text area for editing stereotypes.
- Informações adicionais:** A label followed by a large empty text area for additional information.

Figura 50. Área da tela principal para detalhamento dos elementos da modelagem

11.2.4. Modelagem do Processo Padrão

A definição do Processo Padrão ocorre por meio da elaboração de um diagrama de pacotes que apresenta as dependências entre as Disciplinas que o mesmo apresenta. Cada Disciplina, por sua vez, deve ser descrita por meio de um diagrama de pacotes que mostra as dependências entre as Definições de Tarefa contidas nessa Disciplina. As Definições de Tarefa, por sua vez, devem ser descritas por meio de um diagrama de classes onde são expostas as relações dessa Definição de Tarefa com os demais elementos que fazem parte do Processo Padrão, esses elementos são: Definição de Papel, Definição de Ferramenta, Definição de Produto de Trabalho e Procedimento.

Como pôde ser observado na Figura 46, as Definições de Papéis, Ferramentas, Produtos de Trabalho e os Procedimentos não são agrupados dentro das Disciplinas. Esses elementos são organizados por tipo separadamente, uma vez que eles não estão diretamente relacionados a uma disciplina do Processo Padrão, por exemplo, uma Definição de Ferramenta ou um Procedimento podem ser associados com Definições de Tarefas de diferentes Disciplinas.

Como pôde ser observado na Figura 46 e na Figura 49, acima do diagrama que descreve o elemento atualmente selecionado numa das barras de navegação existe uma barra de ferramentas. Para criar novos elementos deve-se clicar no botão com o ícone correspondente ao elemento que se deseja criar e, em seguida, deve ser clicado no diagrama o ponto onde esse elemento deve ser posicionado. Ao criar um novo elemento o mesmo será imediatamente adicionado numa das árvores navegação.

Durante a modelagem do Processo Padrão, para adicionar em um diagrama um elemento já criado, deve-se localizá-lo na árvore de navegação e arrastá-lo para o ponto do diagrama onde esse elemento deve ser posicionado.

Para todos os elementos da modelagem é possível o detalhamento desse elemento por meio de campos que são apresentados de acordo com o tipo do elemento. Caso esses campos não sejam suficientes para organizar todas as

informações que se deseja associar a um dado elemento, é possível a utilização de notas.

11.2.5. Modelagem dos Processos Instanciados

A partir de um Processo Padrão deve-se definir um processo que esteja de acordo com as características de um projeto específico, denominado Processo Instanciado. Apesar de esse ser o fluxo convencional para modelagem do processo de software e de acordo com o modelo para definição de processos utilizado por esse trabalho, a modelagem de um Processo Instanciado pode ser realizada sem a necessidade de um Processo Padrão estar modelado.

A Figura 51 mostra uma parte da tela da Spider-PM onde se encontra um modelo de Processo Instanciado não finalizado. Na Figura 51 é possível visualizar a estrutura interna de um Processo Instanciado.

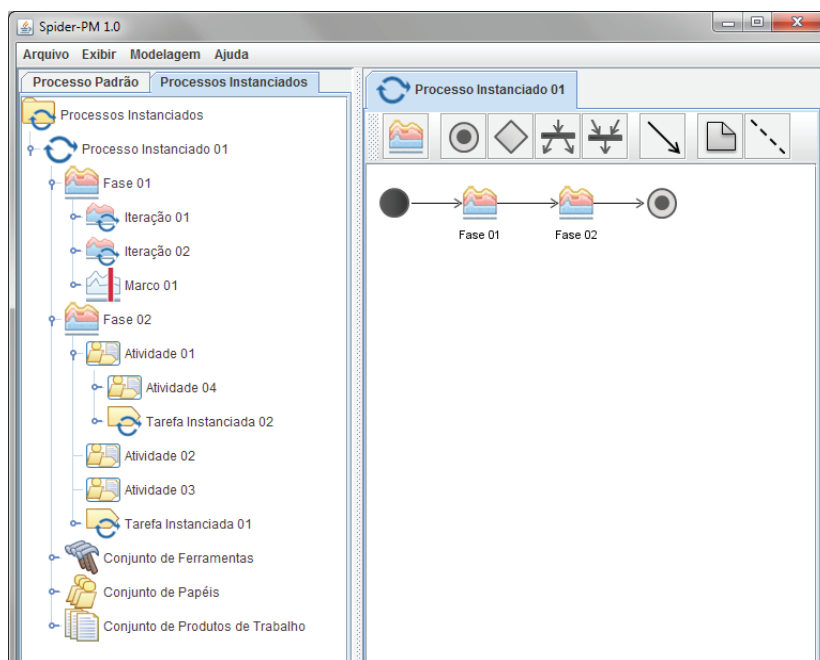










Figura 51. Modelagem de um Processo Instanciado

A modelagem de um Processo Instanciado segue da mesma maneira como descrito para o Processo Padrão. Por outro lado, é possível reutilizar as Definições de Tarefa, de Papéis, de Ferramentas, arrastando esses componentes do processo padrão para os diagramas utilizados na modelagem de um Processo Instanciado. Ao arrastar um desses elementos o mesmo é convertido para um elemento correspondente no contexto do Processo Instanciado. A Tabela 16 mostra a correspondência entre esses elementos. Ao arrastar um componente do Processo Padrão para um Processo Instanciado, todos os demais componentes com os quais o componente arrastado se relaciona também são convertidos e adicionados ao Processo Instanciado.

Tabela 16. Correspondência entre os elementos do Processo Padrão e do Processo Instanciado

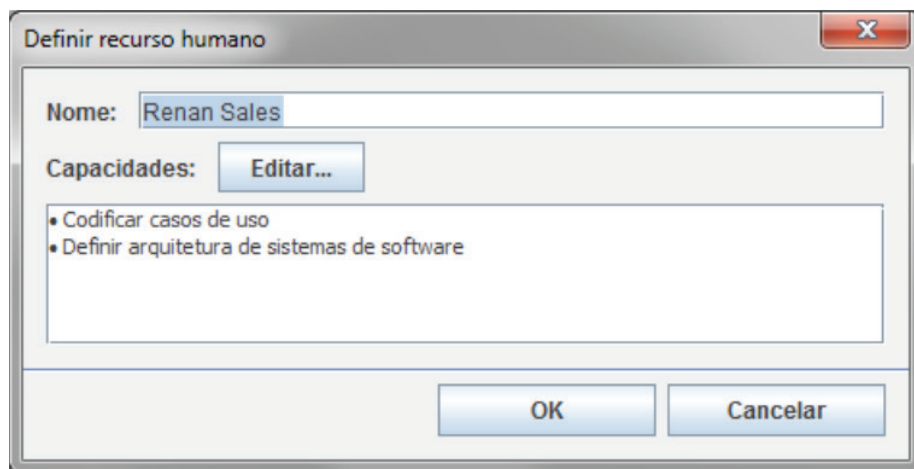
Tipo do Elemento do Processo Padrão	Tipo Correspondente do Elemento no Processo Instanciado
	
	
	
	

A diferença básica entre os elementos está relacionada com a natureza das informações que cada um representa, por exemplo, um Papel Instanciado pode estar associado com um recurso humano do projeto, um Produto de Trabalho Instanciado pode estar associado a um arquivo específico, etc. Assim, os elementos do Processo Padrão apresentam informações mais genéricas, enquanto os elementos de um Processo Instanciado apresentam informações que estão diretamente relacionadas com o processo que será de fato executado.

Além de fazer a reutilização dos elementos utilizados na modelagem do Processo Padrão, também é possível arrastar as Fases, Iterações, Marcos, Atividades e demais elementos de um Processo Instanciado para outro. Essa funcionalidade visa facilitar a definição de Processos Instanciados que são semelhantes. Assim, todas as características compartilhadas entre dois ou mais Processos Instanciados podem ser definidas apenas uma vez em um Processo Instanciado e reutilizadas nos demais Processos Instanciados onde elas devem estar presentes.

11.2.6. Controle de Recursos Humanos

É possível associar recursos humanos aos Papéis Instanciados dos Processos Instanciados. Para que essa associação seja possível deve-se cadastrar esse recurso humano para que posteriormente o mesmo possa ser associado. A Spider-PM considera que um recurso humano apresenta uma lista de capacidades. Assim, como pode ser observado na Figura 52, a definição de um recurso humano apenas requer que sejam fornecidos o nome desse recurso humano e uma lista contendo as capacidades do mesmo.



Definir recurso humano

Nome: Renan Sales

Capacidades: Editar...

- Codificar casos de uso
- Definir arquitetura de sistemas de software

OK Cancelar

Figura 52. Definição de um recurso humano

Esse controle de recursos humanos apenas representa uma facilidade adicional provida pela Spider-PM, a utilização da mesma é opcional, pois os modelos de processos podem ser definidos sem que nenhum Papel Instanciado dos Processos Instanciados estejam relacionados com algum recurso humano. Por outro lado, a associação com recursos humanos visa uma melhor descrição do Processo Instanciado.

11.2.7. Controle de Estereótipos

A Spider-PM permite a utilização de estereótipos UML para a extensão da semântica dos elementos e relacionamentos disponibilizados por ela disponibilizados. Para que um estereótipo seja utilizado na modelagem, o mesmo pode ser criado por meio da tela ilustrada na Figura 53.

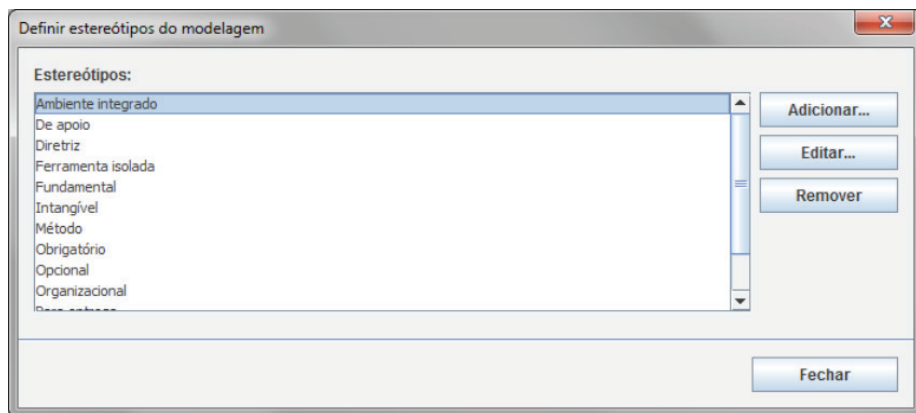


Figura 53. Definição de estereótipos

11.2.8. Controle de Grupos de Estereótipos

A Spider-PM também apresenta uma funcionalidade que permite a definição de regras para utilização dos estereótipos. Basicamente, pode-se definir

um grupo de estereótipos e associá-lo a um determinado tipo de elemento da modelagem. Quando um desses estereótipos for atribuído a um elemento do tipo determinado, os demais estereótipos dessa lista não mais poderão ser atribuídos a esse componente. Por exemplo, um grupo com os estereótipos “fundamental”, “de apoio” e “organizacional” pode ser definido e associado às Definições de Tarefas. Nesse caso, uma Definição de Tarefa do Processo Padrão poderá ser classificada de acordo com os estereótipos dessa lista e não poderá receber dois ou mais desses estereótipos concomitantemente, ou seja, uma Definição de Tarefa não poderá ser “fundamental” e “de apoio” ao mesmo tempo, por exemplo. Além de verificar as regras e restrições da linguagem SPIDER_ML, a Spider-PM também verifica se os elementos respeitam as definições de estereótipos criadas pelos usuários. Na Figura 54 é mostrada a tela onde são definidos esses grupos de estereótipos.

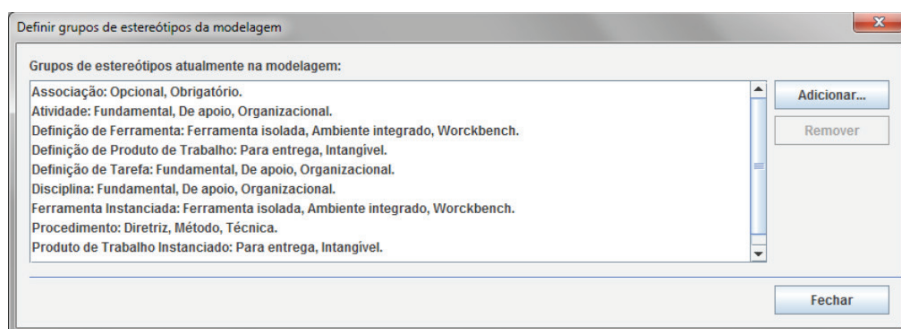


Figura 54. Definição de grupos de estereótipos



Capítulo 12

Implementação do Processo de Avaliação e Melhoria do Processo Organizacional

em [Mezzomo e Oliveira, 2011a], [Mezzomo e Oliveira, 2011b]

Leonardo Possamai Mezzomo
Sandro Ronaldo Bezerra Oliveira

Para se desenvolver um software atualmente, é exigido muito mais do que a ideia de criar uma ferramenta ou utilizar determinada linguagem de programação, é necessário que haja também um esforço coletivo, complexo e criativo, pois a qualidade de um produto de software depende fortemente das pessoas, da organização e dos procedimentos usados para criá-lo e disponibilizá-lo [Fuggetta, 2000].

A qualidade não é apenas um diferencial de mercado para a empresa conseguir vender e lucrar mais, mas sim um pré-requisito que a empresa deve conquistar para conseguir colocar seu produto no mercado global. No contexto de desenvolvimento de software, qualidade pode ser entendida como um conjunto de características a serem satisfeitas, de modo que o produto de software atenda às necessidades de seus usuários [Moraes, 2010]. A busca pela qualidade de seus produtos tem forçado as empresas desenvolvedoras de software a buscarem nos processos de software a solução que guie o que, como e quando fazer algo. Acredita-se que a chave para a qualidade do software depende da qualidade do processo usado para construí-lo [Oliveira, 2007].

De acordo com Humphrey (1989), um processo de software é o conjunto de tarefas de engenharia de software necessárias para transformar os requisitos dos usuários em software. Portanto, para se definir um processo, devem-se considerar algumas informações: atividades a serem realizadas; recursos utilizados; artefatos consumidos e gerados; procedimentos adotados; paradigma e tecnologia adotados; e o modelo de ciclo de vida utilizado [Falbo *apud* Pfleeger, 1998]. Dessa forma, um processo de software pode ser considerado como um conjunto de etapas ou atividades que devem ser seguidas por uma equipe técnica com a finalidade de desenvolver um produto ou bem de serviço.

À medida que implementam seus processos, as organizações também têm se preocupado em avaliá-lo em melhorá-lo. Um processo de software não pode ficar estagnado no tempo, refletindo pontos críticos que levam a problemas indesejados no desenvolvimento de um produto. Eles precisam constantemente passar por refinamentos e modificações com o intuito de aumentar sua habilidade e agilidade no decorrer de projetos. Portanto, eles necessitam ser continuamente melhorados [Fuggetta, 2000].

O processo de Avaliação e Melhoria do Processo Organizacional tem o objetivo de determinar quanto os processos da organização contribuem para alcançar os objetivos de negócio e para apoiar, realizar e implantar melhorias contínuas nos processos [SOFTEX, 2012a]. Apesar de ser um processo independente, é imprescindível sua implementação caso exista o processo de definição implantado.

Por mais que se discuta muito sobre avaliações e melhorias contínuas nos processos, os engenheiros de software e as empresas ainda possuem dificuldades em definir seus processos, basicamente porque não existe um processo de software possível de ser genericamente aplicado, pelas diversidades das organizações [Koscianski, 2007].

Entretanto, o processo pode e deve ser melhorado a cada novo projeto, para ser refinado continuamente e para aumentar sua capacidade de lidar com os requisitos e expectativas do mercado e da organização, e aumentando a eficiência e produtividade da organização como um todo. Estas observações

motivaram uma variedade de trabalhos dedicados à criação de modelos da qualidade e métodos para melhoria de processos de software [Kruchten, 2003; Fuggetta, 2000].

Para conseguir tais melhoria nos processos é importante avaliá-los. A avaliação de um processo aborda diversos aspectos, que podem ser desde seus ativos (tarefas, atividades, procedimentos, ferramentas, etc.) até seus produtos resultantes de projetos. Ela é essencial para prover apoio à elaboração de estratégias de melhoria [Fuggetta, 2000].

Nos dias atuais, o processo de Avaliação e Melhoria do Processo Organizacional é tratado pelos modelos de qualidade. Esses modelos, guias e normas visam orientar as organizações com as boas práticas sobre aplicação e implantação desses processos. Dentre os principais, pode-se citar o modelo CMMI, o programa MPS.BR e a norma ISO/IEC 12207.

Dessa forma, este trabalho consiste em uma abordagem sobre os processo de Avaliação e Melhoria do Processo Organizacional que esteja alinhada aos modelos de qualidade do MPS.BR, CMMI e ISO/IEC 12207. Essa abordagem terá como base a especificação e implementação de um ferramental composto por: framework de processo; e uma ferramenta livre de apoio sistêmico às atividades definidas no *framework*. Assim, com este ferramental, espera-se simplificar a implementação do processo de avaliação e melhoria em organizações que buscam o aprimoramento, padronização e institucionalização de seus processos de desenvolvimento de software, com ênfase na aplicação dos ativos organizacionais em projetos de software.

12.1. Framework de Avaliação e Melhoria de Processo de Software

O *framework* de processo tem o intuito de tornar a organização interessada na qualidade de seus processos de avaliação e melhoria, aderente não só ao modelo MR-MPS-SW mas também a outro modelo e norma.

Todas as atividades constantes no *framework* possuem correlação com um ou mais itens constantes nos modelos e normas tratadas neste trabalho. Contudo, nem todos os itens foram transferidos para o *framework*, casos esses que foram justificados quanto à relevância para o framework. Isso não quer dizer que a atividade é irrelevante, mas sim que não tem influência significativa no contexto do *framework*, já que se trata de atividade não técnica relativa a ações administrativas/burocráticas.

Para se elaborar o *framework* foram utilizados três fluxos, sendo um deles denominado de macro fluxo contendo macro atividades e dois possuindo atividades específicas. As macro atividades foram chamadas de Avaliar Processo e Melhorar Processo, possuindo, cada uma, um conjunto de outras atividades que formam os fluxos específicos.

O *framework* foi modelado usando as notações do BPMN – *Business Process Modeling Notation* [OMG, 2011] com o auxílio do software livre *BizAgi Process Modeler* 1.5.1. O BPMN é uma notação gráfica para modelagem de processos que possui diferentes símbolos para identificar cada um dos elementos de um processo.

A Figura 55 apresenta um fluxo geral entre as macro atividades. Neste fluxo considera-se a necessidade de se avaliar ou melhorar um processo. As estruturas de decisões auxiliam o entendimento sobre os possíveis passos que podem ser realizados. A primeira estrutura de decisão tende a identificar a existência de um processo, possibilitando que seja posteriormente avaliado ou melhorado pelas atividades específicas. Imagina-se que já exista um processo, definido anteriormente, e ele sirva de entrada para a macro atividade “Avaliar Processo”. Existe, ainda, outra decisão que possibilita que um processo possa ser melhorado, caso seja identificadas oportunidades de melhoria ao final da avaliação. Em resumo, se parte do pensamento de que um processo possa ser avaliado e melhorado de maneira genérica.

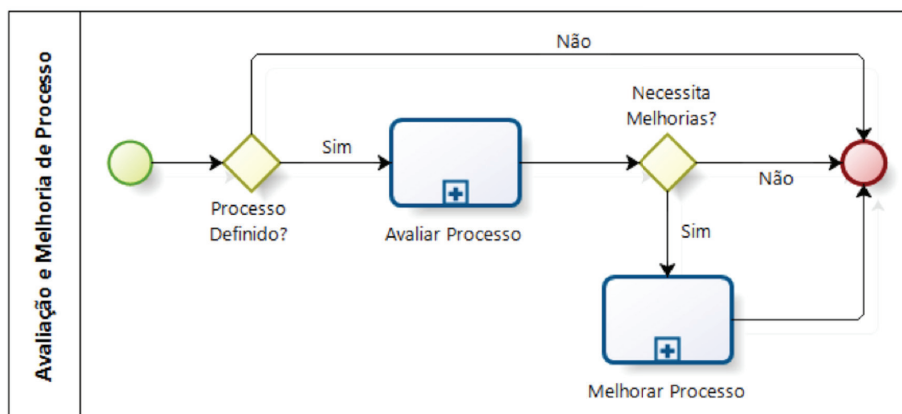


Figura 55. Fluxo Geral com as Macro Atividades do *Framework*

Inseridos nas macro atividades existem as atividades que compõem os fluxos direcionados ao propósito de avaliação ou melhoria do processo. A seguir, são listadas algumas dessas atividades:

- **Avaliar Processo:** atividades relacionadas à avaliação por critérios objetivos ou por métricas;
- **Melhorar Processo:** atividades relacionadas à aplicação de melhoria a partir do uso do modelo IDEAL - *Initiating Diagnosing Establishing Acting Learning*.

Os fluxos dessas atividades seguem linhas de raciocínios sequenciais ocorrendo tomadas de decisão quando necessário. A primeira atividade existente no fluxo geral é a Avaliar Processo. Ao todo, este fluxo específico possui 14 (quatorze) atividades executadas por apenas um papel, o de Gerente de Processo.

O *framework* de avaliação inicia com uma atividade que possibilita que informações de uso do processo possam ser consultadas antes de utilizar de fato o processo. Em seguida, as necessidades e os objetivos da avaliação são definidos e é realizada a seleção do tipo de avaliação, que poderá ser através

de métricas (metricamente – usando a ferramenta Spider-MsControl, disponível em www.spider.ufpa.br) ou questões objetivas (objetivamente – usando a ferramenta Spider-CL, também disponível em www.spider.ufpa.br). As atividades contidas em avaliações utilizando métricas seguem um fluxo simples de definição de objetivos e medidas, e posteriormente a coleta e análise das medidas. O outro tipo de avaliação possui um conjunto de atividades que envolvem a definição de questões e aplicação de questionário, para, posteriormente, ser realizada a coleta e análise de dados. Entretanto, os dois tipos de avaliação geram ao final um relatório que serve de base para a melhoria do processo. A Figura 56 ilustra as atividades de avaliação de processo do *framework*.

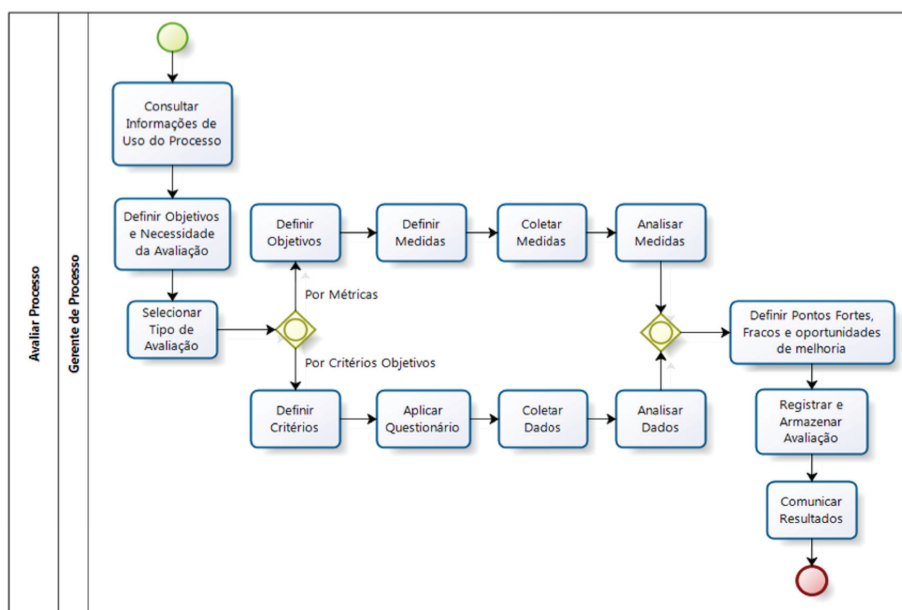


Figura 56. Atividades de Avaliação de Processo do *Framework*

O relatório produzido pelas atividades de avaliação possui grande importância, pois é através dele que a atividade Definir Pontos Fortes, Fracos e Oportunidades de Melhoria provê apoio à elaboração do que deve ser melhorado no processo. A partir deste insumo, esta atividade também gera

um relatório contendo todos os pontos fracos e oportunidades de melhoria que poderão ser alcançados no processo avaliado. Em seguida, o resultado da avaliação é registrado e armazenado através da atividade Registrar e Armazenar Avaliação e comunicado a todos interessados pela atividade Comunicar Resultados.

Além de contar com atividades de avaliação, o *framework* também possui atividades relacionadas com a melhoria de processos. O relatório de melhorias gerado na avaliação é considerado fundamental para as atividades de melhorias. Neste contexto, o *framework* é composto de 17 (dezesete) atividades baseadas em um modelo de melhoria chamado IDEAL. O modelo IDEAL é um padrão de aperfeiçoamento organizacional que serve como um guia para iniciar, planejar e implementar ações de melhoria, além de fornecer uma abordagem usável e fácil de entender para o aprimoramento contínuo por mostrar os passos necessários para se estabelecer um programa de melhoria de processos bem sucedido [Mcfeeley, 1996].

A palavra IDEAL é um acrônimo em inglês para Iniciar (*Initiating*), Diagnosticar (*Diagnosing*), Estabelecer (*Establishing*), Agir (*Acting*) e Aprender (*Learning*), sendo que cada palavra representa uma fase do modelo. Seguir as fases, atividades e princípios do modelo IDEAL têm-se provado benéfico em muitos esforços de melhorias. Este modelo surgiu da necessidade de implementação de um guia específico que adotasse novas ferramentas de engenharia de software, processos e métodos.

Muitos esforços de melhoria, incluindo melhoria do processo de software, contínuo gerenciamento de riscos, ou a introdução de um novo ambiente de desenvolvimento, são tão complexos, e seus efeitos tão de longo alcance, que eles requerem uma abordagem especializada e sistemática para gerenciar o ciclo de vida de adoção da tecnologia. Assim, o *Software Engineering Institute* (SEI) desenvolveu e refinou o Modelo IDEAL para ajudar a satisfazer essa necessidade. Optou-se por utilizar este modelo, pois ele disponibiliza um guia disciplinado de engenharia para a melhoria contínua, foca no gerenciamento

do programa de melhoria e estabelece a base para uma estratégia de melhoria em longo prazo.

Foi assumida para esta parte do *framework* a presença dos papéis de Gerente e Projetista do Processo, sendo o primeiro mais relacionado aos gestores da organização, tendo a responsabilidade de discutir com a alta gestão o que deve ser melhorado e viabilizar meios para que a melhoria ocorra, enquanto o segundo executa as atividades de melhoria propriamente ditas. A Figura 57 a seguir apresenta atividades de melhoria de processo contidas no *framework*, bem como o alinhamento das fases do modelo IDEAL.

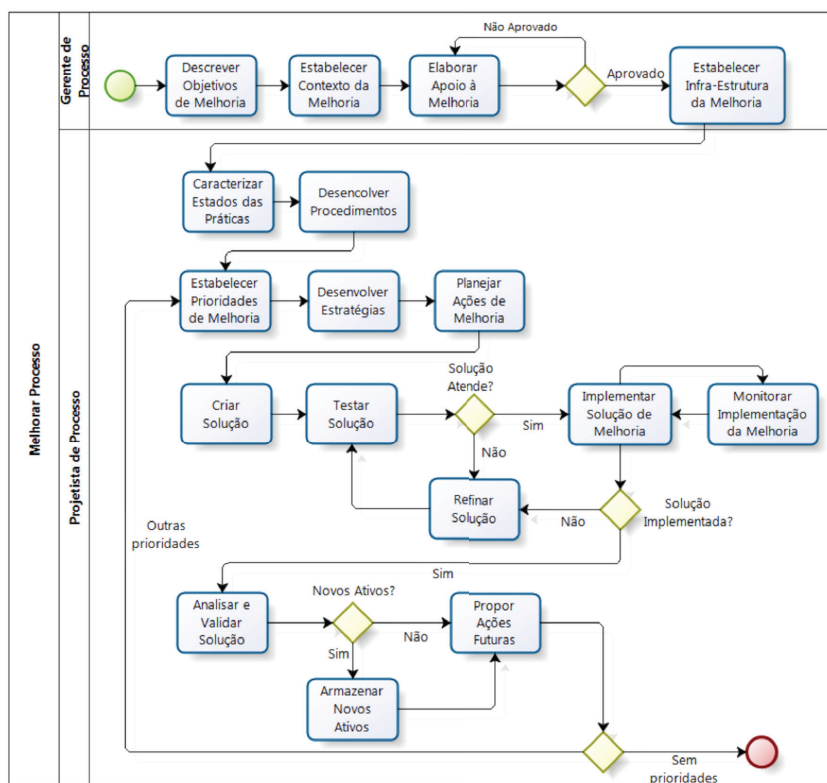


Figura 57. Atividades de Melhoria de Processo do *Framework*

As atividades iniciais executadas pelo Gerente do Processo dizem respeito à fase de Iniciação do modelo IDEAL. A primeira delas Cadastrar

Objetivos da Melhoria trata de reconhecer as razões de estímulo para realizar uma mudança na organização, que, neste caso, seriam as melhorias advindas de uma avaliação. Após a definição dos objetivos, a próxima atividade visa Estabelecer o Contexto da Melhoria, que procura abordar os esforços, benefícios e vantagens que uma melhoria necessita e pode proporcionar. Com os objetivos e o contexto bem estabelecidos, a atividade Elaborar Apoio a Melhoria possui a finalidade de estabelecer o patrocínio para as melhorias, que leva em conta a quantidade de melhorias, esforço, valor financeiro e de estrutura, entre outros. Este apoio pode ser aprovado ou reprovado, podendo ser refinado quando não aprovado.

Uma vez que o patrocinador esteja comprometido com a melhoria, a atividade Estabelecer Infraestrutura de Melhoria leva em consideração o tamanho do esforço e a complexidade das melhorias. No caso de aprimoramento de processos, por exemplo, pode-se exigir de uma organização a alocação de mais um membro da equipe para realizar determinada melhoria. Além disso, esta atividade também trata de infraestrutura física para o tratamento das melhorias.

A segunda fase do modelo IDEAL, chamada Diagnóstico, engloba duas atividades: Caracterizar Estados das Práticas e Desenvolvimento de Procedimentos. A primeira tem a finalidade de especificar o estado atual das práticas antes de uma melhoria e qual o estado desejado após a conclusão da execução da melhoria. A segunda visa estabelecer os procedimentos e recomendações necessários para se alcançar o estado almejado pela melhoria.

O propósito da terceira fase do modelo IDEAL, denominada Estabelecimento, é de desenvolver um planejamento de trabalho, sendo ela composta por três atividades. A primeira delas Estabelecer Prioridades define, por exemplo, as prioridades para o esforço da melhoria de um processo organizacional. Neste caso, devem-se levar em consideração alguns fatores, tais como: a dependência entre as atividades recomendadas, fatores externos, recursos limitados, entre outros.

Após a caracterização do estado das práticas e da definição das prioridades, há um acúmulo de informações sobre o entendimento do escopo das melhorias, forçando a existência da atividade Desenvolver Estratégia, que tem como foco criar uma abordagem para realizar os trabalhos e administrar a disponibilidade de recursos. Com a abordagem bem definida, um planejamento detalhado pode ser desenvolvido. Através da atividade Planejar Ações, é possível planejar agenda, tarefas, pontos de decisão, recursos, responsabilidades, métricas, mecanismos que controlem riscos e estratégias.

Ação é o nome da quarta fase do modelo IDEAL e tem por objetivo implementar soluções para atender as atividades anteriores. De acordo com o *framework*, existem 5 (cinco) atividades nesta fase. A atividade Criar Solução prende-se a elementos chaves que podem contribuir como uma solução de melhoria, que podem ser: ferramentas, processo, conhecimento, habilidades, entre outros. Uma solução quando criada deve ser testada para que possa ser validada ou não, sendo que, em casos de reprovação, a solução pode ainda ser refinada pela atividade Refinar Solução, onde elementos podem ser retirados ou acrescidos em seu escopo. Quando uma solução é aceita, ela passa para outro estágio ou atividade, denominada no *framework* como Implementar Solução, que trata do tipo de abordagem que deve ser definido para que a melhoria seja implementada. Esta implementação da melhoria pode e deve ser monitorada, através da atividade Monitorar Implementação da Melhoria para que haja um controle maior do que está sendo realizado.

Por fim, a quinta e última fase do modelo IDEAL, chamada Aprendizado, possui atividades responsáveis por coletar as experiências alcançadas com a melhoria, verificando também se os objetivos inicialmente elaborados foram contemplados. Com isso, ações futuras podem ser propostas, a fim de que em outros procedimentos de melhoria ganhe-se maturidade e eficiência para o aprimoramento dos processos. Vale ressaltar, ainda, que caso a melhoria identifique que novos ativos devam fazer parte do processo, eles precisam ser documentados e armazenados no repositório da organização.

Assim, o módulo de avaliação de processos da Spider-PM e a ferramenta Spider-PI abordados neste trabalho, visam sistematizar as atividades descritas neste *framework*.

12.2. O Módulo de Avaliação da Ferramenta Spider-PM e a Ferramenta Spider-PI

O projeto SPIDER tem, como um dos focos principais, apresentar um levantamento das ferramentas de software livre com características adequadas para possibilitar a criação de produtos de trabalhos (artefatos que evidenciam a implementação do programa da qualidade organizacional) derivados dos resultados esperados/práticas específicas descritos nos objetivos das áreas de processo do modelo MR-MPS-SW e CMMI-DEV.

Em decorrência deste levantamento, pretende-se internamente no projeto especificar e desenvolver um SUITE de ferramentas a fim de propiciar um uso mais integrado das suas funções/operações disponíveis, de modo a apoiar a implantação dos processos/áreas de processo do modelo MR-MPS-SW e CMMI-DEV, obedecendo o fluxo de dependência proposto por este modelo de qualidade de processo.

O objetivo deste projeto então visa apresentar alternativas viáveis com relação a ferramentas de software para auxiliar a implementação dos modelos de qualidade nas organizações, sem a necessidade de aquisição de softwares proprietários e com a possibilidade da ferramenta ser customizada para atender as especificidades da organização, diminuindo os custos e o tempo ao longo da implementação deste programa de maturidade.

Dentre as ferramentas integradas na SUITE do projeto, atualmente, podemos citar as principais:

- Spider-PM, ferramenta de definição e modelagem de processos de software;

- Spider-REUSE, ferramenta de apoio ao processo de Gerência de Reutilização de Ativos de Software;
- Spider-ACQ, ferramenta de apoio o processo de Aquisição de Produtos e Serviços;
- Spider-QA, ferramenta de apoio ao processo de Garantia da Qualidade;
- Spider-MsControl, ferramenta de apoio ao processo de Medição;
- Spider-CL, ferramenta de elaboração de questionário.

Neste contexto, assim, possuímos a Spider-PM sendo uma ferramenta de licença GPL, voltada especificamente para apoiar a modelagem e a definição de processos, usando o padrão SPEM. Com a implementação do módulo, a ferramenta Spider-PM também passou a apoiar a avaliação de processos de software. Já a Spider-PI é uma ferramenta que também possui licença GPL e apoia o planejamento e a execução de melhorias em processos de software.

O módulo de avaliação e a ferramenta de melhoria podem ser utilizados em diferentes organizações, independentemente do tamanho da mesma, e se encontram disponíveis em http://www.spider.ufpa.br/projetos/spider_amp/spiderAMP.rar.

Haja vista o papel fundamental da avaliação de processos de software, pode-se destacar que a utilização de ferramental para apoiar o processo de avaliação é de grande importância, como ratificado por Hunter, Robinson e Woodman (1997):

Ferramentas de [...] avaliação de software são, portanto, de valor tanto para os produtores de software que desejam melhorar seus processos em comparação com seus concorrentes, como para compradores de software que desejam avaliar os processos de potenciais contratantes [...].

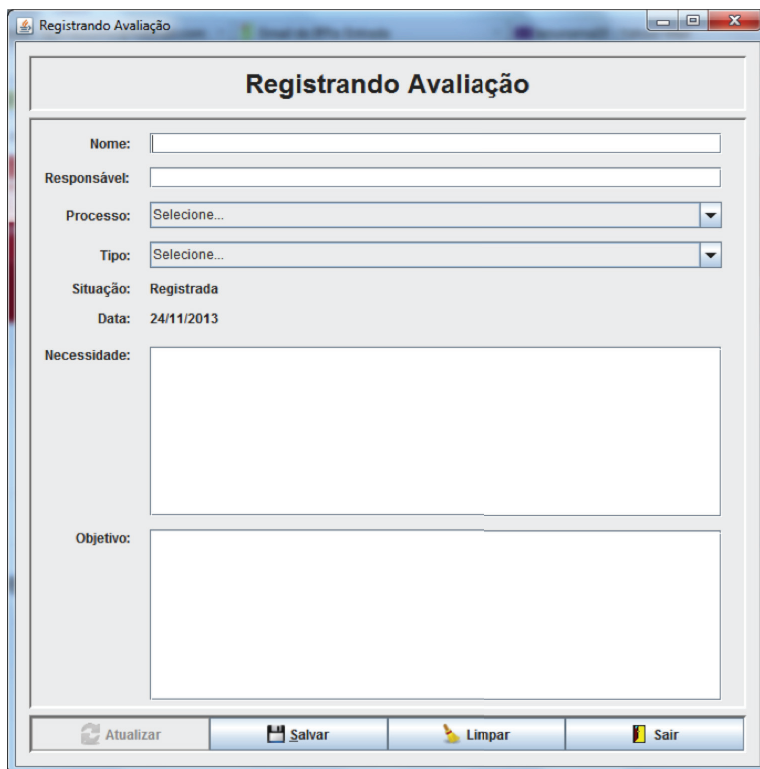
Atualmente, a maioria das avaliações de processos de software é realizada apenas registrando-se as informações em documentos de texto e em planilhas eletrônicas. Isso torna o trabalho bem manual, com mais custo de tempo e com maior dificuldade para manipular as informações da avaliação [Neiva e Oliveira, 2010].

Dentro deste contexto, o módulo e a ferramenta visam auxiliar as organizações interessadas na avaliação e melhoria de processos, usando os modelos de referência MR-MPS-SW, CMMI-DEV e a norma ISO/IEC 12207, provendo um acompanhamento sistematizado de atividades do processo de avaliação e melhoria. Apesar do apoio sistêmico, é imprescindível enfatizar que o módulo e a ferramenta não tendem a eliminar o trabalho intelectual do avaliador, necessário em uma avaliação, por conta da subjetividade existente, mas sim dar suporte à análise e ao registro dos resultados obtidos neste processo.

12.2.1. O Módulo de Avaliação

Para utilizar o módulo de avaliação é necessário que a ferramenta Spider-PM seja acessada, não sendo necessário que o usuário faça autenticação por meio de login e senha. Dentro da ferramenta, o usuário poderá acessar as funções do módulo através do menu Processo e em seguida a opção Avaliação.

A primeira e uma das principais funcionalidades do módulo é a que registra a avaliação. Ela é responsável por possibilitar cadastro de uma nova avaliação, possibilitando ao usuário que ele crie uma avaliação preenchendo alguns campos como: nome, responsável, seleção do processo a ser avaliado, tipo de avaliação (por critérios objetivos ou por métricas), descrição da necessidade e do objetivo. A Figura 58 mostra como uma avaliação é registrada/cadastrada.



Registrando Avaliação

Nome:

Responsável:

Processo:

Tipo:

Situação: Registrada

Data: 24/11/2013

Necessidade:

Objetivo:

Atualizar Salvar Limpar Sair

Figura 58. Tela de Registro de Avaliação

A funcionalidade de consulta à avaliação apresenta aos usuários todas as avaliações de processos que foram cadastradas. Há a possibilidade de se listar todas as avaliações registradas ou pode-se realizar consultas considerando três filtros:

- Por Tipo: lista as avaliações de acordo com o tipo de avaliação registrados no momento do registro;
- Por Situação: lista as avaliações de acordo com sua situação momentânea, podendo ser: a) Registrada: quando a avaliação foi apenas cadastrada; b) Inicializada: quando a avaliação já estiver sido iniciada; c) Finalizada: quando os relatórios da avaliação já tiverem sido gerados e carregados no módulo;

- Por Processo: lista as avaliações de acordo com o processo selecionado no momento do registro.

Independentemente do tipo de consulta selecionada, os itens apresentados são os seguintes: nome, responsável, processo, tipo, situação e data. Após a busca na base de dados, é possível selecionar a avaliação e alterá-la ou mesmo excluí-la. A Figura 59 apresenta como é realizada a consulta a uma avaliação.

A interface 'Consultando Avaliação' possui um título centralizado. Abaixo dele, há uma seção 'Consulta por:' com três opções de radio button: 'Tipo', 'Situação' e 'Processo'. Cada opção é seguida por um campo de texto com o rótulo 'Selecione o Tipo:', 'Selecione a Situação:' ou 'Selecione o Processo:', respectivamente, e um botão 'Listar' com um ícone de lupa. Abaixo disso, há uma seção 'Resultados:' contendo uma tabela com seis colunas: 'Nome', 'Responsável', 'Processo', 'Tipo', 'Situação' e 'Data'. A tabela está vazia. Na base da interface, há uma barra com quatro botões: 'Alterar' (com ícone de ferramenta), 'Listar Todas' (com ícone de documento), 'Excluir' (com ícone de lixeira) e 'Sair' (com ícone de porta).

Figura 59. Tela de Consulta de Avaliação

Após a avaliação ser registrada, é necessário que a mesma seja inicializada pelo módulo. A tela desta funcionalidade apresenta um componente que lista todas as avaliações com a situação “Registrada”, que automaticamente podem ser iniciadas. Selecionando uma avaliação e clicando-se no botão “Iniciar”, será carregada a outra ferramenta para realizar a avaliação, dependendo do tipo selecionado no momento do registro da mesma, por exemplo,

se for escolhido o tipo “Por critérios objetivos”, a ferramenta Spider-CL será aberta; por outro lado, se o tipo escolhido for “Por métricas”, a ferramenta Spider-MsControl será iniciada. O módulo possibilita ainda que os dois tipos de avaliação sejam selecionados conjuntamente, podendo ser iniciadas as duas ferramentas ao mesmo tempo. A Figura 60 mostra como a avaliação pode ser iniciada.

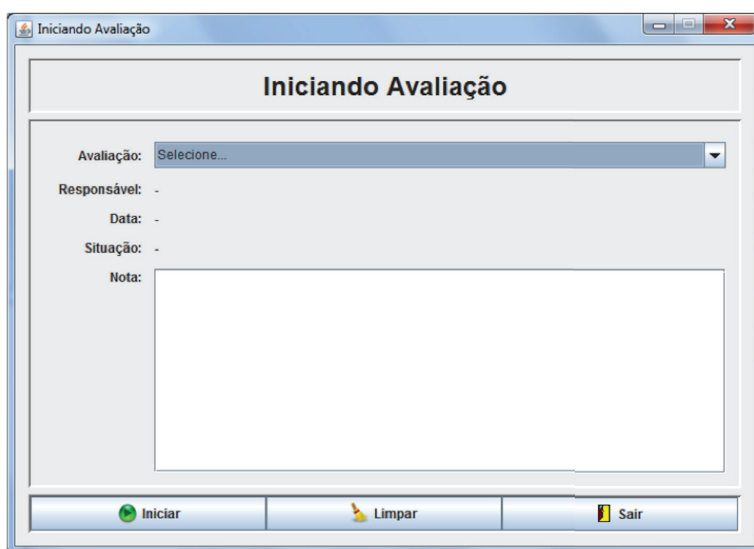


Figura 60. Tela de Início da Avaliação

A opção de armazenar a avaliação possibilita que os relatórios produzidos pela avaliação sejam armazenados no módulo. A interface gráfica desta funcionalidade, como pode ser visto na Figura 61, apresenta um componente que lista todas as avaliações com a situação “Iniciada”, que caso já tenham sido realizadas, podem ter seu(s) relatório(s) armazenado(s). Dependendo da avaliação selecionada, os demais campos são habilitados. Os relatórios devem ser carregados do computador para o módulo e posteriormente salvos no repositório previamente configurado. Portanto, esta interface gráfica possibilita que os relatórios de uma avaliação sejam armazenados em um repositório

para que fique à disposição de outros interessados. Após o armazenamento do relatório a situação da avaliação passa a ser “finalizada”.

A interface 'Armazenando Avaliação' apresenta os seguintes campos e botões:

- Avaliação:** Menu suspenso com o texto 'Selecione...'.
- Responsável:** Campo de texto com o caractere '-'.
- Data:** Campo de texto com o caractere '-'.
- Situação:** Campo de texto com o caractere '-'.
- Carregando Relatório(s):**
 - 1) Relatório de Avaliação por Métricas: Campo de texto e botão 'Procurar'.
 - 2) Relatório de Avaliação por Critérios: Campo de texto e botão 'Procurar'.
- Salvando Relatório(s) no Repositório:**
 - 1) Relatório de Avaliação por Métricas (URL): Campo de texto e botão 'Procurar'.
 - 2) Relatório de Avaliação por Critérios (URL): Campo de texto e botão 'Procurar'.
- Botões de Ação:** 'Salvar' (com ícone de disquete) e 'Sair' (com ícone de porta de saída).

Figura 61. Tela de Armazenamento da Avaliação

O módulo ainda permite que se realize um diagnóstico da avaliação. Esta funcionalidade permite que após o carregamento dos relatórios no repositório seja diagnosticada a avaliação. A interface gráfica desta funcionalidade apresenta um componente que lista todas as avaliações com a situação “Finalizada”. Nesta opção o usuário pode criar um diagnóstico preenchendo alguns campos, como:

- Avaliação: com as avaliações em situação “finalizada”;
- Origem: descrição da fase ou etapa do processo avaliado;
- Item: descrição do item do processo avaliado com algum problema;
- Tipo de Indício: com os tipos de indício que pode caracterizar a adição do item de diagnóstico;
- Ponto forte: aspecto positivo do processo avaliado;
- Ponto fraco: aspecto negativo do processo avaliado;

- Oportunidade de melhoria: possível oportunidade de melhoria do processo avaliado;
- Ameaça: possível ameaça/risco que pode causar problema ou gargalo no processo avaliado;
- Descrição: descrição do problema identificado;
- Resolução: descrição de uma possível resolução para o problema identificado.

Em geral, o diagnóstico dá-se pela adição de vários itens relacionados a uma determinada avaliação. Todos os itens cadastrados podem ser listados pelo botão “Listar”. A Figura 62 apresenta a adição de um item de diagnóstico.

A janela de software intitulada "Adicionando Diagnóstico" apresenta o seguinte layout:

- Título:** Adicionando Diagnóstico
- Campos de Entrada:**
 - Avaliação:** Menu suspenso com o texto "Selecione..."
 - Origem:** Campo de texto simples
 - Item:** Campo de texto simples
 - Tipo Indício:** Menu suspenso com o texto "Selecione..."
 - Descrição:** Área de texto grande para a descrição do problema
 - Resolução:** Área de texto grande para a descrição da resolução
- Botões:**
 - Sugestão:** Botão localizado ao lado do campo de resolução.
 - Barra de Menu:** Localizada na base da janela, contendo os ícones e textos "Listar", "Salvar", "Limpar" e "Sair".

Figura 62. Tela de Adição de Diagnóstico da Avaliação

Através do módulo há a possibilidade também apresentar aos usuários todos os itens de diagnóstico de processos avaliados que foram cadastrados. Há, também, a possibilidade de se listar todos os itens cadastrados ou podem-se realizar consultas considerando dois filtros:

- Por Avaliação: lista os itens de diagnóstico de acordo com a avaliação selecionada no momento da criação do item;
- Por Tipo de Indício: lista os itens de diagnóstico de acordo com o tipo de início selecionado no momento da criação do item.

Independentemente do tipo de consulta selecionada, os itens apresentados são os seguintes: nome da avaliação, origem do processo, item do processo, tipo de indício e descrição. Após a busca na base de dados, é possível selecionar o item de diagnóstico e alterá-lo ou mesmo excluí-lo. Ainda é possível que seja gerado um arquivo PDF com todos os itens relacionados a uma avaliação bastando selecionar um item da consulta. A Figura 63 apresenta como é realizada a consulta aos itens de diagnóstico.

A interface 'Consultando Diagnóstico' possui uma barra de título com o mesmo nome. O conteúdo principal é dividido em seções. A primeira seção, 'Consulta por:', contém duas opções de radio button: 'Avaliação' (selecionada) e 'Tipo Indício'. Cada opção tem um campo de texto 'Selecione a Avaliação:' ou 'Selecione o Tipo:' com uma seta para baixo e um botão 'Listar' com um ícone de lupa. Abaixo, a seção 'Resultados:' contém uma tabela com cinco colunas: 'Avaliação', 'Origem', 'Item', 'Tipo Indício' e 'Descrição'. O corpo da tabela está vazio. Na base da interface, há uma barra com cinco botões: 'Gerar PDF' (com ícone de documento), 'Alterar' (com ícone de ferramenta), 'Listar Todos' (com ícone de lupa), 'Excluir' (com ícone de lixeira) e 'Sair' (com ícone de porta).

Figura 63. Tela de Consulta de Diagnóstico

O módulo disponibiliza ainda que seja gerado um gráfico do diagnóstico com a ilustração da quantidade de indícios cadastrados por

avaliação. Basicamente, através da seleção do tipo de processo e pelo nome da avaliação é possível que se tenha uma base visual do diagnóstico elaborado a partir de uma determinada avaliação. Esta visão é relevante, já que no momento do diagnóstico somente visualiza-se a inserção individual de cada item, ou seja, esta interface gráfica permite entender o panorama de registros de indícios detectados de uma avaliação. A Figura 64 apresenta como o gráfico é gerado.

Normalmente, ao final de cada avaliação os resultados devem ser comunicados aos envolvidos que utilizam o processo avaliado. Na maioria das vezes, a socialização dos resultados ocorre através de reuniões. Dessa forma, foi implementado no módulo um mecanismo que realiza o registro de uma reunião. As Figuras 65 e 66 apresentam, respectivamente, como pode ser elaborada e consultada uma ata de reunião considerando o nome da avaliação.

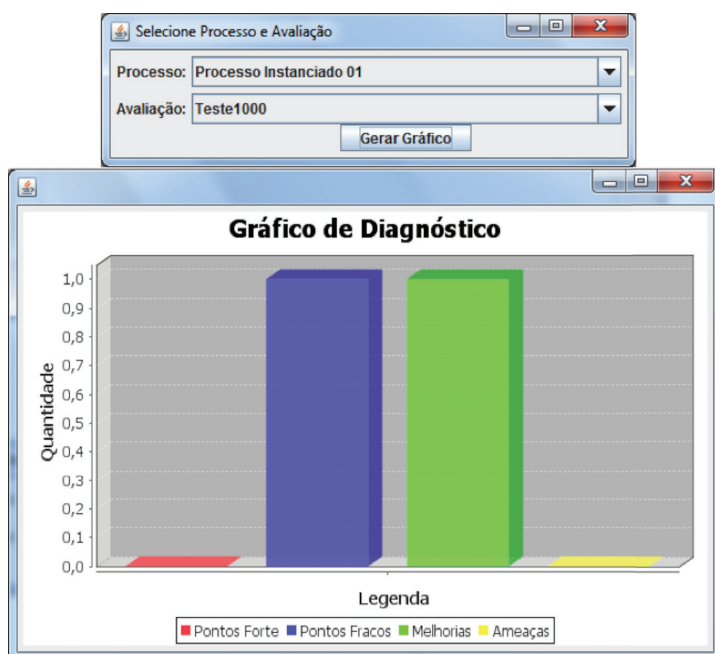


Figura 64. Tela de Geração do Gráfico

Ata da Avaliação

Adicionar Ata

Avaliação: Seleccione...

Nome Ata:

Responsável:

E-mail:

Data: 29/11/2013

1. Informações Iniciais:

2. Participantes:

3. Pautas:

4. Assuntos Tratados:

5. Decisões:

Consultar Salvar Limpar Sair

Figura 65. Tela de Elaboração de Ata de Reunião

Consultando Ata

Consulta por:

☐ Avaliação

Selecione a Avaliação: Seleccione...

Listar

Resultados:

Nome	Avaliação	Responsável	Data
------	-----------	-------------	------

Gerar PDF Alterar Listar Todas Excluir Sair

Figura 66. Tela de Consulta de Ata de Reunião

12.2.2. A Ferramenta Spider-PI

A ferramenta Spider-PI é responsável por sistematizar as atividades de melhoria de processos de acordo com os modelos e norma de qualidade. Como o módulo de avaliação, ela foi elaborada a partir de estudos preliminares, elaboração de mapeamento do processo entre os modelos e norma de qualidade, e construção do framework de processo. Dentre as principais funcionalidades desta ferramenta podem-se citar: criação, inicialização, visualização, execução, armazenamento e finalização da solução de melhoria. Esta ferramenta possui as seguintes características:

- Criação de Nova Melhoria: responsável por possibilitar cadastro de uma nova melhoria. Nesta opção o usuário pode criar uma melhoria preenchendo alguns campos de registro;
- Iniciação da Melhoria: possibilita que a melhoria seja inicializada. A interface gráfica desta funcionalidade apresenta um componente que lista todas as melhorias cadastradas, que automaticamente podem ser iniciadas;
- Visualização das Melhorias: responsável por apresentar aos usuários todas as melhorias de processos que foram cadastradas;
- Definição da Razão da Melhoria: possibilita aos usuários que seja cadastrado a razão/objetivo da melhoria;
- Definição das Práticas da Melhoria: provê que os usuários definam as práticas necessárias para se solucionar a melhoria;
- Definição do Contexto da Melhoria: possibilita que os usuários cadastrem os objetivos, trabalhos existentes e os benefícios a serem alcançadas pela melhoria;
- Definição do Apoio à Melhoria: possibilita que o representante da organização denominado de patrocinador pela ferramenta realize a aprovação da continuação da execução da melhoria efetivando seu apoio;

- Definição de Infraestrutura a Melhoria: responsável por registrar as informações sobre a alocação de infraestrutura para melhoria, como por exemplo, o esforço humano que deverá ser despendido para solucionar a melhoria;
- Caracterização das Práticas de Melhoria: possibilita que seja caracterizado o estado atual da prática e o estado que se almeja alcançar com a execução da melhoria;
- Definição dos Procedimentos da Melhoria: possibilita que os usuários estabeleçam os procedimentos que deverão ser adotados para a conclusão da solução de melhoria;
- Definição das Prioridades da Execução das Práticas: responsável por estabelecer a ordem de prioridade de execução das práticas da melhoria;
- Definição do Planejamento da Melhoria: possibilita que seja realizado o planejamento da execução da melhoria;
- Criação da Solução da Melhoria: responsável pela elaboração da solução da melhoria;
- Teste da Solução de Melhoria: responsável por testar a solução da melhoria elaborada;
- Implantação da Solução de Melhoria: possibilita que a solução da melhoria elaborada seja implantada;
- Análise e Validação da Melhoria: responsável por validar a melhoria propriamente dita;
- Elaboração de Ações Futuras: possibilita que seja registrada ações futuras que podem ser realizadas em relação àquela melhoria.

Para utilizar a Spider-PI é necessário inicialmente que o usuário realize sua autenticação na ferramenta por meio de *login* e senha. Dentro da ferramenta, o usuário poderá acessar suas funções através do menu Arquivo e Serviços.

Acessando o menu Arquivo é possível cadastrar uma nova melhoria. Ela é a funcionalidade responsável por realizar o cadastro de uma nova melhoria, possibilitando que o usuário preencha alguns campos como: nome, descrição e responsável. A Figura 67 mostra como uma melhoria é registrada/cadastrada.

Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software

Nova Melhoria Contínua

Nome:

Descrição:

Data do Registro: 29/11/2013

Situação: Nova

Responsável:

Membros Disponíveis:

Membro	Perfil
--------	--------

Membros Alocados:

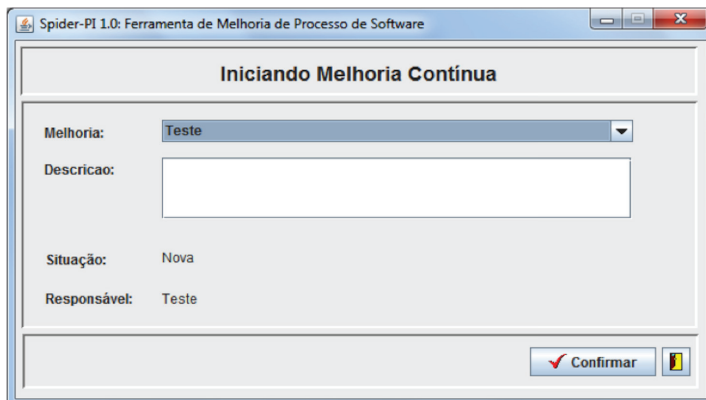
Membro	Perfil
--------	--------

Botões de navegação: [seta para direita] [seta para esquerda]

Barra de ferramentas: [Listar] [Cancelar] [Alterar] [Remover] [Inserir] [Ajuda]

Figura 67. Tela de Criação de Nova Melhoria

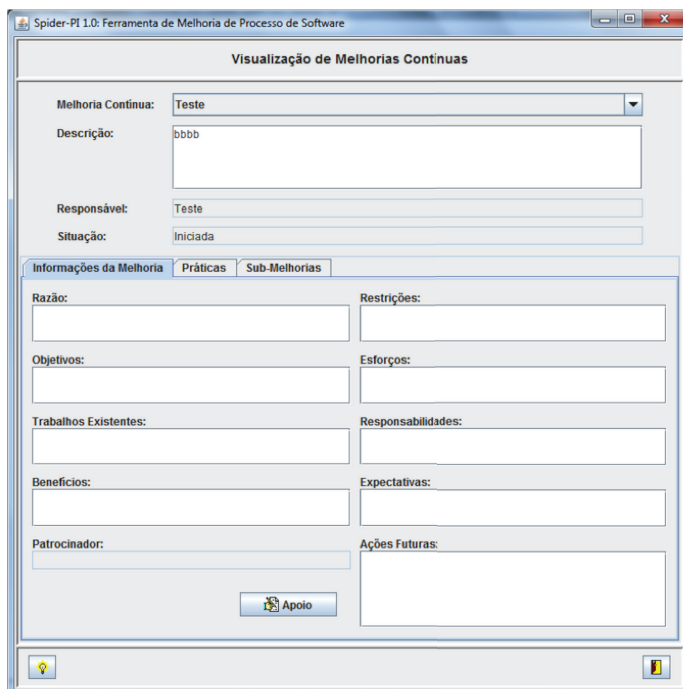
Após a melhoria ser criada, ela pode ser inicializada. Este processo inicia os passos para a execução da mesma seguindo-se as atividades definidas no *framework* de processo. A Figura 68 apresenta a interface gráfica que realiza a inicialização da melhoria.



A screenshot of the 'Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software' window. The title bar shows the application name and standard Windows window controls. The main window has a title 'Iniciando Melhoria Contínua'. Below the title, there are four labeled fields: 'Melhoria:' with a dropdown menu showing 'Teste', 'Descricao:' with a large empty text box, 'Situação:' with the value 'Nova', and 'Responsável:' with the value 'Teste'. At the bottom right, there is a 'Confirmar' button with a checkmark icon and a small yellow icon.

Figura 68. Tela de Inicialização da Melhoria

Com as melhorias sendo cadastradas é possível que se realize a visualização das informações que são inseridas em cada uma delas. A interface gráfica da Figura 69 mostra os dados da melhoria que estiver selecionada.



A screenshot of the 'Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software' window showing the 'Visualização de Melhorias Contínuas' screen. The title bar is the same. The main window has a title 'Visualização de Melhorias Contínuas'. Below the title, there are four labeled fields: 'Melhoria Contínua:' with a dropdown menu showing 'Teste', 'Descrição:' with the value 'bbbb', 'Responsável:' with the value 'Teste', and 'Situação:' with the value 'Iniciada'. Below these fields, there are three tabs: 'Informações da Melhoria', 'Práticas', and 'Sub-Melhorias'. The 'Informações da Melhoria' tab is selected. It contains a grid of fields: 'Razão:', 'Restrições:', 'Objetivos:', 'Esforços:', 'Trabalhos Existentes:', 'Responsabilidades:', 'Benefícios:', 'Expectativas:', 'Patrocinador:', and 'Ações Futuras:'. Each field has a corresponding text box. At the bottom center, there is an 'Apoio' button with a lightbulb icon. At the bottom right, there is a small yellow icon.

Figura 69. Tela de Visualização das Melhorias

Após serem inicializadas, as melhorias contínuas passam para o estágio de execução. Através do menu Serviços da ferramenta é possível prosseguir com a execução das demais atividades para se alcançar a solução de melhoria. Seguindo o raciocínio do fluxo do *framework*, as próximas interfaces gráficas/funcionalidades ilustrarão o cadastro de informações relacionadas à execução da solução de melhoria propriamente dita. A tela da Figura 70 mostra como pode ser definida a razão ou motivo da melhoria.

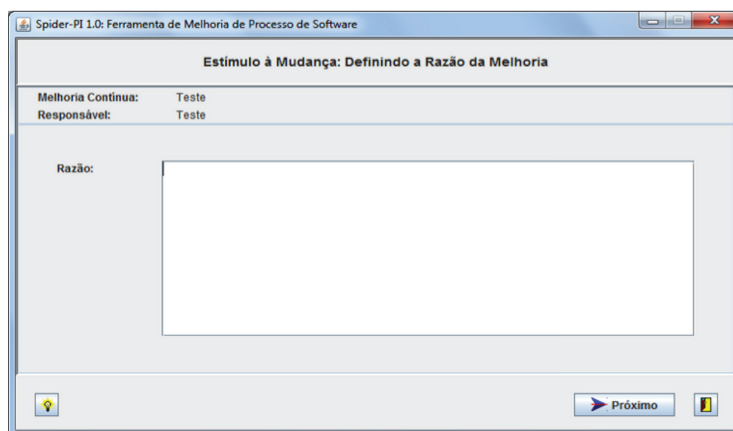
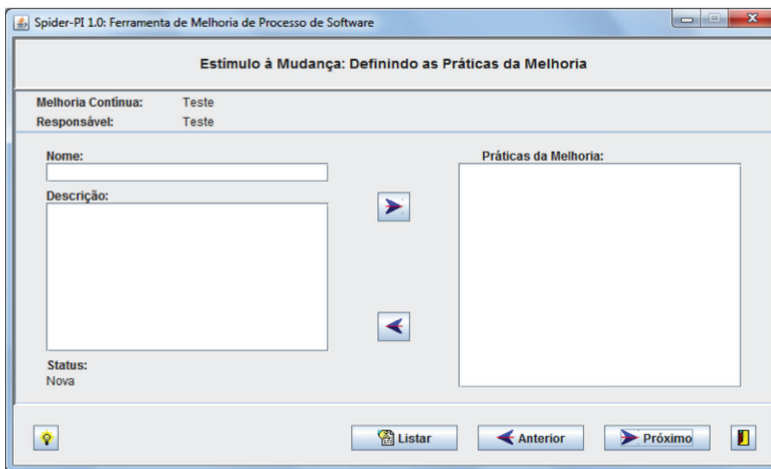


Figura 70. Tela de Definição da Razão da Melhoria

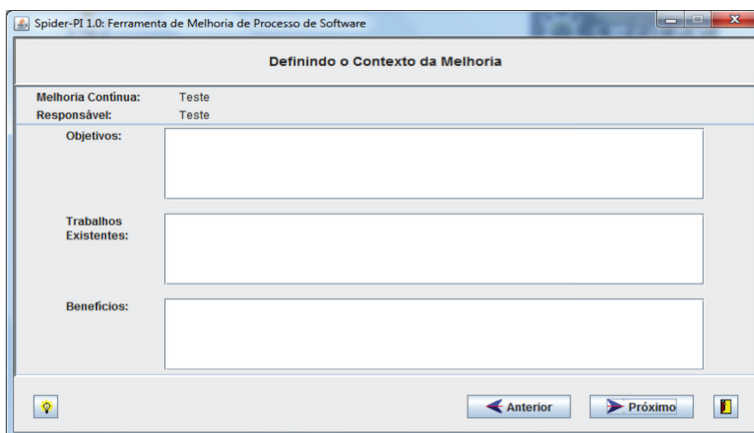
Em seguida é possível que seja cadastrada as práticas que auxiliarão a elaboração da solução da melhoria. A Figura 71 apresenta como as práticas são definidas.



A screenshot of the 'Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software' window. The title bar shows the application name. The main window has a title 'Estímulo à Mudança: Definindo as Práticas da Melhoria'. Below this, there are two rows of labels and values: 'Melhoria Continua: Teste' and 'Responsável: Teste'. The main area is divided into two sections. On the left, there are two text input fields labeled 'Nome:' and 'Descrição:', with a 'Status:' label below them showing 'Nova'. To the right of these fields are two arrow buttons, one pointing right and one pointing left. On the right side of the window, there is a large empty text area labeled 'Práticas da Melhoria:'. At the bottom, there is a toolbar with a lightbulb icon, a 'Listar' button with a magnifying glass icon, and 'Anterior' and 'Próximo' buttons with left and right arrow icons respectively. A small document icon is also present on the far right of the toolbar.

Figura 71. Tela de Definição das Práticas da Melhoria

Após a razão e as práticas serem definidas, o contexto da melhoria deve ser estabelecido. Este passo é o momento onde são cadastrados os objetivos da melhoria, o que já existe de trabalho que pode contribuir para a solução de melhoria e os benefícios que a melhoria pode agregar ao processo. A Figura 72 ilustra a interface gráfica desse cadastro.



A screenshot of the 'Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software' window. The title bar shows the application name. The main window has a title 'Definindo o Contexto da Melhoria'. Below this, there are two rows of labels and values: 'Melhoria Continua: Teste' and 'Responsável: Teste'. The main area is divided into three sections, each with a label and a large text input field: 'Objetivos:', 'Trabalhos Existentes:', and 'Benefícios:'. At the bottom, there is a toolbar with a lightbulb icon, 'Anterior' and 'Próximo' buttons with left and right arrow icons respectively, and a small document icon on the far right.

Figura 72. Tela de Definição do Contexto da Melhoria

Outro passo não menos importante é cadastrar os dados de apoio à melhoria. Esse “apoio” deve ser proporcionado por um representante da organização que será apelidado de “patrocinador”, que ficará responsável por aprovar ou reprovar a continuação da execução da solução de melhoria. A Figura 73 apresenta a interface gráfica que define os dados do patrocinador.

A interface gráfica, intitulada "Construindo Apoio à Melhoria", pertencente ao "Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software", apresenta os seguintes elementos:

- Campos de texto para "Melhoria Continua:" (contendo "Teste") e "Responsável:" (contendo "Teste").
- Um campo de texto para "Patrocinador:".
- Um campo de texto para "E-Mail:".
- Um menu suspenso para "Status:" com a opção "Iniciada" selecionada.
- Dois botões de ação: "Notificar" (com ícone de envelope) e "Apoio" (com ícone de mão).
- Na barra inferior, há um ícone de lâmpada, um botão "Anterior" com uma seta para a esquerda e um botão "Próximo" com uma seta para a direita.

Figura 73. Tela de Definição do Apoio à Melhoria

Finalizada a parte inicial de criação, definição e aprovação da execução da solução de melhoria, segue-se os demais passos para a efetivação da construção da melhoria. Após a aprovação da execução realizada pelo patrocinador, os dados de alocação de infraestrutura da melhoria devem ser cadastrados e definidos. A Figura 74 apresenta o que deve ser alocado para a execução da melhoria.

A screenshot of the 'Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software' window. The title bar shows the application name. The main window has a title 'Alocando Infra-Estrutura da Melhoria'. Below the title, there are two labels: 'Melhoria Contínua: Teste' and 'Responsável: Leonardo'. The main area is divided into four sections: 'Restrições:', 'Esforços:', 'Responsabilidades:', and 'Expectativas'. Each section has a large empty text box for input. At the bottom right, there is a 'Próximo' button with a right-pointing arrow. There are also help icons (question marks) at the bottom left and right.

Figura 74. Tela de Alocação de Infraestrutura para a Melhoria

Após este procedimento, o estado das práticas deverá ser definido. Esta funcionalidade trata da descrição do estado atual da prática e o estado que se almeja alcançar. A Figura 75 mostra como ocorre o cadastro das práticas.

A screenshot of the 'Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software' window. The title bar shows the application name. The main window has a title 'Caracterizando Estados das Práticas da Melhoria'. Below the title, there are two labels: 'Melhoria Contínua: Teste' and 'Responsável: Leonardo'. The main area has three input fields: 'Prática(s):' with a dropdown arrow, 'Estado Atual:', and 'Estado Desejado:'. Below these fields is a 'Validação' button with a checkmark icon. At the bottom, there are five buttons: 'Inserir' (with a plus icon), 'Listar' (with a list icon), 'Anterior' (with a left arrow), 'Próximo' (with a right arrow), and a help icon (question mark) on the far right.

Figura 75. Tela de Caracterização das Práticas da Melhoria

Com as práticas definidas e caracterizadas, passa-se para a elaboração dos procedimentos das práticas. Este é o momento onde se descreve como determinada prática será executada. A Figura 76 ilustra como são definidos os procedimentos.

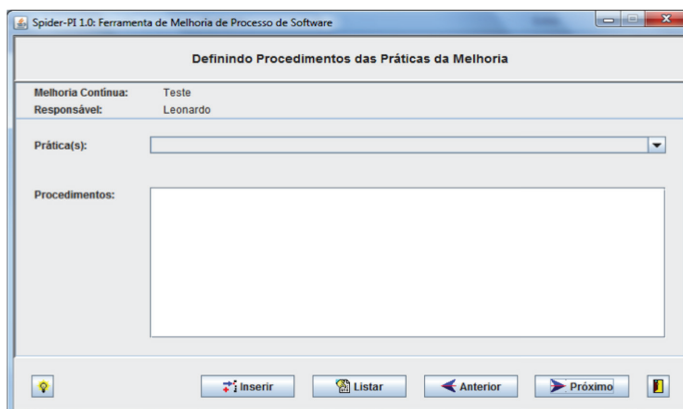


Figura 76. Tela de Definição dos Procedimentos das Práticas da Melhoria

Em situações em que se há mais de uma prática cadastrada, é possível que se priorize a execução das mesmas. Neste passo, a Spider-PI associa a cada prática uma sub-melhoria que terá sua execução planejada e que, posteriormente, irá contar com uma proposta de solução. Em resumo, a prática é definida e associada a uma sub-melhoria que estará diretamente associada a uma melhoria. A Figura 77 apresenta a interface gráfica definindo a prioridade de execução das práticas de melhoria bem como o cadastro da sub-melhoria.

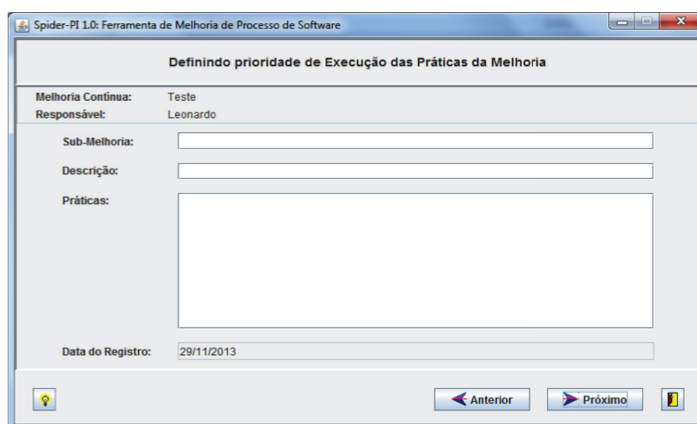


Figura 77. Tela de Definição da Prioridade de Execução das Práticas da Melhoria

Em seguida, é realizado o planejamento de execução da sub-melhoria. A Figura 78 mostra as interfaces gráficas nas quais são definidas e cadastradas as informações deste planejamento.

The figure displays three sequential screenshots of the Spider-PI 1.0 software interface, specifically the 'Planejando a Execução da Sub-Melhoria' (Planning the Execution of the Sub-Improvement) screen. Each screenshot shows a form with the following fields:

- Melhoria Contínua:** Melhoria Teste
- Responsável:** Leonardo
- Sub-Melhoria:** Sub-Melhoria 1

The top screenshot shows the initial form with fields for 'Recursos', 'Métricas', 'Tracking', 'Estratégias', and 'Observações'. The middle screenshot shows the 'Atividade' section with 'Data de Início' and 'Data Final' fields, and a large 'Atividades da Sub-Melhoria' area. The bottom screenshot shows the 'Risco' section with 'Mitigação' and 'Probabilidade' fields, and a large 'Riscos da Execução da Sub-Melhoria' area. All three screenshots have a title bar 'Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software' and a footer with 'Listar', 'Anterior', and 'Próximo' buttons.

Figura 78. Telas de Elaboração do Planejamento da Execução da Sub-melhoria

Após a elaboração do planejamento de execução, a sub-melhoria então tem sua solução definida. A Figura 79 mostra as interfaces gráficas de como a solução é cadastrada.

Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software

Criando a solução para a Sub-Melhoria

Melhoria Contínua: Melhoria Teste
 Responsável: Leonardo
 Sub-Melhoria: Sub-Melhoria 1

Solução:

Ferramentas:

Processos:

Habilidades:

Anterior Próximo

Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software

Criando a solução para a Sub-Melhoria

Melhoria Contínua: Melhoria Teste
 Responsável: Leonardo
 Sub-Melhoria: Sub-Melhoria 1

Conhecimentos:

Ajuda Externa:

Observações:

Anterior Próximo

Figura 79. Telas de Cadastro da Solução para a Sub-melhoria

A solução para a sub-melhoria pode e deve ser testada antes de ser implantada. A Figura 80 ilustra a interface gráfica de teste da solução.

A screenshot of the 'Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software' window. The title bar shows the application name and standard Windows window controls. The main window has a title 'Testando a solução proposta para a Sub-Melhoria'. Below the title, there are three labels: 'Melhoria Contínua:' with value 'Melhoria Teste', 'Responsável:' with value 'Leonardo', and 'Sub-Melhoria:' with value 'Sub-Melhoria 1'. Below these is a 'Teste:' label followed by a large empty text box. Then, 'Data de Início:' followed by a date input field, and 'Data Final:' followed by a date input field. To the right of the date fields is a 'Status:' label with a dropdown menu showing 'Não atendido'. Below the date fields is an 'Observações:' label followed by a large empty text box. At the bottom, there is a toolbar with icons for help, insert, list, previous, next, and a lightbulb icon.

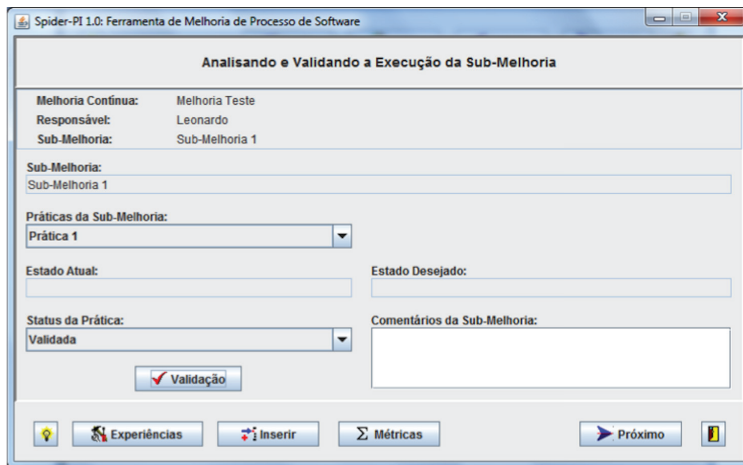
Figura 80. Tela de Teste da Solução para a Sub-melhoria

Finalizados os testes, a solução da sub-melhoria poderá ou não ser implantada. A Figura 81 apresenta tal aprovação ou reprovação.

A screenshot of the 'Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software' window. The title bar shows the application name and standard Windows window controls. The main window has a title 'Implantando a solução proposta para a Sub-Melhoria'. Below the title, there are three labels: 'Melhoria Contínua:' with value 'Melhoria Teste', 'Responsável:' with value 'Leonardo', and 'Sub-Melhoria:' with value 'Sub-Melhoria 1'. Below these is a 'Solução:' label followed by a text box containing 'Solução 1'. Then, a 'Status:' label followed by a dropdown menu showing 'Implantada'. Below the dropdown menu is a list box containing 'Implantada', 'Implantada', and 'Não Implantada'. At the bottom, there is a toolbar with icons for help, previous, next, and a lightbulb icon.

Figura 81. Tela de Implantação da Solução da Sub-melhoria

A parte final da execução da melhoria trata-se de realizar a análise e validação da sub-melhoria. Neste ponto é possível que se valide ou não a sub-melhoria/prática. A Figura 82 ilustra como é realizada esta validação.

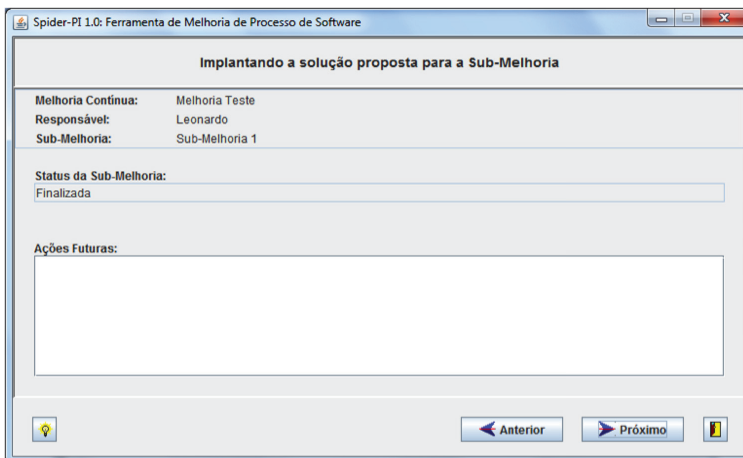


A interface da Spider-PI 1.0, intitulada "Ferramenta de Melhoria de Processo de Software", apresenta a seguinte estrutura:

- Título da Janela:** Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software
- Seção de Cabeçalho:** Analisando e Validando a Execução da Sub-Melhoria
- Informações de Contexto:**
 - Melhoria Contínua: Melhoria Teste
 - Responsável: Leonardo
 - Sub-Melhoria: Sub-Melhoria 1
- Formulário de Análise:**
 - Sub-Melhoria: Sub-Melhoria 1 (campo de texto)
 - Práticas da Sub-Melhoria: Prática 1 (menu suspenso)
 - Estado Atual: (campo de texto)
 - Estado Desejado: (campo de texto)
 - Status da Prática: Validada (menu suspenso)
 - Comentários da Sub-Melhoria: (área de texto grande)
- Botões de Ação:**
 - Validação (botão com ícone de checkmark)
 - Experiências (botão com ícone de lâmpada)
 - Inserir (botão com ícone de setas)
 - Métricas (botão com ícone de gráfico)
 - Próximo (botão com ícone de seta para a direita)

Figura 82. Tela de Análise e Validação da Execução da Sub-melhoria

Por fim, caso seja necessário, é possível ainda registrar ações futuras que podem ser realizadas a respeito daquela sub-melhoria/melhoria. A seguir, é apresentada na Figura 83 a interface gráfica que faz este registro.



A interface da Spider-PI 1.0, intitulada "Ferramenta de Melhoria de Processo de Software", apresenta a seguinte estrutura:

- Título da Janela:** Spider-PI 1.0: Ferramenta de Melhoria de Processo de Software
- Seção de Cabeçalho:** Implantando a solução proposta para a Sub-Melhoria
- Informações de Contexto:**
 - Melhoria Contínua: Melhoria Teste
 - Responsável: Leonardo
 - Sub-Melhoria: Sub-Melhoria 1
- Status da Sub-Melhoria:** Finalizada (campo de texto)
- Formulário de Registro:**
 - Ações Futuras: (área de texto grande)
- Botões de Navegação:**
 - Anterior (botão com ícone de seta para a esquerda)
 - Próximo (botão com ícone de seta para a direita)

Figura 83. Tela de Análise e Validação da Execução da Sub-melhoria



Capítulo 13

Implementação do Processo de Gerência de Reutilização

em [Alho et al., 2011]

Fabício de Medeiros Alho
Kleverton Oliveira Macedo
Sandro Ronaldo Bezerra Oliveira

O desenvolvimento de um software é algo que tem evoluído de uma estrutura monolítica, centralizada e estática para uma estrutura modular, distribuída e dinâmica, tanto no nível de processo quanto no de produto. Neste contexto, o reuso de software vem sendo discutido, estudado e implantado nas empresas que buscam se adaptar a uma realidade mais qualitativa e competitiva.

Segundo Murta (2008), a reutilização aplicada ao domínio de software parte do pressuposto de que grande parte dos sistemas desenvolvidos não é totalmente nova, eles apenas representam variações de sistemas desenvolvidos anteriormente. Isso faz com que muitas organizações desenvolvam sistemas baseados em determinadas linhas de negócio, denominadas domínio de aplicação.

Para Frakes (2005), o propósito da reutilização envolve melhorar a qualidade do produto e a produtividade durante o seu desenvolvimento. A qualidade do produto é assegurada através da utilização de ativos que já foram previamente testados, aprovados e reutilizados em outros projetos. A reutilização

destes ativos faz com que a produtividade aumente, diminuindo o tempo de atendimento ao mercado (*time-to-market*) e aumentando a competitividade da organização. Para atingir estes objetivos é necessário definir uma estratégia que torne a reutilização sistemática e parte das atividades cotidianas da organização, baseando-se em modelos de maturidade de software consolidados e que sejam compatíveis com normas técnicas existentes.

Neste contexto, boas práticas para reutilização de software são encontradas em modelos e normas de qualidade, como o MR-MPS-SW e nas normas ISO/IEC 12207 e IEEE 1517 [IEEE, 1999], que norteiam a implantação e sistematização de um processo de gerência de reutilização nas organizações. Para tanto, o sucesso desta implantação e sistematização também depende do uso de ferramentas de software que apoiem as atividades desenvolvidas pelas empresas na aplicação do reuso de software, tornando-as aderentes aos modelos e normas de qualidade citados.

Este trabalho objetiva apresentar a Spider-Reuse, uma solução de software livre para apoio a uma abordagem de reutilização baseada no modelo MR-MPS-SW e nas normas ISO/IEC 12207 e IEEE 1517.

13.1. A Abordagem de Reutilização

Segundo Pietro-Díaz (1991), o objetivo principal da reutilização de artefatos de software é reduzir o tempo e o esforço necessários para construir sistemas. Além disso, a reutilização proporciona benefícios como confiabilidade, facilidade na manutenção, entre outros.

Além das vantagens inerentes à reutilização de software, é notória que a demanda por profissionais e empresas qualificadas na área de desenvolvimento de software tem levado a uma busca globalizada e crescente pelos modelos/normas de qualidade, visando uma maior maturidade nos processos de software instituídos. A criação e a evolução desta maturidade demandam um esforço significativo entre os envolvidos, assim como uma escalada difícil na

curva de aprendizado que normalmente se apresenta, principalmente em processos de software que, apesar de possuírem uma base em modelos de maturidade consolidados e que sejam compatíveis com normas técnicas existentes, não estão consideravelmente disseminados entre as empresas que desenvolvem softwares, diminuindo o acervo de lições aprendidas na implantação e manutenção destes processos.

Existem vários modelos e normas de qualidade que caracterizam e definem a reutilização de software, como o modelo MR-MPS-SW que define um processo de gerência de reutilização no nível de maturidade E (parcialmente definido), cujo propósito envolve gerenciar o ciclo de vida de ativos reutilizáveis. A norma ISO/IEC 12207 estabelece uma estrutura comum para os processos de ciclo de vida de software, com terminologia bem definida, que pode ser referenciada pela indústria de software. A norma IEEE 1517 é uma extensão da norma ISO/IEC 12207 e define procedimentos, atividades, e as tarefas que permitem a identificação, a construção, a manutenção, e a gerência de ativos reutilizáveis.

Por meio de um mapeamento entre o modelo MR-MPS-SW, a norma ISO/IEC 12207 e a norma IEEE 1517, foi elaborada uma abordagem que auxilia as empresas interessadas na sistematização do processo de gerência de reutilização, contribuindo na aplicação da maturidade para este processo, disponível em www.spider.ufpa.br/projetos/spider_reuse/mapeamento_reuso.zip. O mapeamento realizado permitiu a identificação e o tratamento de *gaps* existentes entre o modelo e as normas estudadas. A ferramenta Spider-Reuse tem o propósito de apoiar a utilização da referida abordagem, a partir de um fluxo de atividades que contempla as boas práticas resultantes do mapeamento realizado. O fluxo encontra-se, de maneira macro, apresentado na Figura 84, descrito usando as notações do BPMN. Um melhor entendimento do fluxo pode ser obtido em www.spider.ufpa.br/projetos/spider_reuse/framework_reuso.zip.

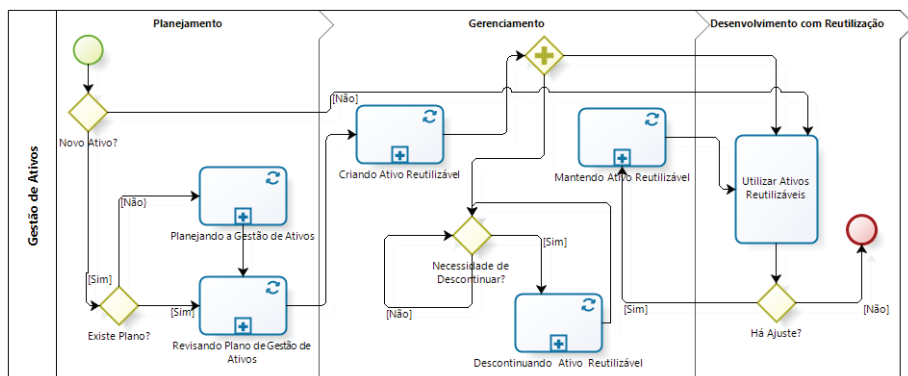


Figura 84. Macro Fluxo da Abordagem de Reutilização

A abordagem apresentada divide o processo de gerência de reutilização em três macro-atividades: Planejando a Gestão de Ativos, Gerenciando Ciclo de Vida do Ativo e Desenvolvendo com Reutilização. Um detalhamento destas macro-atividades é apresentado abaixo, bem como os resultados esperados do MR-MPS-SW, vide detalhamento em [SOFTEX, 2012a], que estas atividades são aderentes:

- **Planejando a Gestão de Ativos:** o planejamento abrange desde a definição organizacional para ativo, dos critérios de aceitação, certificação, classificação, descontinuidade e de qualidade de ativos, até a apresentação do plano de gestão de ativos à gerência de configuração, o armazenamento deste plano no repositório organizacional e a comunicação dos envolvidos. Esta macro-atividade está aderente aos resultados esperados GRU1, GRU4 e GRU5;
- **Gerenciando Ciclo de Vida do Ativo:** o gerenciamento contempla a criação de novos ativos reutilizáveis, por meio da avaliação dos critérios de aceitação e certificação definidos, e a classificação destes ativos de acordo com os critérios elaborados. Contempla também o controle de modificações do ativo, por meio da avaliação de solicitações de mudança apresentadas, da gestão de configuração e

da comunicação dos envolvidos. O gerenciamento compõe ainda a descontinuação dos ativos segundo critérios estabelecidos. Esta macro-atividade está aderente aos resultados esperados GRU1, GRU2, GRU3, GRU4 e GRU5;

- **Desenvolvendo com Reutilização:** o desenvolvimento abrange a necessidade do registro do reuso do ativo e do *feedback* de utilização do mesmo. Encontra-se aderente ao resultado esperado GRU3.

O detalhamento das macro-atividades apresentadas foi modelado de acordo com o mapeamento proposto. A ferramenta Spider-Reuse utiliza o conceito de ator para especificar quem realiza determinada atividade no processo de reutilização. Existem cinco tipos de atores:

- **Gerente de Ativos:** detém a maior parte das atividades da abordagem como a elaboração do plano de gestão de ativos, a avaliação de critérios de aceitação e certificação de novos ativos, a análise de solicitações de mudanças e a decisão na descontinuação dos ativos;
- **Engenheiro de Domínio:** é responsável pela classificação dos ativos, pelo armazenamento dos ativos na biblioteca organizacional, pelo planejamento e execução das modificações nos ativos reutilizáveis, assim como pelo cumprimento do plano de comunicação;
- **Consumidor de Ativos:** deverá registrar o reuso dos ativos e o *feedback* de utilização dos mesmos;
- **Gerente de Configuração:** deve avaliar o plano de gestão de ativos e os ativos reutilizáveis pela gerência de configuração;
- **Gerente de Qualidade:** deve avaliar todos os artefatos manipulados na abordagem pela garantia de qualidade.

É importante ressaltar que um usuário da ferramenta pode representar mais de um ator, dependendo do grau de envolvimento com a abordagem de reutilização.

13.2. A Ferramenta Spider-Reuse

A ferramenta Spider-Reuse é uma ferramenta de licença GPL voltada especificamente para apoiar a abordagem apresentada na Seção 13.1, o qual traz uma solução para sistematizar o processo de gestão de reuso de ativos de software em uma organização. Esta ferramenta é parte integrante do Projeto SPIDER.

Como já apresentado, ter um processo de reuso de ativos de software estabelecido em recomendações e boas práticas contribui significativamente na garantia da qualidade e produtividade do produto desenvolvido. Dentro deste contexto, a Spider-Reuse busca auxiliar as organizações interessadas no processo de gerência de reutilização de ativos de software, oferecendo um acompanhamento sistematizado de todas as fases envolvidas no ciclo de vida de um ativo reutilizável de software.

A ferramenta pode ser utilizada em diferentes organizações, independentemente do tamanho da mesma, e encontra-se disponível em *www.spider.ufpa.br/projetos/spider_reuse/Spider-Reuse.zip*.

A ferramenta Spider-Reuse é gratuita e de código aberto, desenvolvida para *desktop*. A ferramenta pode ser utilizada em diferentes organizações, independentemente do tamanho da mesma, e encontra-se disponível em *www.spider.ufpa.br/projetos/spider_reuse/Spider-Reuse.zip*, e é:

- Baseada em Modelos e Guias – é aderente aos resultados esperados do processo de Gerência de Reutilização do MR-MPS-SW, sendo também aderente às atividades da norma ISO/IEC 12207 e das tarefas da IEEE 1517;
- Fornece evidências – emite relatórios técnicos, gerenciais e os planos envolvidos no reuso de ativos de software. Além disto, possibilita a centralização dos artefatos que evidenciam o ciclo de vida de ativos reutilizáveis. Desta forma, a organização pode fazer uso dos mesmos para que, em uma possível avaliação oficial do MPS.BR

ou outro modelo adotado pela organização, os artefatos possam ser utilizados como indicadores de que a organização está implementando o processo de gerência de reutilização de software;

- Integração com ferramentas – a ferramenta Spider-Reuse é integrada a ferramentas livres bem estabelecidas no mercado, como exemplo, o Redmine e o SVN. Pretende-se, com isso, que o impacto da utilização da Spider-Reuse em uma organização seja minimizado, pois existe uma boa chance desta organização já ter trabalhado com uma dessas ferramentas, diminuindo, assim, o esforço despendido com o aprendizado necessário para utilizar a ferramenta Spider-Reuse;
- Possui a customização de critérios objetivos – com o auxílio da ferramenta Spider-CL, é permitido à organização adquirente criar seus próprios critérios objetivos que serão usados nas atividades que necessitam deste nível de imparcialidade, não estando a organização "presa" a um pré-determinado conjunto de critérios estabelecidos na ferramenta.

A ferramenta possui dois módulos funcionais, divididos em: **Administração**, que abrange a configuração das integrações necessárias, os cadastros básicos como unidade organizacional, categorias, subcategorias e palavras-chaves, usuários, perfis e os controles de acesso ao sistema; o módulo de **Execução**, que envolve do planejamento da gestão do reuso de ativos ao desenvolvimento com reutilização, organizado nas 3 (três) fases propostas pelo *framework* do processo. Na seção seguinte serão apresentadas algumas das principais funcionalidades da Spider-Reuse.

13.2.1. Uma Visão Geral da Ferramenta

A Figura 85 apresenta a tela inicial da ferramenta Spider-Reuse, possuindo um menu superior agrupando funções de configurações de integração com

outras ferramentas (credenciais de acesso as ferramentas Redmine, Spider-CL e Repositório de Ativos), um menu lateral que permite a navegação das funcionalidades de administração e de execução descritas a seguir.

Para utilizar a ferramenta, o usuário necessita de autenticação por meio de *login* e senha. De acordo com o perfil de acesso do usuário, a ferramenta montará a árvore de menu lateral com as atividades que o usuário tem permissão de operar.

A ferramenta também implementa as restrições definidas no fluxo do *framework*, onde cada atividade tem que ser executada em uma ordem de dependência, bem como as fases propostas no *framework* do processo.

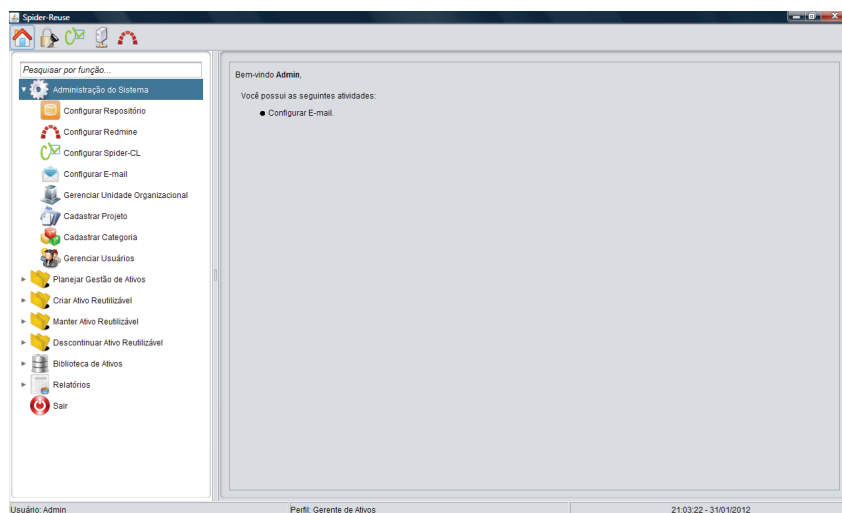


Figura 85. Tela Principal da Spider-Reuse

A primeira atividade da ferramenta possibilita a integração com as ferramentas externas. A Figura 86 exibe esta funcionalidade do sistema, sendo notória a simplicidade da função do ponto de vista do usuário. Para a configuração de cada ferramenta externa, se faz necessário registrar a URL da ferramenta, o *host* do banco de dados, a porta de acesso ao banco, o nome, usuário e senha de acesso ao banco de dados. Estas configurações são necessárias para a integração com as ferramentas Spider-CL e Redmine.

Para acesso ao repositório organizacional de ativos de software, são necessários os registros do *host*, do caminho, da porta e do protocolo de acesso ao repositório, além das credenciais de acesso.

A integração também é forte o suficiente para garantir que a complexidade seja totalmente transparente para o usuário, onde, a partir de um clique, o usuário estará "logado" nas ferramentas externas e com a função pretendida aberta na tela.

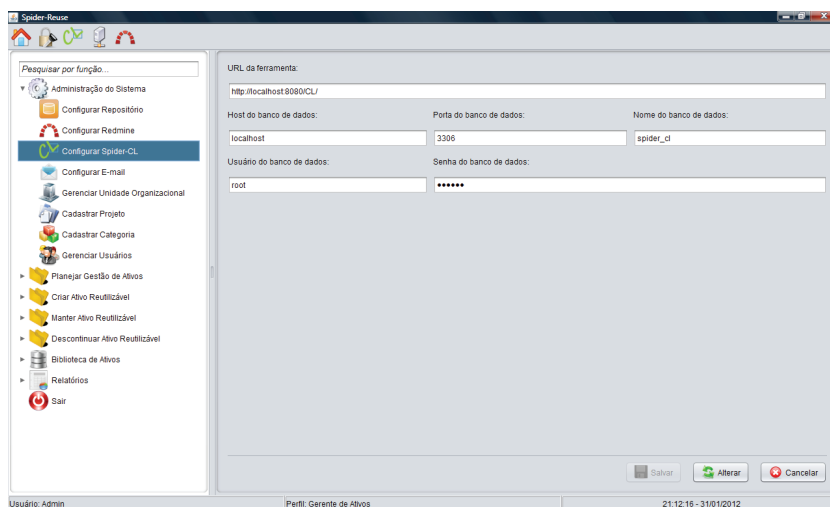


Figura 86. Tela de Configuração de Integração com Ferramentas Externas

Antes de realizar o planejamento da gestão de reuso de ativos de software, alguns cadastros básicos precisam ser registrados. O cadastro da Unidade Organizacional permite o registro de dados da gestão organizacional, assim como a indicação de localização do Plano de Gerência de Configuração da organização, conforme a Figura 87.

O cadastro de projetos permite uma visão mais granular do reuso de ativos de software, pois um ativo reutilizável pode ser de origem organizacional, assim como de projeto. Neste caso, se o ativo reutilizável for originado por um projeto específico, este deve ser registrado na ferramenta Spider-Reuse, conforme Figura 88.

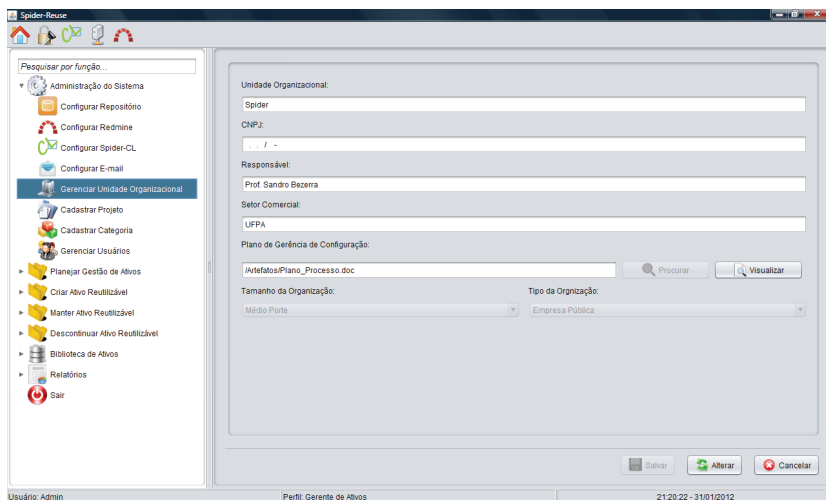


Figura 87. Tela de Configuração da Unidade Organizacional

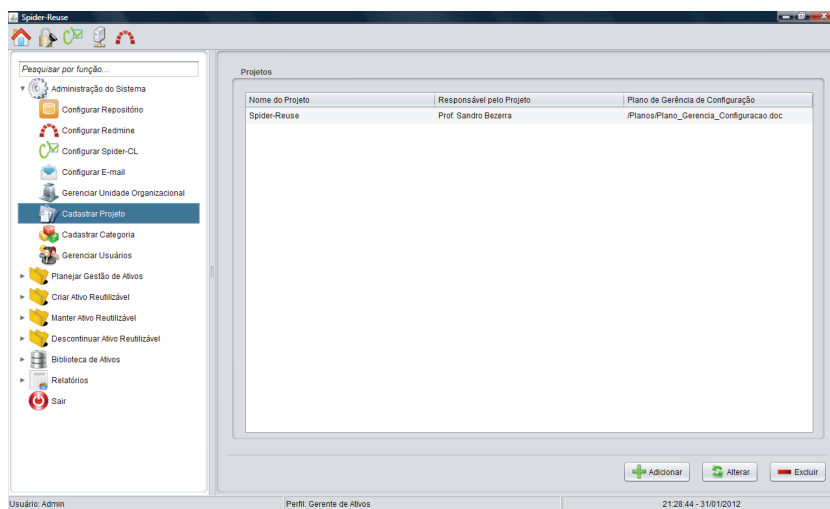


Figura 88. Tela de Cadastro de Projetos

O cadastro de categorias permite a criação de uma rede de classificação de ativos reutilizáveis, facilitando a busca dos mesmos. Nesta rede de classificação, são definidas palavras-chaves que tornam representativos os vários contextos abordados pelo reuso de ativos de software. Uma palavra-chave

pode estar vinculada a outra palavra-chave, aumentando a abrangência de busca e facilitando a consulta por contexto.

Uma categoria criada pode abranger diversas subcategorias, permitindo uma melhor definição organizacional dos contextos trabalhados. Uma subcategoria deve estar vinculada a uma ou várias palavras-chaves, vinculadas entre si ou não, conforme Figura 89.

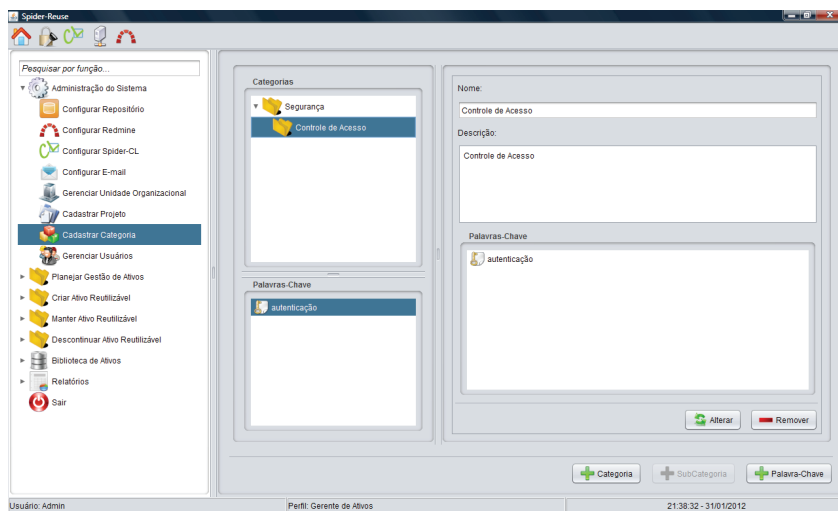


Figura 89. Tela de Cadastro de Categorias

No módulo de execução, a primeira macro-atividade a ser executada é o planejamento da gestão de reuso de ativos. A organização precisa iniciar o planejamento por meio da definição organizacional de Ativo Reutilizável. Esta definição é necessária para registrar a abrangência de um ativo organizacional segundo a visão da gestão da organização. Um ativo reutilizável pode ser entendido como módulos prontos ou como fragmentos de software que estejam preparados para serem utilizados em vários contextos.

Na tela de definição de ativo reutilizável o usuário também pode consultar informações sobre o repositório organizacional, assim como sobre os projetos que fazem parte do cenário de reuso de ativos de software, facilitando e tornando coerente a definição organizacional de ativo reutilizável.

A Figura 90 apresenta a tela de definição de ativo reutilizável, da ferramenta Spider-Reuse.

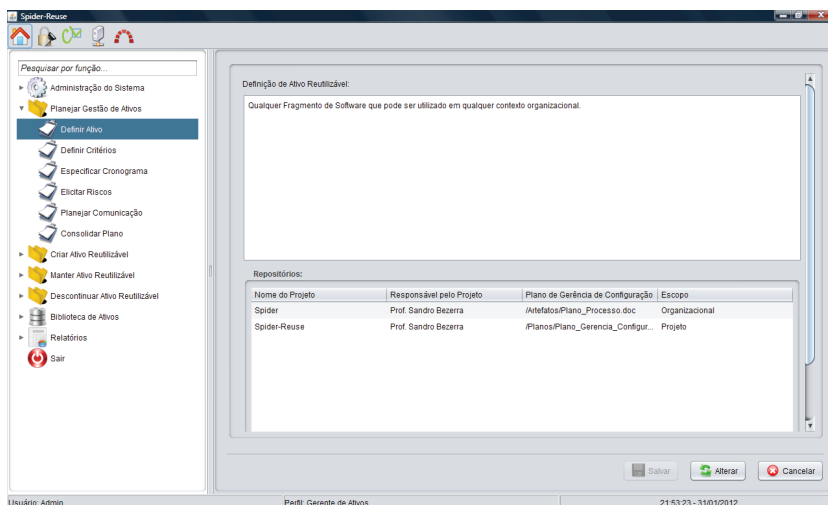


Figura 90. Tela de Definição de Ativo Reutilizável

A definição dos critérios de aceitação, certificação, qualidade, descontinuidade e classificação, faz uso da integração da Spider-Reuse com a ferramenta externa Spider-CL (vide Figura 91). Os critérios e os *checklists* de cada grupo de critérios são definidos na ferramenta Spider-CL e posteriormente carregados na ferramenta Spider-Reuse. O acesso a Spider-CL é feito por meio de um simples *click* de botão, permitindo ao usuário um acesso direto e transparente, devido a configuração de integração já efetuada.

Ao final do planejamento da gestão de reuso de ativos é gerado o plano de gestão de ativos, que contempla desde a definição de ativo organizacional, passando pela definição dos critérios, do cronograma, dos riscos, até a definição do plano de comunicação. Para a devida liberação do plano de gestão de ativos no repositório organizacional, são necessárias duas avaliações: a da Garantia de Qualidade e da Gerência de Configuração (vide Figura 92). Após a avaliação positiva do gerente de qualidade e do gerente de configuração, o gerente de ativos precisa liberar o plano de gestão de ativos

para disponibilizar o mesmo para consulta a todos os envolvidos no reuso de ativos de software.

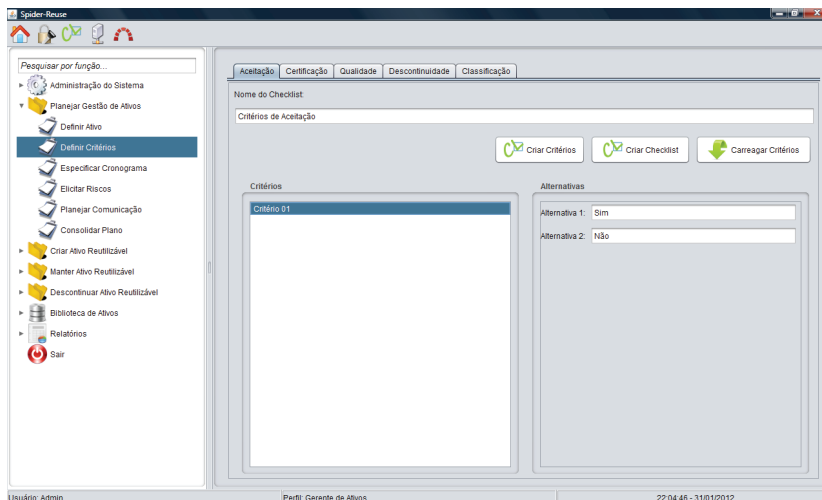


Figura 91. Tela de Definição dos Critérios

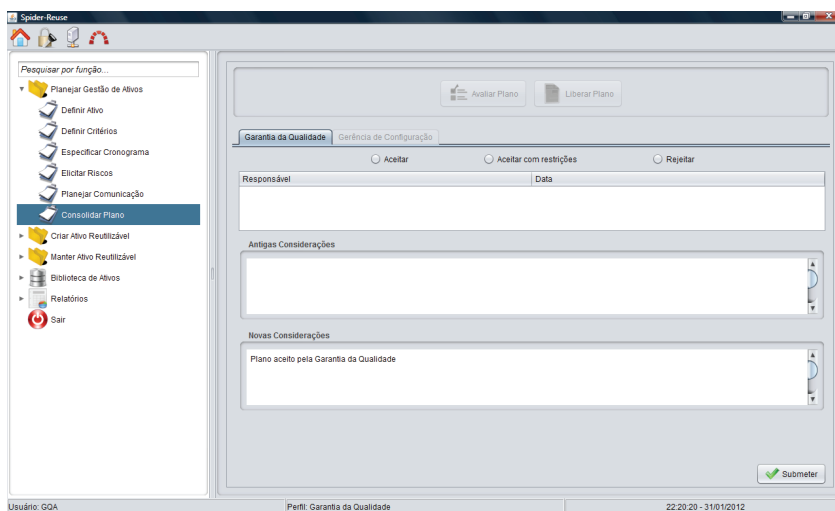


Figura 92. Tela de Consolidação do Plano de Gestão de Ativos

Ainda no módulo de execução, após a geração do plano de gestão de ativos, a criação de novos ativos reutilizáveis é disponibilizada aos usuários

que possuem permissão para esta macro-atividade. A criação de um novo ativo reutilizável tem início quando o gerente de domínio faz uma solicitação de avaliação de um novo ativo, submetendo ao gerente de ativos, conforme Figura 93. Esta solicitação pode ser feita para novos ativos, ativos rejeitados em solicitações anteriores ou para ativos descontinuados.

Figura 93. Tela de Solicitação de Avaliação de Novo Ativo

Quando submetida uma solicitação de avaliação de um novo ativo, o gerente de ativos precisa aplicar os *checklists* de aceitação, certificação e de qualidade, por meio da integração com a ferramenta externa Spider-CL. O gerente de ativos aplica os *checklists* na ferramenta Spider-CL, cujo acesso é feito pela ferramenta Spider-Reuse, e depois gera arquivos PDF que vão espelhar a aplicação dos critérios necessários ao ativo requerido na avaliação.

Na ferramenta Spider-Reuse o gerente de ativos carrega os arquivos PDF gerados, registrando a aplicação dos *checklists*, conforme Figura 94.

Depois de importados os *checklists*, o gerente de ativos precisa avaliar se o ativo organizacional será transformado em ativo reutilizável. Na tela de avaliação do novo ativo, o gerente de ativos precisa aceitar ou rejeitar o ativo submetido, conforme Figura 95.

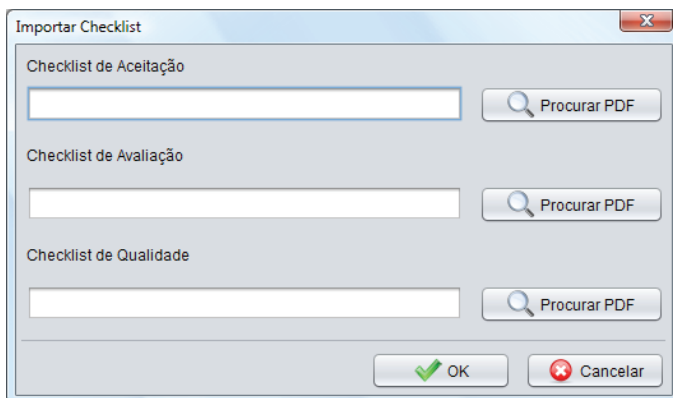


Figura 94. Tela de Importação dos *Checklists*

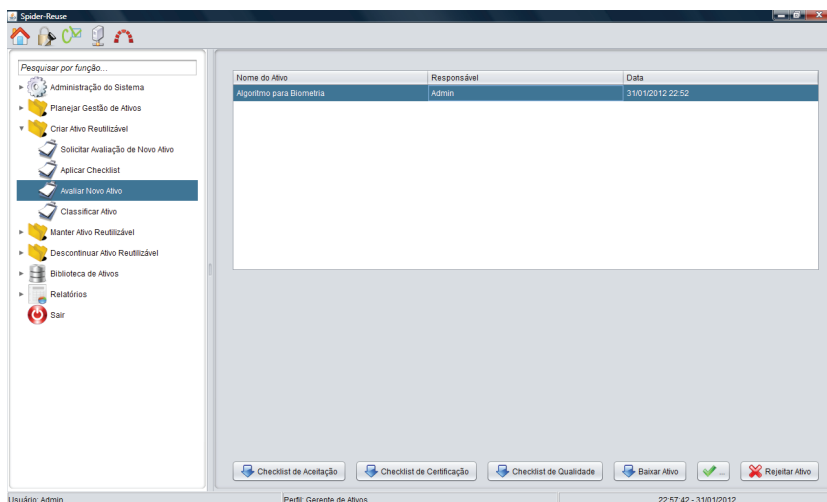


Figura 95. Tela de Avaliação do Novo Ativo

Os ativos rejeitados na avaliação ficam disponíveis para uma nova solicitação de avaliação, se necessário. Para os ativos aceitos como ativos reutilizáveis, estes devem ser classificados e disponibilizados na biblioteca de ativos reutilizáveis (vide Figura 96).

A ferramenta Spider-Reuse permite o controle das mudanças efetuadas ao longo da evolução de um ativo reutilizável. Este controle possibilita uma visão geral da manutenção dos ativos reutilizáveis, tendo início com a

solicitação de modificação de um ativo, submetida pelo engenheiro de domínio, para a aprovação do gerente de ativos, conforme Figura 97.

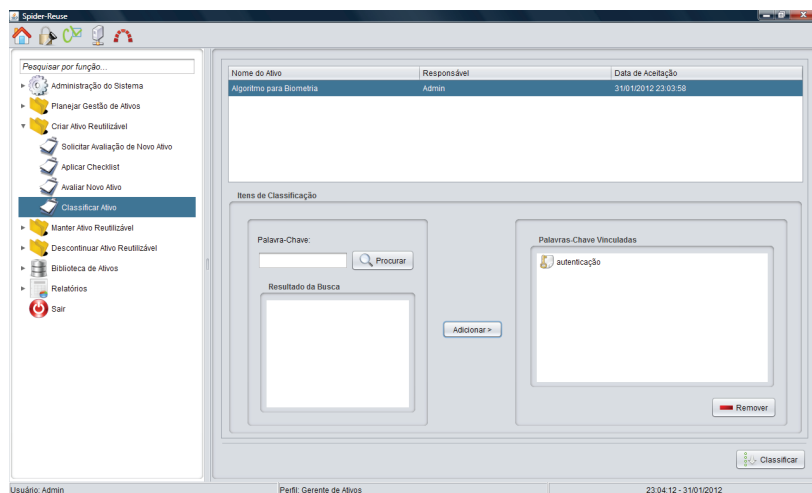


Figura 96. Tela de Classificação do Novo Ativo

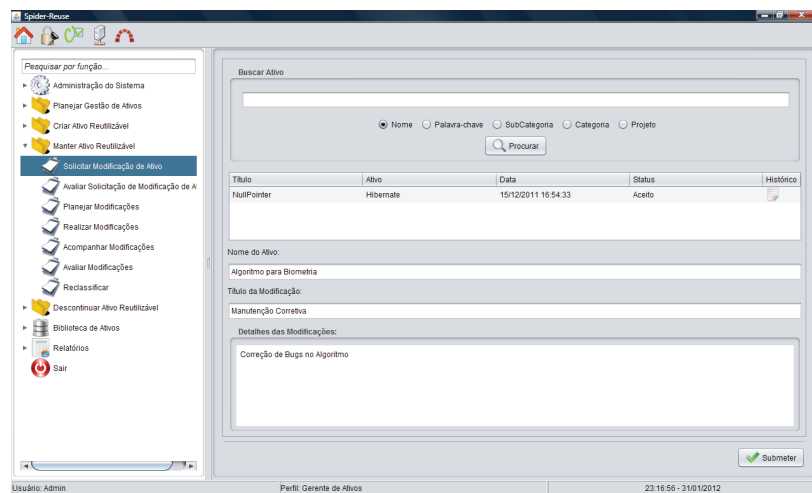


Figura 97. Tela de Solicitação de Manutenção de Ativo

O próximo passo é a avaliação da solicitação de modificação do ativo reutilizável submetida, cujo responsável é o gerente de ativos. Nesta avaliação

o gerente de ativos precisa aceitar ou rejeitar as modificações solicitadas, fornecendo um parecer para a decisão tomada, conforme Figura 98.

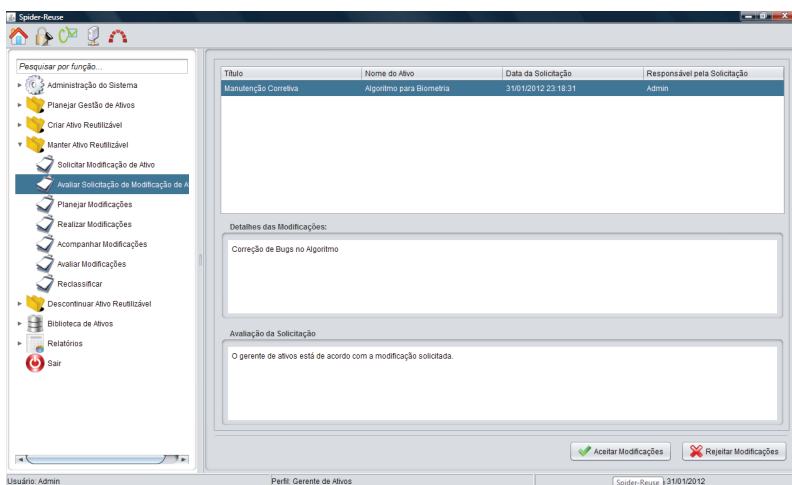


Figura 98. Tela de Avaliação da Solicitação de Manutenção de Ativo

Se a avaliação da solicitação de modificação do ativo for aceita, o gerente de ativos precisa planejar as modificações (vide Figura 99), apontando os participantes da execução das modificações, assim como a periodicidade de aplicação das mudanças.

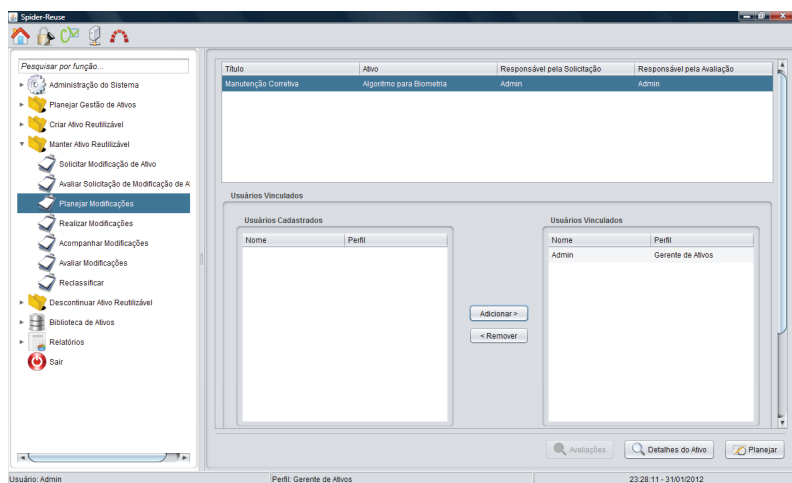


Figura 99. Tela de Planejamento das Modificações do Ativo

A atividade seguinte ao planejamento das modificações do ativo é a realização das modificações. Para esta atividade será trabalhada a integração com ferramenta externa Redmine, que permite a gestão de mudanças por meio de uma máquina de estados. O objetivo da realização das modificações é de criar um projeto na ferramenta Redmine, com o nome semelhante ao título da modificação fornecido na tela de solicitação de modificação do ativo, incluindo no projeto uma entrada (*issue*) com o *status* inicial da modificação. A Figura 100 apresenta a tela de realização das modificações.

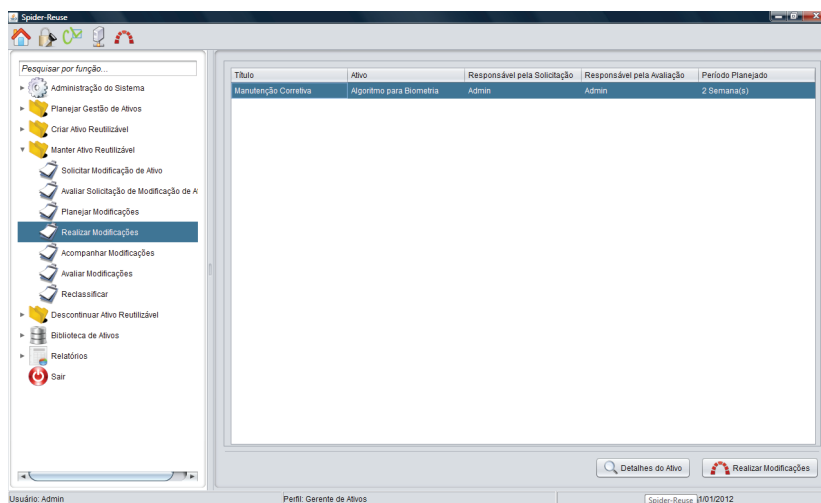
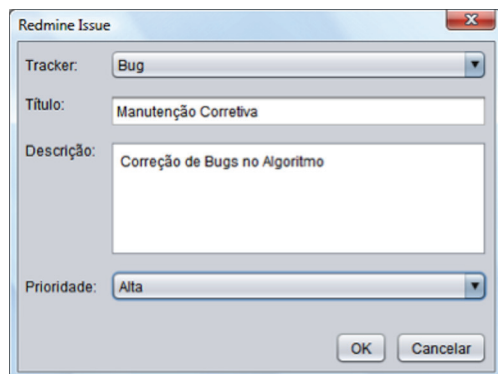


Figura 100. Tela de Realização das Modificações

A ferramenta Spider-Reuse antes de criar o projeto na ferramenta Redmine, solicita a definição do tipo da *tracker*, que classifica o tipo de atividade criada, e da prioridade para a modificação requerida (vide Figura 101).



Redmine Issue

Tracker: Bug

Título: Manutenção Corretiva

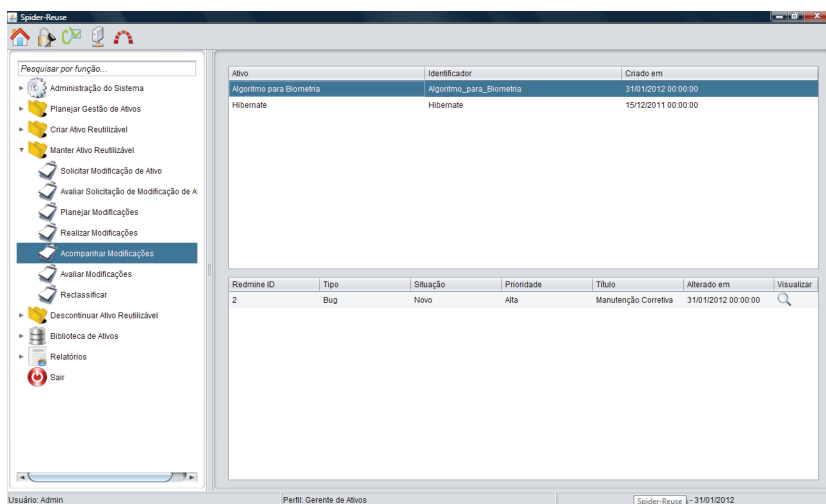
Descrição: Correção de Bugs no Algoritmo

Prioridade: Alta

OK Cancelar

Figura 101. Tela de Definição da Entrada no Redmine

Depois de criado o projeto na ferramenta externa Redmine, o acompanhamento das modificações realizadas pode ser feito pela ferramenta Spider-Reuse, conforme Figura 102. Para consultar informações mais detalhadas sobre as modificações, o usuário pode acessar a ferramenta Redmine, clicando na lupa ao lado direito da entrada registrada.




Spider-Reuse

Pesquisar por função...

- Administração do Sistema
- Planejar Gestão de Ativos
- Criar Ativo Reutilizável
- Manter Ativo Reutilizável
- Solicitar Modificação de Ativo
- Avaliar Solicitação de Modificação de Ativo
- Planejar Modificações
- Realizar Modificações
- Acompanhar Modificações
- Avaliar Modificações
- Reclassificar
- Descontinuar Ativo Reutilizável
- Biblioteca de Ativos
- Relatórios
- Sair

Ativo	Identificador	Criado em
Algoritmo para Biometria	Algoritmo_para_Biometria	31/01/2012 00:00:00
Hibernate	Hibernate	15/12/2011 00:00:00

Redmine ID	Tipo	Situação	Prioridade	Título	Alterado em	Visualizar
2	Bug	Novo	Alta	Manutenção Corretiva	31/01/2012 00:00:00	

Usuário: Admin Perfil: Gerente de Ativos Spider-Reuse - 31/01/2012

Figura 102. Tela de Acompanhamento das Modificações

Para finalizar o ciclo de mudanças de um ativo reutilizável, as modificações precisam ser avaliadas pelo gerente de ativos, quando o projeto criado

na ferramenta externa Redmine apresenta o status de conclusão. Se na avaliação das modificações do ativo, o parecer do gerente de ativos apresentar rejeição nas modificações, esta solicitação volta para a tela de planejamento de modificações, sendo necessário reiniciar o ciclo de modificações a partir deste ponto. Caso o parecer seja favorável e o gerente de ativos aceite as modificações, o ativo reutilizável modificado precisa ser reclassificado para estar novamente disponível na biblioteca de ativos.

A Figura 103 apresenta a tela de avaliação das modificações de um ativo reutilizável.

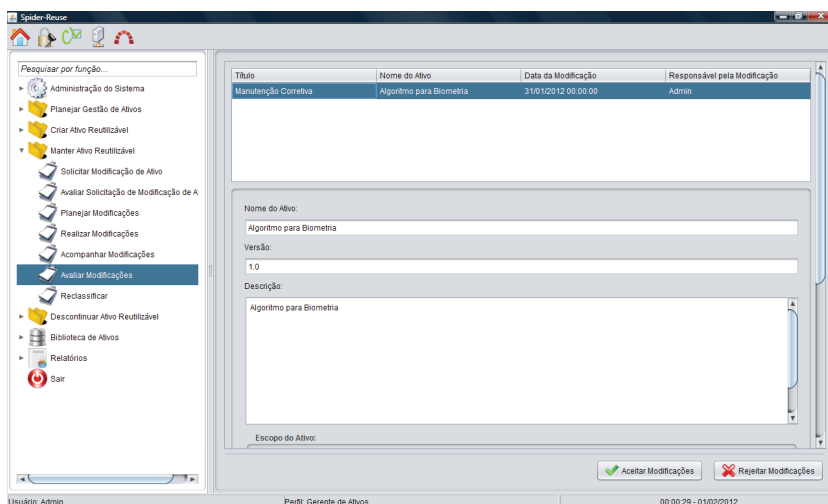


Figura 103. Tela de Avaliação das Modificações

A ferramenta Spider-Reuse permite que um ativo reutilizável seja descontinuado e retirado da biblioteca de ativos reutilizáveis. A descontinuidade de um ativo reutilizável é responsabilidade do gerente de ativos e deve ser iniciada com a aplicação dos *checklists* de descontinuidade definidos na ferramenta externa Spider-CL, conforme Figura 104.

De forma semelhante em funcionalidades anteriores, o gerente de ativos aplica os *checklists* na ferramenta Spider-CL, cujo acesso é feito pela ferramenta Spider-Reuse, e depois gera arquivos PDF que vão espelhar a

aplicação dos critérios de descontinuidade necessários ao ativo reutilizável analisado.

Na ferramenta Spider-Reuse o gerente de ativos carrega os arquivos PDF gerados, registrando a aplicação dos *checklists* de descontinuidade.

Após aplicar os *checklists* de descontinuidade, o gerente de ativos precisa avaliar o ativo reutilizável fornecendo um parecer quanto a manutenção ou descontinuidade do ativo na biblioteca de ativos reutilizáveis (vide Figura 105).

A última macro-atividade do módulo de execução da ferramenta abrange o desenvolvimento com reutilização. A ferramenta Spider-Reuse permite a busca e o reuso de todos os ativos reutilizáveis disponíveis na biblioteca de ativos da organização, conforme Figura 106.

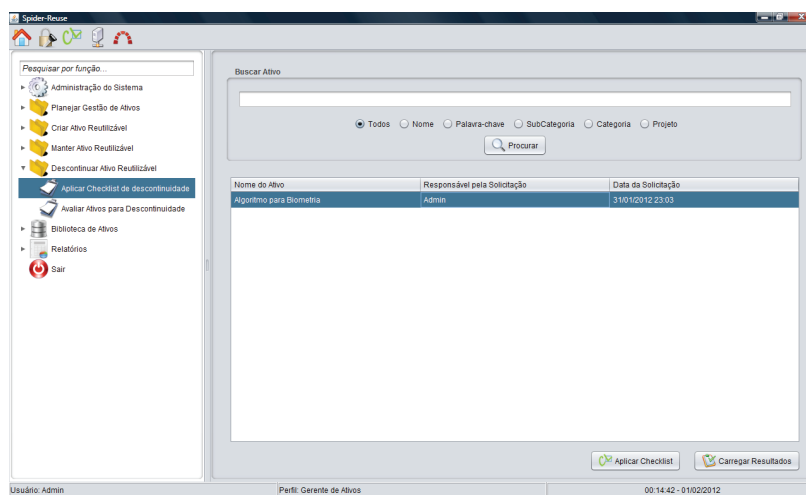


Figura 104. Tela de Aplicar *Checklist* de Descontinuidade

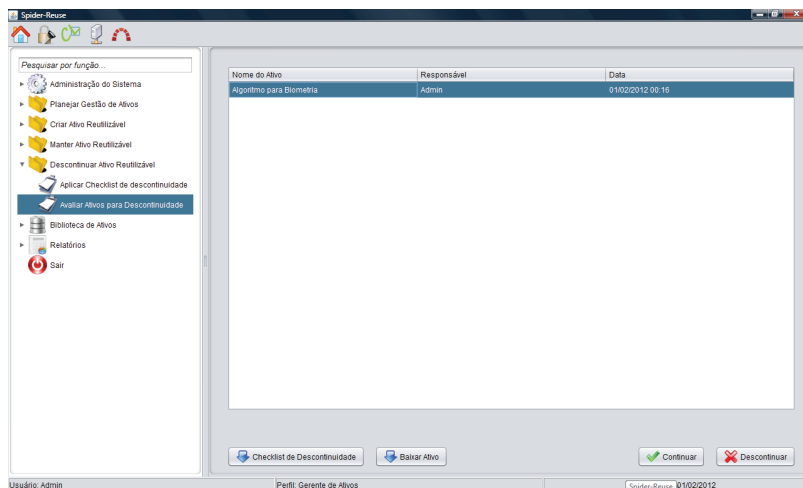


Figura 105. Tela de Avaliar Ativos para Descontinuidade

Para todo reuso de ativos reutilizáveis se faz necessário o registro de um *feedback*, fornecido pelo consumidor de ativos responsável pelo reuso. O *feedback* é obrigatório, por ser o insumo utilizado pelo gerente de ativos na aplicação de melhorias no ativo reutilizável, e a ferramenta Spider-Reuse não permite o reuso de ativos com *feedbacks* em aberto, pelo mesmo consumidor de ativos.

A Figura 107 apresenta a tela da ferramenta Spider-Reuse que possibilita o registro de um *feedback* para um ativo reutilizado.

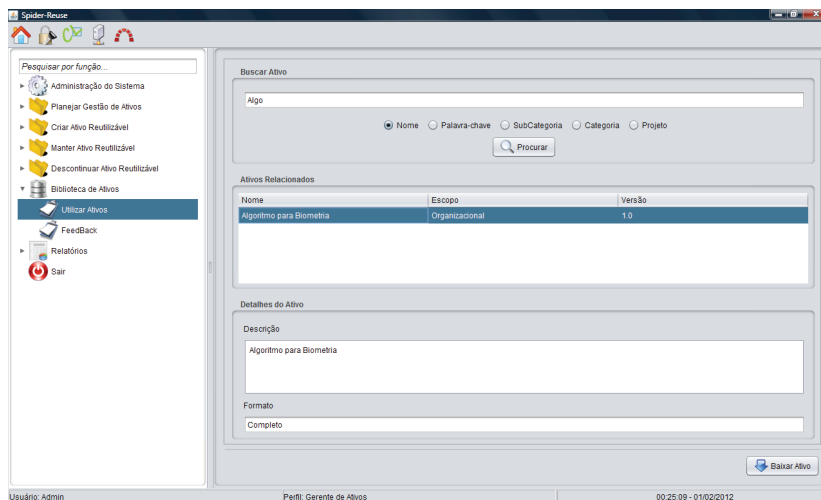


Figura 106. Tela de Utilização de Ativos Reutilizáveis

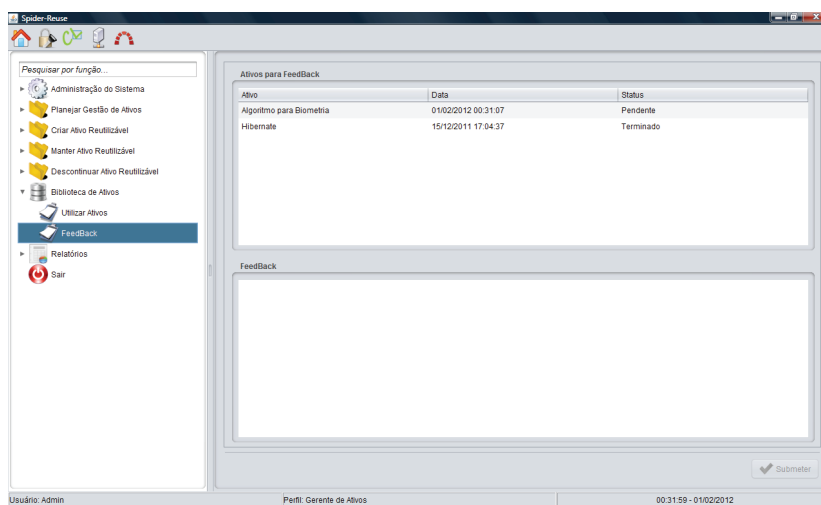


Figura 107. Tela de *Feedback*



Capítulo 14

Implementação do Processo de Desenvolvimento de Requisitos

em [Yoshidome *et al.*, 2012]

Ewelton Yoshio Chiba Yoshidome

Maurício Ronny de Almeida Souza

Wallace Michel Pinto Lira

Sandro Ronaldo Bezerra Oliveira

Alexandre Marcos Lins de Vasconcelos

A Engenharia de Requisitos tem a função de auxiliar o entendimento do sistema a ser desenvolvido, incluindo tarefas para analisar as necessidades do cliente e a forma de como o sistema irá interagir com os usuários finais [Pressman, 2006]. A Engenharia de Requisitos pode ser dividida em duas atividades [SOFTEX, 2012a; SEI, 2010]: Desenvolvimento de Requisitos e Gerência de Requisitos.

Segundo SOFTEX (2012a) e SEI (2010), o objetivo do processo de Desenvolvimento de Requisitos é interpretar as necessidades do cliente para a criação dos requisitos do sistema a ser desenvolvido. No CMMI-DEV, a área de processo *Requirements Development* (RD) está estruturada em dez práticas específicas (SP – *Specific Practices*, "é a descrição de uma atividade considerada importante para alcançar uma meta específica associada" [SEI, 2010]), que são:

- SP1.1: Elicit Needs;
- SP1.2: Transform Stakeholder Needs into Customer Requirements;
- SP2.1: Establish Product and Product Component Requirements;
- SP2.2: Allocate Product Component Requirements;
- SP2.3: Identify Interface Requirements;
- SP3.1: Establish Operational Concepts and Scenarios;
- SP3.2: Establish a Definition of Required Functionality and Quality Attributes;
- SP3.3: Analyze Requirements;
- SP3.4: Analyze Requirements to Achieve Balance;
- SP3.5: Validate Requirements.

No contexto do MR-MPS-SW, o processo de Desenvolvimento de Requisitos (DRE) está estruturado em oito resultados esperados aderentes às práticas específicas do CMMI-DEV [SOFTEX, 2012a]. Um resultado esperado "é um resultado observável do sucesso do alcance do propósito do processo" [ISO/IEC, 1995]. Em seguida, são listados os resultados esperados do processo de Desenvolvimento de Requisitos:

- DRE1: As necessidades, expectativas e restrições do cliente, tanto do produto quanto de suas interfaces são identificadas;
- DRE2: Um conjunto definido de requisitos do cliente é especificado a partir das necessidades, expectativas e restrições identificadas;
- DRE3: Um conjunto definido de requisitos funcionais e não-funcionais, do produto e dos componentes do produto que descrevem a solução do problema a ser resolvido é definido e mantido a partir dos requisitos do cliente;
- DRE4: Os requisitos funcionais e não-funcionais de cada componente do produto são refinados, elaborados e alocados;

- DRE5: Interfaces internas e externas do produto e de cada componente do produto são definidas;
- DRE6: Conceitos operacionais e cenários são desenvolvidos;
- DRE7: Os requisitos são analisados, usando critérios definidos, para balancear as necessidades dos interessados com as restrições existentes;
- DRE8: Os requisitos são validados.

Segundo SOFTEX (2012a) e SEI (2010), Gerência de Requisito é o processo responsável por gerenciar todos os requisitos do projeto e identificar as suas inconsistências com os demais produtos de trabalho. Entretanto, o processo de Gerência de Requisitos está fora do escopo desta pesquisa, focando-se apenas no processo de Desenvolvimento de Requisitos. Existe um trabalho publicado no Workshop de Engenharia de Requisitos de 2010, apresentado pelo mesmo grupo do projeto SPIDER, que descreve uma metodologia com apoio ferramental para atendimento às recomendações do processo de Gerência de Requisitos [Cardias Junior *et al.*, 2010].

14.1. Trabalhos Relacionados

Este trabalho foca-se em propor uma metodologia para apoiar o Processo de Desenvolvimento de Requisitos e atua como uma extensão à metodologia proposta em [Cardias Junior *et al.*, 2010], no qual seu trabalho descreve uma metodologia de uso de ferramentas livres para o apoio do Processo de Gerência de Requisitos do MPS.BR. Pretende-se, a partir desta extensão, utilizar um *kit* de ferramentas em comum que sejam capazes de contemplar os processos de gerência e desenvolvimento de requisitos.

Em [Zaniro e Fabbri, 2008] é descrito um processo de levantamento e modelagem de requisitos para sistemas web com o uso do método Web-SEMP (*Web System Elicitation, Modeling and Planning*). Porém, pelo fato de

seu trabalho não ser direcionado à adequação a modelos de qualidade de processo de software (CMMI-DEV, MR-MPS-SW), nota-se a ausência de algumas práticas exigidas pelo modelos, tais como o uso de critérios objetivos para verificação e validação de requisitos e a definição de interfaces internas e externas.

Martins *et al.* (2006) descrevem uma ferramenta de apoio à Engenharia de Requisitos integrada ao ambiente ODE (*Ontology-based Development Enviroment*). Porém, este trabalho foca-se em apresentar funcionalidades voltadas à gerência e desenvolvimento de requisitos, não apresentando um alinhamento aos modelos de qualidade existentes no mercado. Esse fato pode ser observado pela ausência da funcionalidade/serviço de criação e aplicação de critérios objetivos para realizar o balanceamento entre as necessidades dos interessados às restrições existentes no projeto, e os critérios para executar a validação dos requisitos, junto ao cliente ou usuário final.

Adicionalmente, encontra-se na literatura especializada inúmeros trabalhos que tratam de ferramentas de apoio à gestão e engenharia de requisitos, como [Silva e Martins, 2008] e [Zanlorenci e Burnett, 2000]. Entretanto, percebe-se dois aspectos negativos em relação aos trabalhos propostos: primeiro, cita-se o fato destes trabalhos referenciarem ferramentas que não estão disponíveis para uso/operação, o que não apoia a implementação de melhoria efetivamente; segundo, estes trabalhos não apresentam metodologias para apoiar a operação das ferramentas quando da implementação das boas práticas constantes nos modelos de qualidade, o que pode acarretar no seu desuso quando considerado o contexto dos programas de melhoria dos processos organizacionais.

14.2. Ferramentas de Apoio

Para a implementação da metodologia do processo de Desenvolvimento de Requisitos, foram utilizadas as ferramentas Openproj (apoio à Gerência

de Projetos), OSRMT (apoio à Gerência de Requisitos), Redmine (apoio ao Controle de Mudanças), Astah Community (apoio à modelagem UML - *Unified Modeling Language*) e Spider-CL (apoio a Geração de *Checklists*). Vale salientar que o uso destas ferramentas só fará sentido se o fluxo da metodologia for seguido. As subseções a seguir descrevem com mais detalhes cada uma das ferramentas.

14.2.1. OSRMT

O OSRMT (*Open Source Requirements Management Tool*) é uma ferramenta, desenvolvida na linguagem Java, projetada para apoiar o processo de Gerência de Requisitos. Licenciada sob os termos da GPL, encontra-se disponível em <http://sourceforge.net/projects/osrmt>.

A ferramenta possui suporte ao processo de Desenvolvimento de Requisitos do MR-MPS-SW no que se refere ao armazenamento e registro dos requisitos gerados durante a fase de elicitação.

14.2.2. Openproj

O Openproj, disponível em <http://http://openproj.org/openproj>, é uma ferramenta *desktop*, livre e *open source*, voltada para apoiar a Gerência de Projetos. Esta ferramenta é composta por um grande número de funcionalidades que dão suporte ao cronograma, gestão de recursos humanos e riscos. Outra característica presente na ferramenta é a possibilidade de gerar representações gráficas como WBS (*Work Breakdown Structure*), RBS (*Resource Breakdown Structure*), CPM (*Critical Path Method*) e Gantt Chart.

A versão utilizada na metodologia foi modificada pelo projeto SPIDER com a adição de funcionalidades de suporte a estimativas, riscos e definição do escopo do projeto. O objetivo das alterações foi permitir que a ferramenta

possua uma maior aderência ao MR-MPS-SW. A versão customizada encontra-se disponível em <http://www.spider.ufpa.br/>.

O propósito do uso da ferramenta Openproj na metodologia é realizar o registro e armazenamento do método de elicitação de requisitos.

14.2.3. Redmine

Redmine, disponível em <http://www.redmine.org>, é uma ferramenta web para *bugtracking*, desenvolvida em Ruby, com o objetivo de gerenciar mudanças nos produtos de trabalho de um projeto. Além de realizar a gestão de mudanças, a ferramenta Redmine proporciona suporte à Gerência de Projetos.

Para o contexto da metodologia, a ferramenta de controle de mudanças é utilizada para o controle do ciclo de vida das tarefas de verificação, validação e implementação dos requisitos.

14.2.4. Spider-CL

A Spider-CL, disponível em www.spider.ufpa.br, é uma ferramenta desenvolvida no projeto SPIDER, com o propósito de criar *checklists* compostos por critérios objetivos para utilização em diversos contextos, provendo mecanismos para a aplicação destes *checklists*, mantendo histórico e registrando seus resultados. *Checklist* é um conjunto de atributos que servem para avaliar determinado produto de trabalho. Cada atributo possui uma lista de possíveis alternativas das quais apenas uma pode ser escolhida.

No contexto da metodologia proposta, os *checklists* gerados pela ferramenta Spider-CL são utilizados para a definição de critérios objetivos necessários para a implementação de alguns resultados esperados.

14.2.5. Astah Community

Astah Community, disponível em <http://astah.net>, é uma ferramenta gratuita, mas não *open source*, voltada para a modelagem de diagramas UML – *Unified Modeling Language*. Além do Astah Community, existem outras três versões: Astah UML, Astah Professional e Astah Share, que disponibilizam outras funcionalidades além da modelagem UML, porém, sua licença é comercial.

Na metodologia, a ferramenta Astah Community é utilizada para o desenvolvimento dos diagramas necessários para representar o projeto dos requisitos.

14.3. A Metodologia de Implementação do Processo de DRE

Esta seção descreve a metodologia para implementar o processo de Desenvolvimento de Requisitos com o uso das ferramentas propostas. Esta proposta pretende estar aderente aos modelos constantes nos programas MPS.BR e CMMI.

Salienta-se que para a implementação de um programa de melhoria organizacional, são requeridas ações que tratem de maturidade e capacidade. Para as ações referentes à maturidade, pode-se citar a definição de um processo, institucionalização de uma política, entre outros. Para realizar a evidência de ações que tratem da capacidade, pode-se verificar a aplicação prática do processo definido na organização, treinamento/capacitação nos ativos constantes no processo, entre outros. Uma dessas boas práticas recomenda o uso de ferramentas de software para sistematizar a execução de atividades constantes no processo.

Não é foco deste trabalho definir um processo para a implementação dos resultados esperados/práticas específicas do processo de Desenvolvimento de Requisitos, mas sim propor um conjunto de ferramental de apoio para auxiliar na implementação das boas práticas constantes nos modelos. Assim,

qualquer análise de aderência descrita na metodologia de uso proposta neste trabalho recai na análise do atendimento dos serviços descritos nas ferramentas discutidas de acordo com as características recomendadas pelo processo de Desenvolvimento de Requisitos.

Na Figura 108 pode ser visualizado o fluxo das atividades da metodologia (definido usando o padrão BPMN), junto com o nome das ferramentas as quais serão utilizadas para executar cada atividade constante na metodologia. Nas subseções seguintes serão descritas a metodologia proposta. Para facilitar o entendimento da metodologia, as atividades referentes ao processo de Desenvolvimento de Requisitos foram divididas em três, a saber: Desenvolver Requisitos do Cliente; Desenvolver Requisitos do Produto; e Verificar e Validar os Requisitos.

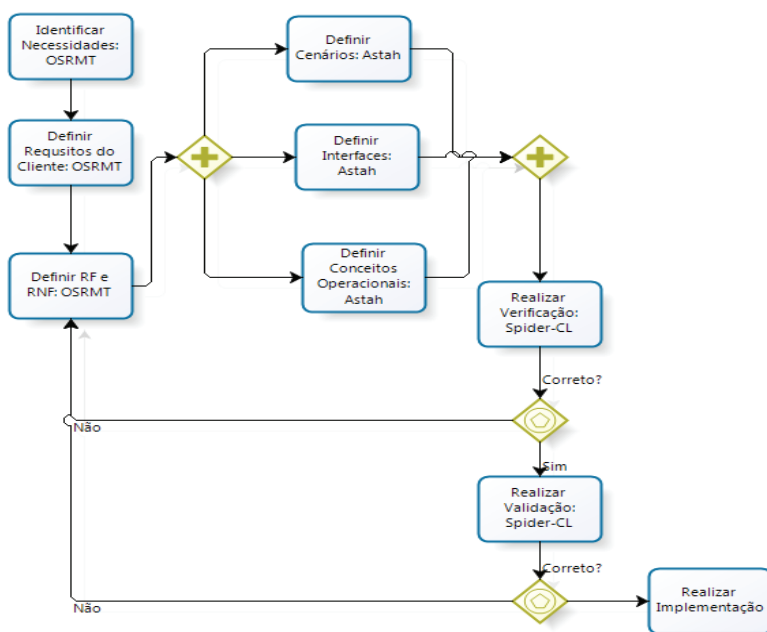


Figura 108. Fluxo das atividades da metodologia

14.3.1. Desenvolver Requisitos do Cliente

O alcance do resultado esperado DRE1 do MR-MPS-SW, compatível com a prática específica SP1.1 da área de processo RD do CMMI-DEV, envolve a utilização de métodos adequados para identificar necessidades, expectativas, restrições e interfaces do cliente [SOFTEX, 2012a; SEI, 2010]. A coleta desses requisitos pode ser feita a partir de métodos de elicitação de requisitos como entrevistas, questionários, *brainstorms*, uso de protótipos, casos de uso, entre outras. Uma boa prática que pode ser adotada é o registro e a descrição da forma de elicitação de requisitos utilizada na empresa, para que estas informações possam ser acessadas por todos da organização.

A descrição da forma de elicitação de requisitos é feita utilizando o campo de descrição do projeto na ferramenta Openproj, como mostra a Figura 109(a). Com o objetivo de separar a descrição do projeto e a forma de coleta de requisitos, pode ser definido um separador, nomeando-se um título para cada um desses itens. Para finalizar a implementação do DRE1/SP1.1, a funcionalidade de *Features* na ferramenta OSRMT é utilizada, onde são cadastradas três *features*: uma representando as necessidades; outra para expectativas; e a última para as restrições do cliente.

Quando as necessidades, expectativas e restrições estão devidamente registradas e detalhadas, é o momento de utilizá-las como insumo para a geração dos requisitos do cliente, para isso, a funcionalidade de *Requirement* da ferramenta OSRMT é utilizada. Na metodologia, define-se a criação de um item denominado "Requisitos do Cliente" com o uso da funcionalidade *requirement*, o qual servirá de recipiente para todos os requisitos derivados a partir das necessidades, expectativas e restrições do cliente. Para cada requisito, um *requirement* referente será criado o qual estará ligado hierarquicamente a "Requisitos do Cliente".

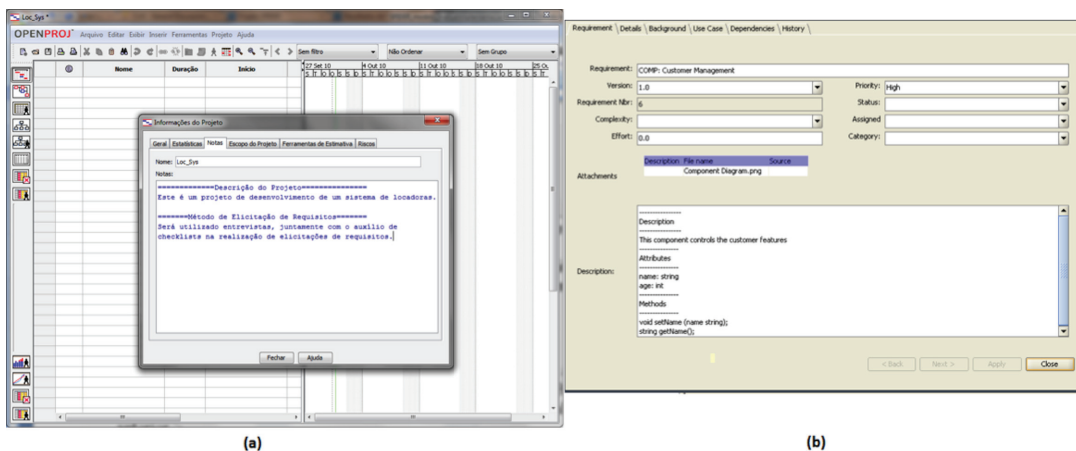


Figura 109 (a). Descrição do método de elicitação de requisitos na ferramenta Openproj; (b). Tela de detalhamento do componente com a descrição de suas interfaces.

Cada requisito do cliente gerado deverá ser mapeado às necessidades, expectativas ou restrições que o originou, além de sua descrição, prioridade, complexidade, estado, versão. Caso existam dependências múltiplas (proveniente de um conjunto de necessidades/expectativas/restrições), estas dependências também devem ser evidenciadas. Após esse procedimento, o resultado esperado DRE2 e a prática específica SP1.2 estarão contemplados.

14.3.2. Desenvolver Requisitos do Produto

Para alcançar o DRE3 e o SP2.1, os requisitos funcionais e não-funcionais são derivados a partir dos requisitos do cliente, que são devidamente registrados, detalhados e mapeados pela ferramenta OSRMT. Salienta-se que a prática específica SP2.1 trata do estabelecimento e refinamento dos requisitos funcionais e não-funcionais.

O registro dos requisitos funcionais e não-funcionais é feito de forma similar ao registro dos requisitos do cliente, o qual deverá ser criado um

item denominado de "Requisitos Funcionais e Não-Funcionais", onde estarão aninhados os requisitos funcionais e não-funcionais. A utilização de critérios para o agrupamento destes requisitos é uma boa prática sugerida pela SOTEX (2012a).

Na metodologia, o critério de agrupamento dos requisitos é feito com o uso do conceito de componente. Componente é um bloco construtivo modular para software de computador [Pressman, 2006]. Para a representação de um componente, é criado um item (*requirement*) com o nome no formato "**COMP: NOME_DO_COMPONENTE**" o qual é registrado dentro do item "Requisitos Funcionais e Não-Funcionais". Em cada componente são cadastrados os requisitos funcionais e não-funcionais do sistema. Com o objetivo de preservar a legibilidade é adotado o uso das *tags* (rótulos) "RF:" e "RNF:", antes do nome de cada requisito, os quais significam requisitos funcionais e requisitos não-funcionais, respectivamente.

Após o cadastro de um requisito funcional ou não-funcional, é imprescindível o seu detalhamento, por isso deve-se descrevê-lo informando sua versão, o nível de sua complexidade e urgência. Além de caracterizar o requisito, é necessário fazer o mapeamento ao componente em que este pertence. O propósito deste mapeamento é evidenciar que o conjunto de requisitos faz parte de um determinado componente, quando uma atividade de rastreabilidade for realizada.

Quando todos os requisitos funcionais e não-funcionais forem detalhados e mapeados ao seu componente, este componente deve, também, ser detalhado. Primeiramente, deve ser feita uma descrição do componente com informações de sua interface (métodos utilizados e dados de entrada e saída), além de descrever suas funcionalidades. Com o objetivo de organizar as informações presentes no campo de descrição, são criados tópicos para separar a descrição, os métodos e os dados de entrada e saída do componente, como visto na Figura 109(b). A utilização de tópicos não precisa se restringir apenas na separação desses conteúdos, para cada novo conteúdo, um novo tópico pode ser criado.

Depois do registro e detalhamento do componente, faz-se necessário refiná-lo, ou seja, devem ser utilizadas técnicas de modelagem para o desenvolvimento de diagramas com o objetivo de proporcionar uma análise do projeto com diferentes perspectivas. A ferramenta responsável pela modelagem, utilizada na metodologia, é a Astah Community, a qual permite a elaboração de diagramas no modelo UML.

No momento em que os diagramas são elaborados, é necessário disponibilizá-los e mapeá-los aos requisitos que os geraram. Para isso, duas abordagens são adotadas na metodologia: (1) os diagramas produzidos são anexados aos componentes ou requisitos que os geraram; (2) no campo "*Design*" é criado designs que representam os diagramas. Na ferramenta OSRMT, design é o componente que representa e descreve os dados arquiteturais do sistema. Para cada *design*, é anexado apenas um tipo de diagrama e seu nome é associado a este tipo, ou seja, se o diagrama anexado ao *design* for um diagrama de classes, este item será rotulado como "Diagrama de Classes".

A abordagem (1) é utilizada nos casos em que os diagramas gerados representam uma parte específica do projeto, como um componente ou requisito. Em casos em que o diagrama possui um escopo mais abrangente, como a representação da comunicação entre componentes, é feito uso da abordagem (2).

Cada diagrama desenvolvido permite uma análise em uma determinada perspectiva, porém, para o MR-MPS-SW e o CMMI-DEV, devem ser gerados diagramas que modelem cenários, conceitos operacionais e interfaces internas e externas, obviamente não se restringindo apenas a essas visões.

Na metodologia, são modelados casos de uso para evidenciar que os cenários foram desenvolvidos. Cada caso de uso produzido é anexado ao componente ou ao requisito que o gerou. Depois de realizado o anexo, devem ser descritas suas pré e pós-condições, juntamente com seus fluxos principal e alternativo no guia "Use Case", na tela de detalhamento do requisito.

Os conceitos operacionais podem ser implementados com o uso de diagramas de implantação, navegabilidade ou atividades. Esta metodologia faz

uso do diagrama de atividades. Cada organização está livre para escolher a melhor forma de representar os conceitos operacionais do projeto, não estando restritos a apenas um tipo de diagrama.

Primeiramente, deve-se criar um item, utilizando a funcionalidade design, com o nome de "Diagrama de Atividades" para realizar o anexo dos diagramas de atividades. Outra necessidade é o detalhamento de cada atividade que compõe o diagrama, além de seu mapeamento aos requisitos que o geraram. Após a definição dos cenários e conceitos operacionais, o resultado esperado DRE6, juntamente com a prática específica SP3.1, estão totalmente satisfeitos.

Com o propósito de alcançar o DRE5 e o SP2.3, faz-se necessário elaborar as interfaces internas e externas do sistema. Para contemplar este resultado esperado, na metodologia são modelados os diagramas de componente contendo suas interfaces de comunicação. Este diagrama deve ser anexado a um *design* denominado "Diagrama de Componentes", além de realizar seu detalhamento. Também é importante que o diagrama de componentes esteja mapeado ao componente ou requisitos funcionais e não-funcionais que serviram de insumo para seu desenvolvimento.

A evidência da implementação do DRE5 e SP2.3 é feita quando for presenciado o diagrama de componentes, juntamente com suas interfaces e descrição. Outra evidência é a presença da lista de métodos e dados de entrada e saída no campo de descrição dos componentes.

Após definir os cenários, os conceitos operacionais e as interfaces, o projeto possuirá insumos suficientes para evidenciar um atributo denominado de Análise Funcional, permitindo contemplar a prática específica SP3.2. O resultado esperado DRE4 está parcialmente contemplado devido ao fato das alocações das tarefas ainda não terem sido realizadas. Análise Funcional é o exame de uma determinada função para a identificação de todas as subfunções necessárias para a sua execução, incluindo a identificação de seus relacionamentos e interfaces [SEI, 2010].

14.3.3. Verificar e Validar os Requisitos

Quando todos os diagramas necessários são modelados, descritos e mapeados, deve-se alocar algum membro responsável para realizar a verificação, validação e implementação dos componentes, junto aos seus diagramas. A alocação é feita atribuindo o nome do responsável no campo “Assigned”, na tela de detalhamento do componente na ferramenta OSRMT.

Além de designar um responsável na ferramenta OSRMT, uma *issue* deve ser instanciada na ferramenta Redmine com o objetivo de informar a alocação da tarefa à pessoa responsável, além de possibilitar o acompanhamento da evolução da *issue* alocada. Uma *issue* é a representação de uma solicitação de mudança, tarefa, defeito ou problema que tem sua evolução acompanhada desde sua criação até seu desfecho.

Nesta metodologia, para cada tarefa de verificação, validação e implementação, é instanciada uma *issue* do tipo tarefa com o nome seguindo a forma de nomenclatura:

- Verificação do componente "Nome_do_componente": Para tarefa de verificação;
- Validação do componente "Nome_do_componente": Para tarefas de validação;
- Implementação do componente "Nome_do_componente": Para tarefas de implementação.

Um componente, antes de ser implementado, deve passar pelo processo de verificação e, em seguida, validação. Com o objetivo de auxiliar no processo de verificação e validação, é utilizado o conceito de *checklist*. Na metodologia, a notificação e acompanhamento das tarefas de verificação, validação e implementação são feitas a partir da ferramenta Redmine, enquanto que os critérios objetivos são cadastrados e aplicados pela ferramenta Spider-CL.

O primeiro processo que deve ser executado é a verificação, com o objetivo de detectar erros, inconsistências, ambiguidades, omissões no documento de requisitos. Na metodologia, é criada uma tarefa solicitando que um componente seja verificado, juntamente com a disponibilização do *checklist* a ser aplicado. Caso seja detectado algum defeito, o estado da tarefa é modificado para "*Feedback*" e ocorre a alocação de um responsável para realizar as devidas correções, além de proceder o anexo do *checklist* aplicado (em formato PDF) à tarefa. Por fim, o componente verificado, retorna para a fase de análise de requisitos.

Caso não tenha sido detectado nenhum defeito no componente, o estado da tarefa é modificado para "*Resolved*" e o *checklist* aplicado é anexado. Além disso, uma tarefa de validação para o componente é instanciada no Redmine.

O processo de validação tem o objetivo confirmar se o produto ou componente do produto atende o seu uso pretendido quando colocado no ambiente para qual foi desenvolvido [SOFTEX, 2012a]. Diferente da verificação, a validação é executada pelo cliente externo, por esse motivo, deve ser disponibilizado um usuário de acesso na ferramenta Spider-CL com o perfil de cliente externo.

Quando o componente receber a aprovação do cliente, a tarefa deve ser modificada para o estado "*Resolved*" e o *checklist* aplicado deve ser anexado. Enfim, uma tarefa solicitando a implementação do componente deve ser instanciada, juntamente com a alocação de um responsável para resolvê-la.

Caso o componente não seja validado, o estado da tarefa é modificado para "*Feedback*" e o *checklist* aplicado é anexado. O componente reprovado na validação é enviado para a fase de análise, visando corrigir os problemas detectados. Como consequência, o componente passa pela fase de verificação novamente, com o objetivo de detectar possíveis novos defeitos provenientes das modificações realizadas.

No momento em que as tarefas de refinamento, detalhamento e alocação dos requisitos funcionais e não-funcionais são feitas, o resultado esperado DRE4 e a prática específica SP2.2 são alcançados completamente. Deve-se

salientar que a prática específica SP2.2 trata apenas de alocação de responsáveis para as tarefas de verificação, validação e implementação. O estabelecimento e refinamento de requisitos funcionais e não-funcionais é de responsabilidade do SP2.1.

Quando os procedimentos para gerenciar os ciclos de vida das tarefas de verificação e validação forem executados, os resultados esperados DRE7 e DRE8 serão contemplados, respectivamente. No caso do CMMI-DEV, as práticas específicas SP3.3 e SP3.4 são satisfeitas no momento que forem gerenciadas as tarefas de verificação, e a prática específica SP3.5 será contemplado quando as atividades de validação forem gerenciadas.

14.4. Análise de Aderência

A análise de aderência da metodologia proposta na Seção 14.3 é evidenciada a partir dos mapeamentos das funcionalidades e serviços presentes no conjunto de ferramentas com as práticas sugeridas pelos resultados esperados e práticas específicas do Processo de Desenvolvimento de Requisitos. A análise de aderência é sintetizada na Tabela 17 o qual mostra a relação entre os resultados esperados (RE), as práticas específicas (SP) e as ferramentas propostas, juntamente com uma breve descrição das funcionalidades e práticas que implementam cada resultado esperado. A aderência dos resultados esperados do MR-MPS-SW às práticas específicas do CMMI-DEV está baseada nos mapeamentos publicados em [SOFTEX, 2012b].

Tabela 17. Aderência entre as funcionalidades das ferramentas, as práticas específicas e os resultados esperados do processo DRE

RE	SP	Ferramentas	Funcionalidades/Práticas
DRE1	SP1.1	Openproj/ OSRMT	Definição do método de elicitação de requisitos no campo de notas; Registro e descrição das necessidades, expectativas e restrições no campo <i>Features</i> .

RE	SP	Ferramentas	Funcionalidades/Práticas
DRE2	SP1.2	OSRMT	Registro e descrição dos requisitos do cliente no campo <i>Requirements</i> .
DRE3	SP2.1	OSRMT	Registro e descrição dos requisitos funcionais e não-funcionais no campo <i>Requirements</i> .
DRE4	SP2.2 SP3.2	OSRMT/ Redmine/ Astah Community	Anexo de diagramas aos componentes e requisitos; Descrição dos diagramas nos componentes e requisitos; Alocação de um responsável na tela de detalhamento do requisito ou componente; Instanciação e acompanhamento de uma <i>issue</i> para alocação de tarefa.
DRE5	SP2.3	OSRMT/ Astah Community	Definição dos métodos e dados de entrada e saída dos componentes no campo <i>description</i> ; Criação de diagramas de componente contendo suas interfaces.
DRE6	SP3.1	OSRMT/ Astah Community	Criação do diagrama de atividades; Descrição de cada atividade no design que corresponde ao diagrama de atividades; Criação dos diagramas de casos de uso; Descrição dos fluxos principal e alternativo, junto com as pré e pós-condições no guia Use Case, da tela de detalhamento do requisito.
DRE7	SP3.3 SP3.4	Redmine/ Spider-CL	Instanciação e acompanhamento da tarefa (<i>issue</i>) de verificação dos requisitos; Criação do <i>checklist</i> contendo os critérios objetivos para realizar a verificação.
DRE8	SP3.5	Redmine/ Spider-CL	Instanciação e acompanhamento da tarefa (<i>issue</i>) de validação dos requisitos; Criação do <i>checklist</i> contendo os critérios objetivos para realizar a validação.



Capítulo 15

Implementação do Processo de Projeto e Construção do Produto

em [Castro e Oliveira, 2014]

Vitor de Souza Castro

Sandro Ronaldo Bezerra Oliveira

Projetos de desenvolvimento de software estão cada vez mais presentes dentro das organizações, mas para que se tenha sucesso deve-se compreender a aplicação do processo às atividades de criação do produto, desde a concepção até a entrega. No Brasil, a SOFTEX desenvolveu o programa MPS.BR - Melhoria do Processo de Software Brasileiro [SOFTEX, 2012a] com o objetivo de impulsionar a melhoria da maturidade e capacidade do desenvolvimento de software e serviços nas empresas brasileiras. Imerso no conjunto de processos contidos no MPS.BR, o processo de Projeto e Construção do Produto (PCP), que se encontra incluso no modelo MR-MPS-SW (Modelo de Referência do MPS para Software), foi o escolhido. A escolha desse processo foi realizada com base no conjunto de avaliações realizadas pelo MPS.BR que apontam somente 2 empresas certificadas no nível em que se encontra esse processo (nível D) e ao número de empresas que estão certificadas nos níveis E, F e G. Isso pode denotar que, nos próximos anos, as organizações que continuarem com o programa de melhoria do MPS.BR estarão

atingindo o nível D e com isso há necessidade de fornecer apoio para a implementação desse processo.

No século XXI o mundo do desenvolvimento de software está ganhando força e movimentando o mercado na web e para os aplicativos mobile. Para potencializar este desenvolvimento, são utilizados processos da engenharia de software que visam dar uniformidade nas atividades para a construção de um produto com qualidade. Evidências desta afirmação podem ser encontradas em: [Boria *et al.*, 2013], onde pode-se identificar o quanto as práticas ágeis estão relacionadas ao modelo de qualidade do MPS.BR; e no cenário internacional, o modelo CMMI [SEI, 2010] também apresenta estratégias para a implantação do seu modelo a partir da adoção das práticas ágeis.

Em pesquisa informal realizada nos repositórios do Simpósio Brasileiro de Qualidade de Software (SBQS) e do Workshop Anual do MPS (WAMPS) nos anos de 2008 a 2013, tendo como objetivo de identificar artigos relacionando o uso de métodos ágeis e modelos de qualidade, foi identificada a incidência em 58% dos artigos que abordam a metodologia SCRUM [Schwaber, 2009] e 25% abordam a metodologia XP - *eXtreme Programming* [Beck, 2004]. Esse elevado percentual denota o quanto essas metodologias são aplicadas no contexto de modelos de qualidade para o processo de software. Mas, em contrapartida, metodologias como FDD – *Feature Driven Development* [Pressman, 2011], Lean [Poppendieck, 2011] e TDD – *Test Driven Development* [Beck, 2003] são pouco exploradas nesse contexto. Por isso a escolha dessas para compor esse trabalho.

Assim, o objetivo do trabalho é apresentar um conjunto de métodos e práticas ágeis que forneça apoio à implementação do PCP juntamente com a identificação do nível de aderência destas práticas aos resultados esperados deste processo. Para a realização do mapeamento entre os resultados esperados do PCP com as práticas dos métodos ágeis foi realizado, primeiramente, um estudo sistemático dos resultados esperados. Foram identificados os objetivos, palavras chaves e possíveis implementações que não estão ligados a uma metodologia de desenvolvimento. Após esse estudo, o foco da pesquisa

foi direcionada à identificação dos métodos ágeis, suas práticas e a possíveis implementações já realizadas pela comunidade de software. Com o conjunto de métodos e práticas ágeis selecionados, realizou-se o mapeamento de cada resultado esperado com as práticas ágeis. Para avaliação desse mapeamento fez-se o uso da técnica de revisão por pares, detalhada a seguir. Foram descartadas as práticas que não possuíam relação com algum resultado esperado do PCP.

15.1. Trabalhos Relacionados

Foram pesquisados alguns trabalhos relacionados na literatura especializada, dentre os quais pode-se destacar o trabalho de Oliveira *et al.* (2008), que apresenta uma implementação no nível F com as metodologias SCRUM e XP, mas não são evidenciadas quais as práticas e a sua relação com os resultados esperados. Já no trabalho de Santos *et al.* (2008), foi adotado um único método ágil para o mapeamento com as atividades da ISO9001:2000, não sendo apresentadas outras possibilidades para o atendimento da norma nos casos em que o método não obteve a aderência.

O trabalho de Catunda *et al.* (2011) está centrado na implantação do MPS.BR em uma organização. A abordagem das práticas é muito ampla no sentido que não houve a associação direta das práticas aos resultados esperados. Outro ponto a destacar é que o método SCRUM foi utilizado e por se tratar de uma organização específica, os outros métodos ágeis usados neste trabalho não foram citados.

Já no trabalho de Carvalho *et al.* (2013) é realizada a análise de aderência do MPS.BR com as funcionalidades disponíveis na ferramenta proposta. O enfoque é a apresentação da ferramenta como principal apoio na implementação do MPS.BR, porém não apresenta propostas de implementação de resultados esperados que não estão contemplados na ferramenta, como os do processo PCP, foco deste trabalho.

No trabalho de Silva *et al.* (2009) a análise de aderência aos resultados esperados em todos os níveis do modelo MPS.BR são realizadas para as metodologias FDD e SCRUM. A limitação do trabalho está na análise de somente duas metodologias e na perspectiva da especificação de quais práticas são necessárias para atingir cada resultado esperado. Também não há justificativas que evidenciem o atendimento das metodologias para cada nível do MPS.BR.

Por fim no trabalho de Prikladnicki *et al.* (2010) são apresentadas práticas para a implementação do MPS.BR com métodos ágeis. O trabalho foi limitado a dois métodos ágeis e não se faz associação dos resultados esperados a cada prática. Além disso, há apenas informações superficiais das práticas para cada nível de implementação.

15.2. Métodos Ágeis Usados

Os métodos ágeis tratados neste trabalho foram selecionados porque tratam de atividades da engenharia de software e suas práticas são especialmente relacionadas com a implementação do processo de desenvolvimento do produto. O *framework* SCRUM, muito conhecido no mercado, não foi escolhido por não abordar práticas de engenharia de software, tendo o foco no contexto da gestão de projetos. A seguir tem-se uma pequena descrição dos métodos ágeis selecionados de acordo com o critério definido no início deste capítulo.

Segundo Pressman (2011), o Crystal é uma família de métodos que premia a "manobrabilidade", tendo como principal objetivo entregar softwares úteis funcionando. Os elementos centrais do Crystal são os papéis, os padrões de processo, produtos de trabalho e práticas específicas. O Lean (Poppendieck, 2011) é baseado no Sistema Toyota de desenvolvimento de produto e tem o objetivo de tornar o desenvolvimento de software de maneira "enxuta", com adaptabilidade, foco na rápida solução do problema e melhoria simultânea da qualidade. O TDD (desenvolvimento dirigido por testes) é uma metodologia

de desenvolvimento de software, na qual deve-se escrever um teste unitário - teste a nível de componente - antes de escrever o código (Beck, 2003).

Já o XP (Beck, 2004) define a codificação como a principal atividade de um projeto de software. É um método para times pequenos e médios, que desenvolvem software com os requisitos vagos e que se modifiquem rapidamente. Por fim, o FDD é um modelo prático de processo para a engenharia de software orientado a objeto (Pressman, 2011). Possui a característica de desenvolvimento que as funções valorizadas pelo cliente devem ser passíveis de entrega em duas semanas ou menos.

15.3. Apoio à Implementação do Processo PCP do MR-MPS-SW usando Métodos Ágeis

O processo PCP (Projeto e construção do Produto) tem como propósito o de projetar, desenvolver e implementar soluções para atender aos requisitos, para isso a SOFTEX (2012a) definiu 8 resultados esperados. Resultado Esperado é "um resultado observável do sucesso do alcance do propósito do processo" [ISO/IEC, 1995]. Nas subseções a seguir são indicados cada um dos resultados esperados, bem como o seu objetivo, o(s) método(s) adotado(s) para sua implementação e a(s) prática(s) presente(s) no(s) método(s). As práticas foram extraídas das literaturas usadas para referenciar cada um dos métodos descritos na seção 15.2.

15.3.1. PCP1 - Alternativas de solução e critérios de seleção são desenvolvidos para atender aos requisitos definidos de produto e componentes de produto.

O objetivo do PCP1 é identificar possíveis soluções para a construção do produto baseado em critérios pré-estabelecidos.

No Crystal foram identificadas duas práticas aderentes: a Exploração 360° e a Rearquitetura Incremental. Para uma possível implementação da primeira prática é necessário desenvolver a reunião de Exploração 360° contendo o planejamento das tecnologias e metodologias e processos a serem utilizados no projeto. O uso da metodologia *shaping*, técnica utilizada para identificar pontos fortes e fracos da organização, pode ser expandido para a análise de alternativas de soluções para o projeto. Já para a prática de Rearquitetura Incremental é necessário avaliar alternativas de solução à medida que o desenvolvimento vai evoluindo, podendo, assim, ao se deparar com uma boa solução, atualizar a arquitetura do projeto de forma incremental.

No Lean somente a prática de Adiar Comprometimento foi identificada, e o seu uso dá-se quando a decisão da escolha da solução deve ser adiada para que se possa avaliar outras soluções. O contexto da palavra "adiar" está relacionada a esperar o último momento para se tomar uma decisão irreversível. Já no TDD, a prática Projetar a Solução é concebida com o desenvolvimento de testes para o atendimento a determinado requisito do produto, assim pode-se avaliar soluções ou estratégias.

Por fim, no XP a prática Projeto Simples, objetiva identificar a solução mais simples e começar o desenvolvimento - simplicidade presumida. Assim, deve-se avaliar o menor investimento inicial que chegue na solução. Para determinar a solução, deve-se ter o pensamento de que cada parte do sistema precisa justificar a sua existência.

No contexto do PCP1 somente a prática de Exploração 360° obteve aderência total ao resultado esperado pois apresenta de forma estruturada os passos para que se consiga realizar a seleção das alternativas de solução, além de enfatizar os aspectos relacionados a força e fraqueza em análise para cada alternativa de solução.

15.3.2. PCP2 – Soluções são selecionadas para o produto ou componentes do produto, com base em cenários definidos e em critérios identificados

O objetivo do PCP2 é selecionar as soluções alternativas e reavaliá-las. Para atendimento desse resultado esperado pode-se fazer o uso das práticas de Esqueleto Ambulante do Crystal, Detalhar por *Feature* do FDD e Eliminar Desperdício do Lean.

A prática de Esqueleto Ambulante remete ao desenvolvimento de um pequeno modelo para cada solução, esse modelo pode ser um protótipo funcional, a ser analisado para assegurar que haja a possibilidade da aplicação da solução. Com a prática Detalhar por *Feature* e de posse de um modelo abrangente, por exemplo um diagrama de classe ou modelo entidade relacionamento, para cada *feature* é necessário identificar e detalhar qual a melhor solução para o projeto da *feature*. O propósito é adequar e validar a solução para cada *feature*.

Já na prática Eliminar Desperdício, as soluções devem ser avaliadas visando a eliminação de desperdício. O contexto da palavra desperdício está relacionado ao pouco valor agregado de solução para o cliente.

Para o PCP2 nenhuma prática atingiu aderência total ao resultado esperado, isso foi motivado pelo fato de nenhuma das práticas definir atividades que remetam a escolha da solução baseada em critérios.

15.3.3. PCP3 - O produto e/ou componente do produto é projetado e documentado

O objetivo do PCP3 é realizar e documentar o design do produto. As práticas Padrões de Codificação e Projeto Simples do XP, Rearquitetura Incremental do Crystal, Detalhar por *Feature* do FDD e Criar Conhecimento do Lean foram selecionadas para a implementação do resultado esperado.

No XP a prática de Padrões de Codificação possibilita a definição de um documento de arquitetura do projeto contendo os padrões a serem utilizados. Já a prática Projeto Simples consiste em conceber o projeto de maneira simples, ou seja, o design do projeto deve acompanhar a simplicidade, a complexidade do projeto sem necessidade deve ser retirada. A Rearquitetura Incremental propõe que à medida que o design for sofrendo alterações, o projeto arquitetural deverá acompanhar as modificações e ser devidamente documentado. Ao passo que o processo desenvolvimento vai ocorrendo, a adequação no modelo é necessária, sendo vital a sua documentação para garantir posterior manutenções.

Já na prática Detalhar por *Feature*, o modelo abrangente fornece a base para o detalhamento da funcionalidade. Nessa prática a *feature* deve ser projetada e documentada. Alterações no modelo abrangente podem ocorrer em virtude de nova *feature* no projeto, sendo necessária a atualização do modelo. Na prática Criar Conhecimento é necessário registrar e tornar público todos os artefatos dos projetos de acordo com a importância. Para a execução deve ser disponibilizado um ambiente para gestão dos produtos de trabalho do projeto. Essa atividade deve estar associada ao processo de gestão de configuração.

Neste resultado esperado, as práticas de padrões de codificação, Detalhar por *Feature* e Criar Conhecimento possuem aderência total ao resultado esperado em virtude das práticas indicarem formas claras de como o produto deve ser projetado e documentado. Pode-se destacar, por exemplo, a prática detalhar por *feature* porque dentro do processo de execução o enfoque está na realização do design da solução, e para isso pode-se fazer uso dos diagramas de UML – *Unified Modeling Language* para documentar.

15.3.4. PCP4 - As interfaces entre os componentes do produto são projetadas com base em critérios predefinidos

O objetivo do PCP4 é projetar as interfaces (serviços) do produto. Foram selecionadas as práticas Detalhar por *Feature* do FDD, Esqueleto Ambulante do Crystal e Otimizar o Todo do Lean. No Detalhar por *Feature* as interfaces e suas dependências devem ser identificadas a cada *feature* detalhada. Deve ser projetado como dar-se-á o comportamento diante dos outros componentes do produto, podendo fazer o uso dos diagramas de classes e de componente da UML.

A prática Esqueleto Ambulante propõe o desenvolvimento de pequenos componentes e sua posterior integração aos demais componentes por meio do design evolutivo. As interfaces são identificadas à medida que os componentes da arquitetura são desenvolvidos. Já a prática Otimizar o Todo preconiza que no processo de desenvolvimento deve-se ter a visão macro da aplicação para a identificação das interfaces e posterior otimização. O uso de um modelo de classe, entidade-relacionamento e/ou de componente pode fornecer apoio para o alcance dessa visão. A prática detalhar por *feature*, a mesma indicada para o PCP3, possui aderência total em virtude que no seu processo de execução apresenta as atividades de design para cada *feature*. Esse design prevalece o uso da UML e de outras ferramentas para o projeto das interfaces do produto.

15.3.5. PCP5 - Uma análise dos componentes do produto é conduzida para decidir sobre sua construção, compra ou reutilização

O objetivo do PCP5 é definir como será concebido o produto, podendo ser por desenvolvimento interno, aquisição ou reutilização. Somente as práticas de Modelagem de Objeto Conceitual do FDD e a Exploração 360° foram selecionadas.

A prática Modelagem de Objeto Conceitual tem o seguinte propósito: por meio da modelagem conceitual são extraídas as informações para subsidiar a decisão da construção, compra ou reuso de componentes. O modelo conceitual pode ser o diagrama de classe, modelo de entidade e relacionamento ou diagrama de atividades em nível de sistema. A partir dessa visão a equipe poderá discutir como será concebido o produto.

Já a Exploração 360° pode ser utilizada para a verificação e o estudo da viabilidade no desenvolvimento, compra ou reuso dos componentes. A metodologia *sharing* pode ser utilizada para a realização da análise. É recomendado que a equipe de desenvolvimento e os envolvidos com o projeto façam a análise juntos, objetivando buscar a melhor decisão.

Ambas as práticas indicadas nessa subseção possuem aderência total ao PCP5 em virtude de fornecer subsídios que possibilitem a decisão a cerca do componente do produto. No caso da Exploração 360°, por exemplo, na realização de sessões de "exploração", a identificação dos pontos fortes e fracos do componente influenciará diretamente nessa decisão.

15.3.6. PCP6 - Os componentes do produto são implementados e verificados de acordo com o que foi projetado

O objetivo do PCP6 é a implementação do produto de acordo com o design estabelecido. As práticas Construir por *Feature* do FDD, Programação por Pares, Testes, Refatoração e Integração Contínua do XP, Esqueleto Ambulante e Programação Lado-a-lado do Crystal, Entregar Rápido do Lean e Projetar a Solução do TDD foram selecionadas como possibilidades para o atendimento do resultado esperado.

Construir por *Feature* remete ao desenvolvimento das funcionalidades dos projetos passando pelo processo de construção e inspeção. Os componentes do produto devem atender a definição de pronto no projeto. A definição de pronto pode conter itens desde a atualização da documentação do

projeto até testes da *feature*. O desenvolvimento da *feature* deve ser de acordo com o projetado.

A Programação por Pares é uma técnica utilizada para a construção do produto na qual duas pessoas realizam o desenvolvimento juntas. Nessa prática há dois perfis: piloto, que possui o controle da atividade de programação; e o copiloto, que possui o papel de auxiliar e propor outras formas para atingir a solução. Os pares devem realizar rodízios para que haja a diversidade de concepções entre os membros envolvidos, gerando um melhor entendimento e projeto da solução.

Os Testes são realizados durante o desenvolvimento do componente de produto. Há um interface dessa atividade com o processo de Verificação – VER, que é definida de acordo com a organização. A prática Refatoração é utilizada para otimizar o código interno do componente do produto sem a alteração do seu comportamento externo. Essa atividade pode trazer alterações na estrutura e na arquitetura do projeto devendo ser documentado.

Na Integração Contínua deve-se estabelecer uma política de integração dos componentes de produto por meio de ferramentas automatizadas que realizam o merge dos códigos fontes juntamente com a execução de todos os testes (unitário, integração e aceitação). Na prática Esqueleto Ambulante, o desenvolvimento dá-se por meio da criação de uma pequena aplicação de forma simples que pode-se integrar aos demais componentes. Essa prática garante o desenvolvimento da aplicação utilizando a tecnologia definida e a arquitetura do projeto. A aplicação desenvolvida pode ser um protótipo funcional, que ao final do processo pode ser integrado ao produto.

Na Programação Lado-a-lado dois desenvolvedores trabalham lado a lado na mesma funcionalidade cada um em um terminal, possibilitando que a visualização do objetivo desenvolvido seja para os dois. Já a prática Entregar Rápido direciona o desenvolvimento do produto de maneira que a arquitetura evolua rapidamente e que antes de quaisquer mudanças, o produto já esteja entregue. Essa prática pode ser auxiliada pelo uso de bibliotecas e/ou componentes de produto que proporciona maior produtividade no desenvolvimento.

Na prática Projetar a Solução, deve ser realizada a construção de cenários de testes para a verificação do componente do produto. A verificação está relacionada à capacidade do componente do produto em atender o que foi projetado. Os cenários de teste são construídos de acordo com o projeto, garantindo o atendimento ao projetado.

Apesar de inúmeras práticas selecionadas, somente as práticas construir por *feature* e esqueleto ambulante possuem aderência total ao resultado esperado. Essa aderência é alcançada pelas atividades de programação e verificação que são evidenciadas na prática construir por *feature*.

15.3.7. PCP7 - A documentação é identificada, desenvolvida e disponibilizada de acordo com os padrões estabelecidos

O objetivo do PCP7 é identificar qual a documentação para o projeto e para o usuário final. A prática da Documentação Abrangente do XP, o Manual do Usuário gerado pelo Crystal, as Documentações de Design e do Usuário pelo FDD no processo de Construir por *Feature*, as práticas Criar Conhecimento, Integrar Qualidade e Eliminar Desperdício do Lean e a Documentação de Prática de Projeto da Solução do TDD, são possibilidades para o atendimento do resultado esperado.

A Documentação Abrangente remete a equipe de desenvolvimento a definir quais documentos serão necessários para o projeto. No Crystal, o manual do usuário é um produto de trabalho e deve estar integrado ao desenvolvimento de cada funcionalidade. Como alternativa de implantação da prática, a documentação citada pode fazer parte da definição de pronto do projeto.

No processo Construir por *Feature* no FDD, é necessária a documentação do design e do usuário, sendo que para cada nova *feature* toda a documentação deve ser atualizada. No Lean a prática Criar Conhecimento propõe que o desenvolvimento da documentação do projeto sirva para garantir

que o conhecimento seja preservado. Já na prática Integrar Qualidade o foco está em desenvolver toda a documentação de maneira padronizada, visando a qualidade do produto. O uso de modelos de documentação organizacional garante essa padronização. Em Eliminar Desperdícios, o desenvolvimento da documentação deve ocorrer somente se for necessária, visando a entrega para o cliente e a sua incorporação no produto final.

Por fim, a prática Projetar Solução do TDD rege que deve ser necessário atualizar as documentações do cenário de teste e das classes do projeto.

No PCP7 duas práticas apresentam aderência total ao resultado esperado: Documentação Abrangente e Criar Conhecimento. Essas práticas apresentam a documentação como papel importante dentro do projeto. Em Criar Conhecimento, o destaque dá-se pelo enfoque na socialização do conhecimento e por consequência a sua formatação e as formas de disponibilização.

15.3.8. PCP8 - A documentação é mantida de acordo com os critérios definidos

O objetivo do PCP8 é manter a documentação consistente e organizada do projeto. Como possibilidade de prática para atendimento do resultado esperado foram selecionadas a Documentação Abrangente do XP, Detalhar por *Feature* e Construir por *Feature* do FDD, e o Projetar Solução do TDD.

A Documentação Abrangente para esse resultado esperado tem o enfoque de manter somente a documentação necessária para o andamento do projeto. No Detalhar por *Feature* é proposto atualizar a documentação do design para cada *feature* do projeto. Já no Construir por *Feature* é necessário atualizar as documentações do projeto durante a construção do produto de acordo com a definição de pronto estabelecida. Para a prática de Projetar Solução, a documentação do projeto é desenvolvida e mantida a cada projeto de teste, por exemplo. Essa documentação de projeto é importante para dar suporte às mudanças que o software venha sofrer ao longo do tempo.

Por fim, no PCP8 somente a prática Documentação Abrangente possui aderência total em virtude da prática propor a reflexão sobre quais documentações devem ser geradas e qual o seu nível de abrangência. Isso faz com que a equipe defina, selecione e mantenha a documentação necessária para a execução do projeto.

15.3.9. Avaliação a partir da Revisão por Pares

Por meio da revisão por pares realizada junto a um avaliador líder inicial do MPS.BR, foi possível obter a avaliação necessária por um especialista. O objetivo da revisão por pares é: avaliar os critérios utilizados para a comparação dos modelos; verificar a aderência entre os elementos presentes nas estruturas dos modelos, quanto a sua correspondência e interpretação dos elementos; e analisar se as considerações feitas esclarecem suas atribuições [Pavan, 2007].

Foi aplicado um questionário para identificar o nível de conhecimento do revisor a cerca do tema em questão, identificando que o mesmo possui conhecimento alto na metodologia XP, médio conhecimento em TDD, FDD e Lean, e baixo conhecimento na metodologia Crystal. Sobre o PCP o revisor atribuiu nível alto de conhecimento, por já ter implementado este processo em algumas unidades organizacionais da indústria de software nacional. O resultado da revisão apontou mapeamento dos resultados esperados ao XP, TDD, FDD, Lean e Crystal como satisfatório e não indicou comentário para as seguintes categorias: Técnico Alto (problema em um item, se não alterado, comprometerá as considerações); Técnico Baixo (item que seria conveniente alterar); Editorial (erros de português ou texto a ser melhorado); Questionamento (dúvidas quanto ao conteúdo das considerações); e Geral (comentário geral em relação às considerações).

15.3.10. Resultados Obtidos

Na Figura 110 é apresentado o total de práticas por metodologia, sendo que a mesma prática pode estar em aderência total para um resultado esperado e para outro resultado esperado estar parcial. Nota-se que as práticas apresentadas possuem incidência com aderência total em todos os resultados esperados, exceto o PCP2, que possui somente práticas com aderência parcial. Para atingir o objetivo no PCP2 será necessária a junção de duas práticas que obtiveram o nível de aderência parcial para atingir o resultado de maneira total.

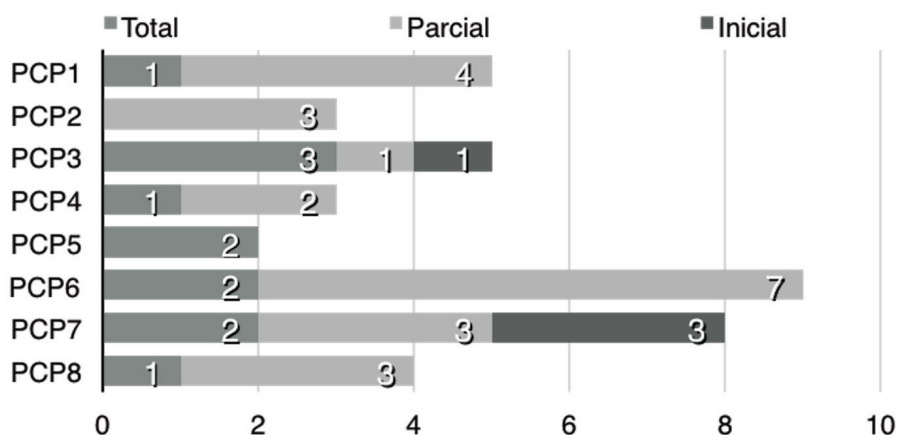


Figura 110. Total de Incidência de Práticas por Resultado Esperado do PCP

Na Figura 111 pode-se observar que a metodologia FDD possui o maior quantitativo de práticas com aderência total. Isso demonstra que o FDD pode ser utilizado como metodologia base e ser associada a outras práticas para a cobertura total de todos os resultados esperados.

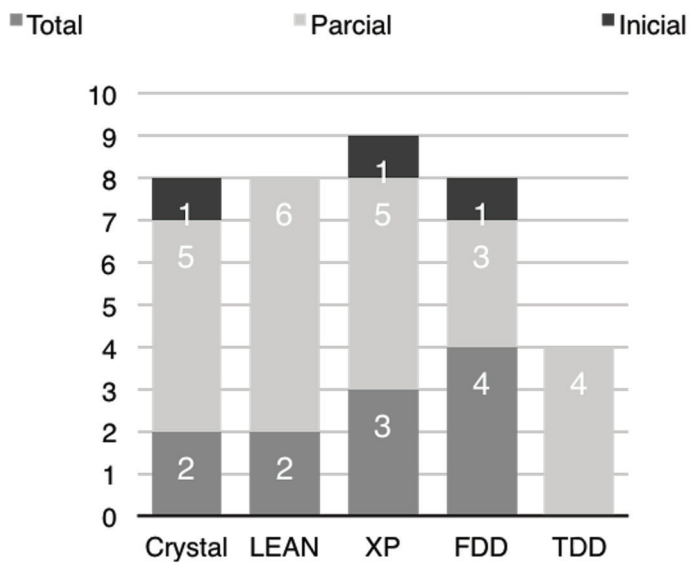


Figura 111. Total de Práticas usadas por Metodologia



Capítulo 16

Implementação dos Processos de Verificação e Validação

em [Brito Neto e Oliveira, 2013]

Olavo Nylander Brito Neto
Sandro Ronaldo Bezerra Oliveira

Objetivar qualidade é um fator crucial nas empresas desenvolvedoras de produtos e serviços correlatos, que implica tanto na melhoria destes resultados quanto no processo de produção e distribuição [SOFTEX, 2012a]. Neste contexto, surgiu o MPS-BR (Melhoria de Processo de Software Brasileiro), que trata-se de um programa mobilizador coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), tendo como objetivos a criação e aprimoramento do modelo MPS (MR-MPS) e a disseminação e adoção deste modelo em todo país com foco principal em pequenas e médias empresas (PME) e abrangendo grandes organizações.

Dentre os processos adotados pelo MR-MPS, destacam-se os processos de Verificação (VER) e Validação (VAL). O processo de Verificação tem como propósito confirmar que cada serviço e/ou produto de trabalho do processo ou do projeto atende apropriadamente os requisitos especificados, já o processo de Validação possui o propósito de confirmar que um produto ou componente do produto atenderá a seu uso pretendido quando colocado no ambiente para o qual foi desenvolvido [SOFTEX, 2012a]. Seu objetivo é assegurar que o

software cumpra com suas especificações e atenda às necessidades dos clientes que estão pagando por eles [Sommerville, 2003].

O contexto deste trabalho foca na implementação de Testes usando como base as recomendações dos processos de Verificação e Validação do MR-MPS-SW e as ferramentas: a Spider-CL, desenvolvida com o objetivo de sistematizar a criação e aplicação de critérios objetivos nas análises promovidas pela organização; e a TestLink para gerir as demais atividades necessárias no processo a fim de planejar e acompanhar as etapas necessárias. Assim, este trabalho objetiva analisar uma proposta de sistematização de Testes a partir dos processos de Verificação e Validação do MR-MPS-SW, utilizando um conjunto de ferramentas de software livre.

16.1. Trabalhos Relacionados

Muitas propostas de implementação do MPS-BR a partir da utilização de ferramentas são encontradas na literatura especializada, como pode ser visto em [Almeida *et al.*, 2011], que propõem um apoio ferramental de forma parcial aos processos de Gerência de Requisitos, Verificação e Validação em um ambiente integrado. Outro trabalho pode ser visto em [Barreto, 2006] que, através da estação TABA, apresenta a ferramenta VerificationManager com objetivo de realizar verificação de software voltada para modelo de construção de Ambientes de Desenvolvimento de Software Orientados a Organização (ADSOrgs).

No contexto da criação de um protótipo de ferramentas para geração automática de testes focando no domínio da área espacial, Ambrósio *et al.* (2008) apresentam o projeto Ambientes de Testes baseados em Injeção de Falhas por Software (ATIFS), que tem como objetivo explorar técnicas e métodos de testes para a validação de software em aplicações espaciais.

De forma semelhante ao apresentado em Yoshidome *et al.* (2010), este artigo aborda uma estratégia para a implementação de Testes usando os

processos de Verificação e Validação do MR-MPS-SW, fundamentado em boas práticas de uso de ferramentas de software livre, desenvolvidas no contexto de um projeto acadêmico, e analisa a aderência de ferramentas presentes no mercado semelhante aos artigos supracitados. A presente pesquisa apresenta uma estratégia de utilização de ferramentas já existentes de software livre (TestLink, JUnit, Selenium) em conjunto com uma ferramenta desenvolvida no Projeto SPIDER (Spider-CL) para apoio à implementação do MPS-BR.

16.2. Metodologia de Implementação dos Processos de VER e VAL com Apoio de Ferramentas Livre

Esta seção apresenta a metodologia proposta para implementação sistematizada de Testes a partir dos processos VER e VAL constantes no MR-MPS-SW, através do uso das ferramentas de software livre: (1) Spider-CL para criação de *checklist*, desenvolvido na linguagem Java e requer um servidor de aplicação e banco de dados Mysql; (2) TestLink (disponível em <http://testlink.org/>), para definição e gerência de casos de teste, oferecendo suporte aos processos de verificação e validação, desenvolvido em PHP e requer servidor web com suporte a linguagem e banco de dados Mysql; e as ferramentas de teste (3) JUnit para a implementação de casos de teste unitários integrado ao código fonte (disponível em <http://www.junit.org/>), desenvolvida para a linguagem Java e adaptada para C#, Python, Fortran e C++, e (4) Selenium para testes automatizados disponível para *browsers* (disponível em <http://seleniumhq.org/>). Ressalta-se que a escolha das ferramentas deu-se com base nos resultados obtidos e relatados em [Oliveira *et al.*, 2011].

A ferramenta Spider-CL foi desenvolvida pelo Projeto SPIDER. Apresenta uma abordagem através de desenvolvimento de *checklists*, usando critérios objetivos, que permite a aplicação em diversos contextos e mantém um registro permanente de todas as utilizações dos modelos desenvolvidos.

Seu objetivo é compor um conjunto de critérios organizados, funcionando como um *template* que pode ser reutilizado diversas vezes em várias aplicações. Mais detalhes acerca do uso da ferramenta, consultar o seu manual disponível em <http://www.spider.ufpa.br>.

Neste trabalho compreende-se metodologia como a codificação de um conjunto de práticas recomendadas, por vezes acompanhada de material de treinamento, programas de educação formal, planilhas, diagramas, tomando como parte um método (processo com uma sequência de passos, para construir um software) [Pressman, 2006]. Esta metodologia procura agregar boas práticas para o uso de softwares livres de apoio aos processos de VER e VAL.

A metodologia foi organizada tomando como base o processo proposto por Ackerman *et al.* (1989) que contempla as atividades de: **Planejar**, responsável pela identificação de produtos de trabalho e a definição de uma estratégia de verificação contendo cronograma, métodos, ferramentas, entre outros; **Especificar**, responsável pela definição de critérios e procedimentos para a realização da verificação nos produtos de trabalho; **Executar**, responsável pela realização das atividades de verificação previamente planejadas e pela identificação e registro dos defeitos detectados; e **Reportar**, responsável pela análise dos resultados das atividades de verificação e disponibilização destes para as partes interessadas.

16.2.1. Planejar

Considerando um ambiente onde estejam devidamente instaladas e configuradas as ferramentas de apoio aos processos de VER e VAL, a metodologia inicia com a definição dos critérios de maior interesse ao processo. Estes critérios têm como objetivo analisar os produtos de trabalho que serão produzidos ao longo do projeto e selecionar quais destes serão verificados e validados. Segundo SOFTEX (2012a), recomenda-se como estratégia para seleção

de produtos de trabalho as contribuições para o alcance dos objetivos e requisitos do projeto. Para esta implementação, será levado em consideração o Plano de Projeto no qual estes artefatos estão presentes. As Figuras 112 e 113 apresentam duas telas da ferramenta Spider-CL, que permite a manutenção e a aplicação de critérios objetivos a partir do uso de *checklists* como forma de seleção dos produtos de trabalho.

A imagem mostra a interface da ferramenta Spider-CL. No topo, há uma barra verde com o texto "Bem-vindo" e um ícone de ajuda. Abaixo, há uma barra de navegação com cinco abas: "Sessão", "Usuário", "Critério", "Checklist" e "Aplicar Checklist". Cada aba contém botões de ação. A aba "Critério" está selecionada, mostrando os botões "Cadastrar Critério" e "Consultar Critério". A aba "Checklist" mostra "Novo Checklist" e "Consultar Checklist". A aba "Aplicar Checklist" mostra "Aplicar Checklist" e "Visualizar Checklist".

Abas de navegação:

- Sessão:** Minha Conta, Sair
- Usuário:** Cadastrar Usuário, Consultar Usuário
- Critério:** Cadastrar Critério, Consultar Critério
- Checklist:** Novo Checklist, Consultar Checklist
- Aplicar Checklist:** Aplicar Checklist, Visualizar Checklist

Aba "Critério" selecionada, mostrando a janela "Cadastro de Critérios".

Cadastro de Critérios

Critério:

Alternativas

Alternativa 1:

Alternativa 2:

Adicionar Alternativa

Salvar Cancelar

Copyright © 2009 - Projeto SPIDER

Figura 112. Informações do *Checklist* utilizando a ferramenta Spider_CL - Cadastro de Critério

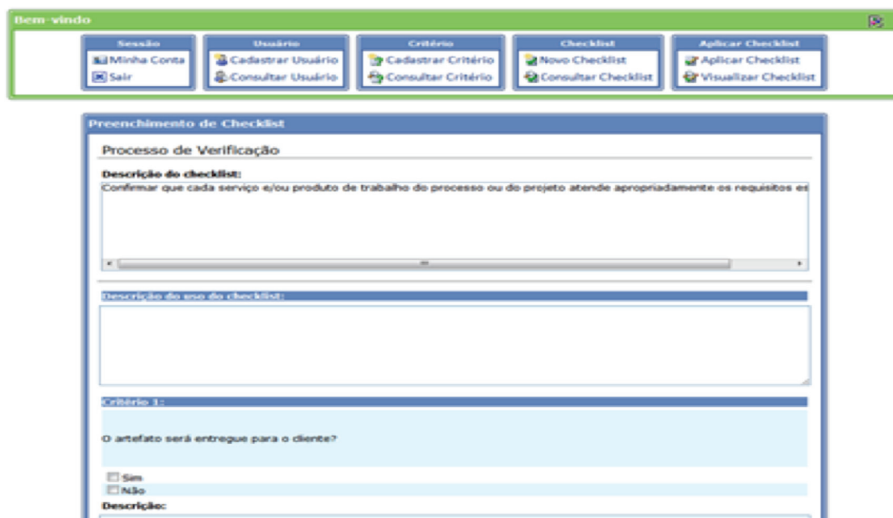


Figura 113. Informações do *Checklist* utilizando a ferramenta Spider_CL - Aplicação do *Checklist*.

Adicionalmente, a estratégia adotada para a execução do processo deve ser registrada num Plano de Verificação e Validação, que deverá ser construído a partir do Plano de Projeto citado anteriormente, definindo uma estratégia com: os (1) procedimentos; (2) a infraestrutura necessária; e (3) as responsabilidades pelas atividades pré-definidas neste plano. Estas informações podem ser registradas na ferramenta TestLink, a partir da funcionalidade "Gerenciar Plano de Teste", que auxiliará outras funcionalidades no projeto. Devem estar presentes os métodos que serão utilizados para verificação e validação de cada produto de trabalho, garantindo revisão por pares e testes [SOFTEX, 2012a]. Este plano pode ser cadastrado usando a ferramenta TestLink, na qual destacam-se alguns pontos deste plano presentes em itens específicos da ferramenta, como as funcionalidades "Gerenciar Plano de Teste", "Especificar", "Inventário", "Executar Testes" e "Relatórios e Métricas", a fim de facilitar a organização durante o acompanhamento deste processo, como pode ser visto nas colunas laterais da ferramenta, a partir da Figura 114.

16.2.2. Especificar

Após a execução da atividade de planejamento, os critérios e os procedimentos que serão utilizados na verificação e validação de cada produto de trabalho, previamente selecionado, são de fato definidos. Quanto aos critérios, estes visam representar de forma confiável os objetivos especificados junto ao cliente, visando manter a integridade do processo e consequentemente seu objetivo. Os critérios de aceitação dos produtos de trabalho podem ser geridos pela ferramenta Spider-CL e sua definição poderá ser feita a critério da organização que está implantando. O MR-MPS-SW sugere alguns critérios que podem ser usados a partir de algumas referências citadas em [SOFTEX, 2012a].

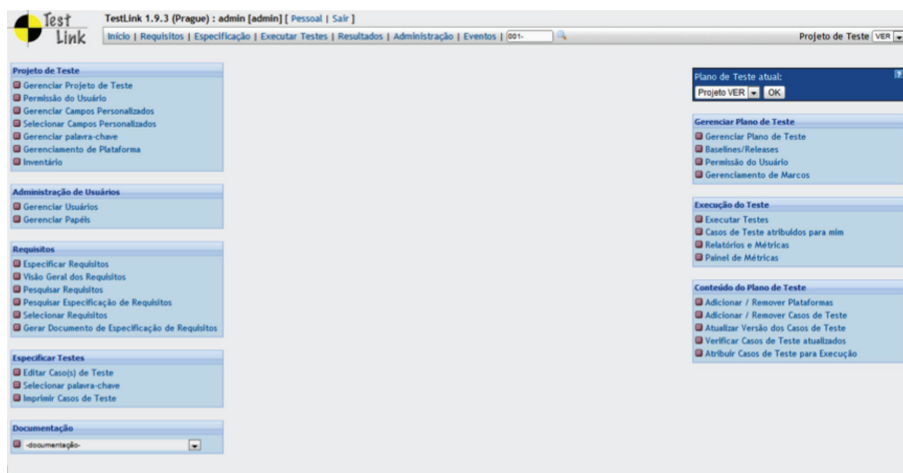


Figura 114. Tela Principal da Ferramenta TestLink

Já os procedimentos (seleção dos critérios objetivos através da Spider-CL e os casos de teste mantidos na ferramenta TestLink a partir da sua funcionalidade "Especificação", que contém a descrição do casos de testes e suas peculiaridades como pré-condição, pós-condição, resumo, ação e demais seções que serão utilizadas de acordo com a organização que fará uso) pas-sam ser armazenados na ferramenta TestLink a fim de auxiliar na execução

dos mesmos, usando a funcionalidade “Executar Testes”. Na Figura 115 é possível identificar um caso de teste que representa um procedimento a ser executado.

Após o registro dos procedimentos a serem executados, a aplicação dos mesmos será realizada a partir da ferramenta, facilitando o acompanhamento até o final do processo. Contudo, será necessário definir um ambiente para verificação e validação, podendo ser gerenciado através da ferramenta TestLink que apresenta a funcionalidade "Inventário", específica para o registro dos itens necessários na alocação de recursos. Assim, esta funcionalidade apresenta um planejamento dos itens a serem utilizados como ferramentas, recursos de hardware, infraestrutura de rede e qualquer outro item necessário.

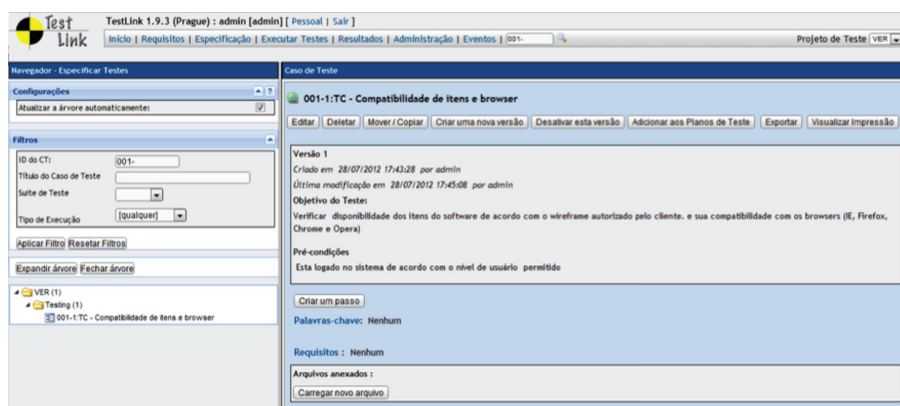


Figura 115. Procedimento a ser executado nos processos de VER/VAL.

16.2.3. Executar

Durante a execução do processo, conforme o planejado, devem estar presentes nos procedimentos de verificação e validação a revisão por pares e testes, a fim de garantir a integridade dos produtos de trabalho gerados pelo projeto durante o processo. O acompanhamento da execução é registrada

na ferramenta TestLink (ver Figura 116) e descrições podem ser incluídas para facilitar futuras correções e, assim, entregar produtos de trabalho de forma íntegra com o especificado nos requisitos. Os registros podem ser feitos similares aos casos de teste, facilitando a execução e registro dessas informações.

Os defeitos encontrados podem ser classificados a critério da organização com o intuito de definir parâmetros que possam ser considerados suficientes para o risco do projeto e o impacto na qualidade do produto. Após a identificação dos defeitos, estes devem sofrer ações corretivas, ficando a critério da organização executar novamente as estratégias de verificação e validação planejadas, para garantir que os defeitos foram removidos adequadamente e que novos defeitos não foram incluídos a partir dessas correções [SOFTEX, 2012a]. Na funcionalidade "Especificação" do TestLink são adicionadas sugestões de ações corretivas com o propósito de corrigir erros gerados durante o desenvolvimento. E em seguida, se necessário, o registro das ações corretivas efetivamente realizadas de fato são cadastradas na mesma ferramenta.

Os testes a serem executados sobre o código fonte são escolhidos a critério da organização responsável, com o intuito de expor defeitos latentes presentes e corrigi-los, antes da entrega do produto. Segundo Sommerville (2003), para um teste bem-sucedido a detecção de defeitos é aquela que faz com que o sistema opere incorretamente, e conseqüentemente, expõe os defeitos existentes. Com esse propósito são utilizadas ferramentas de testes voltadas para o código-fonte.

Uma delas é o Selenium que se apresenta integrado ao *browser* facilitando a ação de testes automatizados para aplicações web, facilitando a execução dos testes automatizados e gerando seu código fonte integrado com outros tipos de testes, como o JUnit. Este possui o propósito de testes unitários o qual, quando aplicados juntos, facilita a homogeneidade dos testes, uma vez que os testes podem ser tratados como testes unitários seguindo o padrão do JUnit, como podem ser vistos nas Figuras 117 e 118.

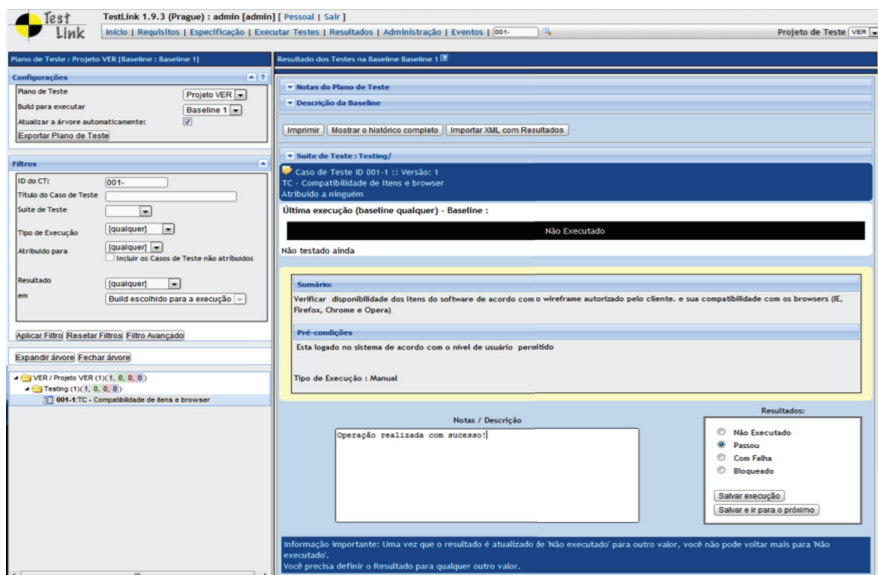


Figura 116. Execução dos procedimentos pré-cadastrados.

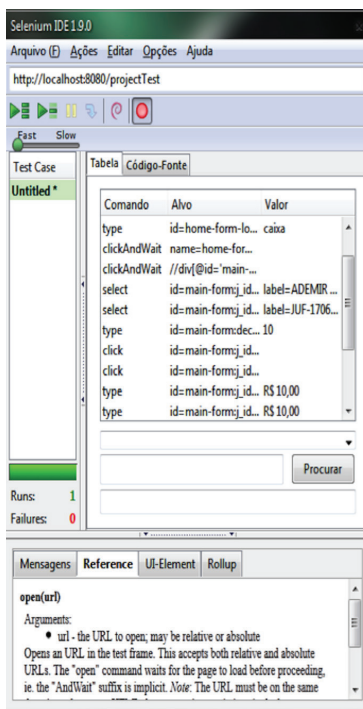


Figura 117. Ferramenta de Teste Selenium

```

import org.junit.Before;
import org.junit.Test;
import java.util.regex.Pattern;

public class Asas {

    @Before
    public void setUp() throws Exception {
        selenium = new DefaultSelenium("localhost", 4444, "chrome", "http://localhost/");
        selenium.start();
    }

    @Test
    public void testAsas() throws Exception {
        selenium.open("/projectTest/login.seam");
        selenium.type("id=home-form-login:j_id49:user-password-field", "caixa");
        selenium.click("name=home-form-login:j_id64");
        selenium.waitForPageToLoad("30000");
        selenium.click("//div[@id='main-div-content']/table/tbody/tr[2]/td/a/table/tbody/tr/td[2]");
        selenium.waitForPageToLoad("30000");
        selenium.select("id=main-form:j_id80:contract-client-field", "label=None do cliente [codename]");
        selenium.select("id=main-form:j_id98:contract-vehicle-filed", "label=JXX-1706 [2009]");
        selenium.type("id=main-form:decorate-numberOfMonths:numberOfMonths-field", "10");
        selenium.click("id=main-form:j_id135:contract-start-date-fieldPopupButton");
        selenium.click("id=main-form:j_id135:contract-start-date-fieldDayCell34");
        selenium.type("id=main-form:j_id225:contract-daily-value-field", "R$ 10,00");
        selenium.type("id=main-form:j_id242:contract-enter-value", "R$ 10,00");
        selenium.click("//tr[@id='main-form:j_id259:contract-paydays:source:1']/td");
        selenium.click("//div[@id='main-form:j_id259:contract-paydayscopy']/div");
        selenium.click("id=main-form:j_id50:anchor");
        selenium.click("id=j_id16:j_id20:anchor");
        assertTrue(selenium.getConfirmation().matches("^Deseja realmente sair do Sistema de controle de projectTest[\\s\\S]*$"));
    }

    @After
    public void tearDown() throws Exception {
        selenium.stop();
    }
}

```

Figura 118. Ferramenta de Teste JUnit aplicado junto com o Selenium.

16.2.4. Reportar

Ao fim da execução e correções necessárias, são realizadas análises dos resultados obtidos em cada atividade de verificação/validação e disponibilizada para as partes interessadas do projeto [SOFTEX, 2012a]. Para este objetivo a ferramenta TestLink gera de forma automática relatórios que facilitam o acompanhamento do projeto, numa visão mais ampla, a partir da funcionalidade "Relatórios e Métricas", mantendo presente qualquer informação que sofra alteração durante o processo. Estes relatórios podem, ainda, ser exportados em alguns formatos de arquivos populares, como podem ser observados nos diversos modelos de relatórios disponíveis na Figura 119. Por fim, estes devem ser disponibilizados para as partes interessadas, especificados nos recursos humanos definidos no Plano de Projeto, usando

algum meio de comunicação: e-mail, publicação no site interno, repositório do projeto, etc.



Figura 119. Modelos de Relatórios gerados pela Ferramenta TestLink.

De uma maneira geral, a integração entre as ferramentas pode ser vista na Figura 120, onde o processo é iniciado através da ferramenta Spider-CL a partir da geração de *checklist* que terá a finalidade de apoiar a seleção dos produtos de trabalho (artefatos) que serão verificados. A ferramenta TestLink centralizará o acompanhamento das atividades do processo: registro do plano de teste; acompanhamento dos testes em execução; registro de defeitos e correções; e disponibilização dos resultados. Complementarmente, ressalta-se que as ferramentas Selenium e JUnit são utilizadas para a execução dos testes a partir dos Casos de Testes definidos e mantidos na ferramenta TestLink.

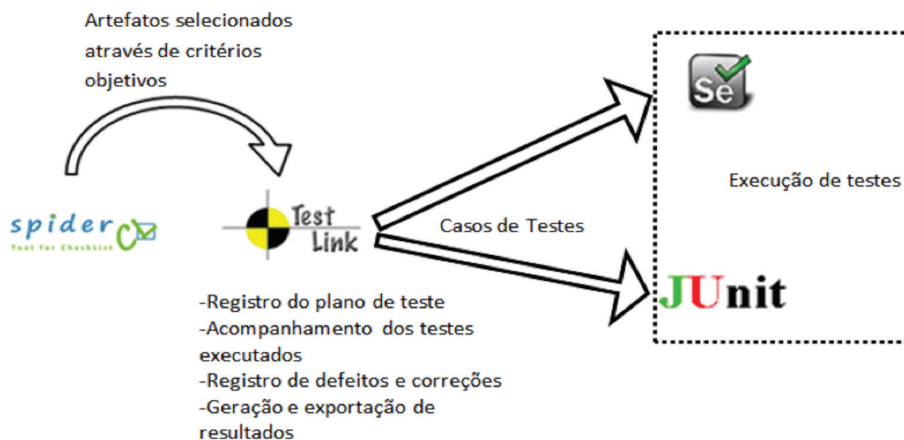


Figura 120. Comunicação entre as Ferramentas propostas.

16.3. Aderência da Metodologia Proposta ao MR-MPS

A análise de aderência entre a metodologia proposta ao modelo MR-MPS-SW é feita através do mapeamento das práticas sugeridas na Seção 16.2 com os resultados esperados dos processos de Verificação e Validação constantes no Guia Geral do MPS-BR [SOFTEX, 2012a], como visto na Tabela 1. É importante ressaltar que a metodologia restringe-se a apresentar boas práticas para que seja possível contemplar os resultados esperados dos processos de VER e VAL em conjunto com práticas definidas no processo de software da organização. Resultado Esperado (RE) é "um resultado observável quando o propósito do processo é alcançado com sucesso" [ISO/IEC, 1995].

Tabela 18. Mapeamento da Metodologia com os Resultados Esperados de VER e VAL do MR-MPS-SW

RE	Ferramentas	Funcionalidades/ Práticas
VER 1 / VAL 1	Spider-CL	Seleção de artefatos através da criação de <i>checklist</i> com critérios objetivos
VER 2 / VAL 2	TestLink	Cadastro do Plano de Verificação (teste)

RE	Ferramentas	Funcionalidades/ Práticas
VER 3 / VAL 3	TestLink	Inclusão no Plano de Verificação dos critérios e procedimentos (Plano de Teste); Registro do ambiente de verificação (Inventário)
VER 4 / VAL 4	TestLink, Selenium, JUnit	Acompanhamento dos testes e registros dos resultados (Casos de Teste); Realização de testes unitários e testes automatizados (Executar Testes e Ferramentas de Teste)
VER 5 / VAL 5	TestLink	Acompanhamento da identificação e registros de defeitos (Casos de Teste)
VER 6 / VAL 6	TestLink	Geração e exportação de relatórios e métricas (Resultados)
VAL 7	-	-



Capítulo 17

Implementação do Processo de Gerência de Riscos

em [Mendes *et al.*, 2014]

Heresson João Pampolha de Siqueira Mendes

Bleno Wilson Franklin Vale da Silva

Diego Oliveira Abreu

Diogo Adriel Lima Ferreira

Manoel Victor Rodrigues Leite

Marcos Senna Benaion Leal

Sandro Ronaldo Bezerra Oliveira

Para se tornarem competitivas, as organizações desenvolvedoras de software devem entregar produtos que satisfaçam as necessidades dos clientes de forma a garantir a confiança e satisfação [SOFTEX, 2012a]. Consequentemente, qualidade é um atributo indispensável a ser considerado em todo processo de produção do software [Koscianski e Soares, 2007].

Além disso, o desenvolvimento de software possui um aspecto não repetível, tornando esta atividade imprevisível [Koscianski e Soares, 2007]. A gerência de riscos tem como objetivo tratar as incertezas de um projeto de forma pró-ativa, evitando que se tornem problemas e prejudiquem a execução do projeto conforme planejado.

Alguns dos principais modelos de qualidade para projetos de software apresentam boas práticas para o gerenciamento efetivo de riscos, entre estes

modelos podem-se citar o MR-MPS-SW, o PMBOK [PMI, 2013], a ISO/IEC 12207 e o CMMI-DEV.

Geralmente, o gerenciamento de riscos é recomendado nos maiores níveis de maturidade da organização, e devido a pouca experiência prática nesses níveis no Brasil [SOFTEX, 2012a], torna-se importante uma ferramenta de software que sistematize e agilize as tarefas deste processo, reduzindo assim custos e facilitando o aprendizado.

Este trabalho aborda os aspectos gerais que envolvem a ferramenta Spider-RM, desenvolvida para auxiliar a implementação do processo de gerência de riscos.

17.1. A Ferramenta Spider-RM

A ferramenta Spider-RM possui licença GPL e foi concebida para apoiar a metodologia de implantação da gerência de riscos em organizações desenvolvedoras de software. A Spider-RM é subprojeto do Projeto SPIDER.

A elaboração da ferramenta deu-se a partir da necessidade de sistematizar práticas abordadas na metodologia, com o objetivo de reduzir custos e tempo ao implementar da gerência de riscos nas organizações, facilitando o aprendizado dos envolvidos em conceitos e técnicas relacionados à área. Sua utilização permite a aderência às necessidades solicitadas pelas normas, modelos e guias de qualidade envolvidos neste trabalho, pois suas funcionalidades foram baseadas na lista de boas práticas obtidas a partir do mapeamento entre os documentos de referência em qualidade.

O uso da ferramenta pode ser aplicado em diversas organizações que tenham como produto software e serviços correlatos, independente do porte da mesma, encontrando-se disponível em <http://www.spider.ufpa.br>.

17.2. As Funcionalidades da Ferramenta Spider-RM

Como mencionado anteriormente, a ferramenta Spider-RM possui licença GPL, logo é gratuita e de código aberto, possuindo as seguintes características como destaque:

- Baseada em Normas Modelos e Guias: é aderente aos resultados esperados do processo de gerência de riscos do MR-MPS-SW, às práticas específicas do processo de gestão de riscos do CMMI-DEV, às tarefas definidas na ISO/IEC 12207 e implementa práticas recomendadas no PMBOK e na ISO/IEC 16085 [IEEE, 2006];
- Fornece Evidências: a ferramenta permite registro histórico das informações relacionadas ao gerenciamento de riscos, logo os dados podem ser consultados tanto em futuros projetos, quanto em possíveis avaliações oficiais realizadas pelas instituições mantenedoras dos modelos citados acima;
- Estrutura Analítica de Riscos: a ferramenta permite a criação e a manutenção de uma EAR, possibilitando um melhor gerenciamento de riscos semelhantes, quando agrupados em uma mesma categoria. Uma EAR é importante para auxiliar durante a identificação de riscos em um projeto, principalmente quando os envolvidos ainda não possuem muita experiência prática com a área;
- Gerência de Planos de Mitigação e Contingência: possibilita a criação, acompanhamento e encerramento de planos de mitigação e contingência para os riscos, de forma que as informações relacionadas à gerência de riscos ficam concentradas em apenas um lugar e podem ser facilmente consultadas em ocasiões futuras.

A ferramenta está dividida em dois módulos: (i) o Organizacional, que possui funcionalidades relacionadas à gerência de riscos no âmbito de toda a organização, independentemente de projeto, como controle do portfólio,

estrutura analítica de riscos institucionalizada, política organizacional e algumas configurações básicas; e o (ii) de Escopo de Projeto, que realiza atividades de gerenciamento de riscos dentro do ciclo de vida de um projeto, como identificação, análise, priorização, monitoramento, mitigação e contingência de riscos.

17.2.1. Visão Geral

A Figura 121 apresenta a tela inicial da ferramenta Spider-RM, que possui uma barra superior com algumas funções relacionadas a configurações, como endereços onde os artefatos utilizados serão armazenados, opção para criar um novo projeto, ou editar a estrutura analítica de riscos organizacional. Também há um menu do tipo *combobox* para selecionar o projeto atual que deverá ser consultado nas demais funcionalidades.

Por ser monousuário, a ferramenta não necessita de cadastro de *login* e senha, porém possui a opção de editar informações relacionadas à organização e aos gerentes de cada projeto.

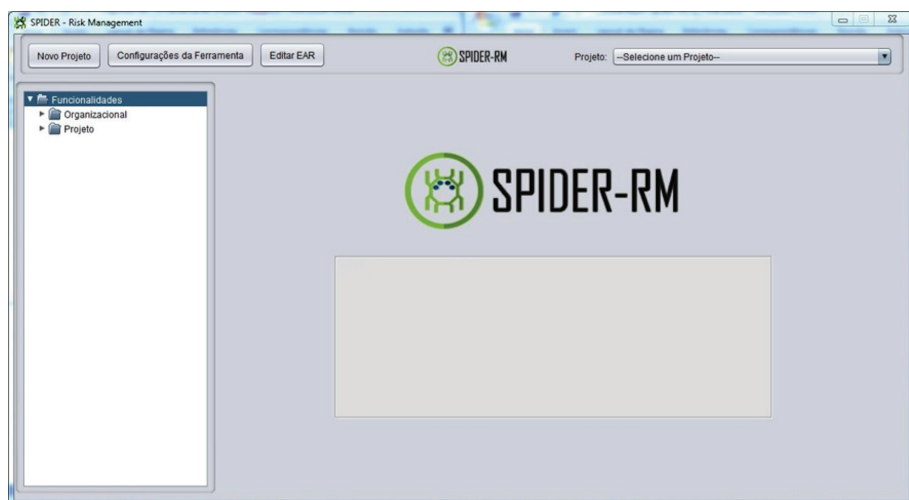


Figura 121. Tela Principal da Spider-RM

No menu à esquerda estão as funções principais que o gerente de risco poderá realizar, e estão divididas em dois grupos organizacional e projeto, que serão detalhados nas subseções seguintes.

As funcionalidades relacionadas à organização estão agregadas no menu à esquerda denominado Organizacional e possui três principais funcionalidades: Política Organizacional, Estrutura Analítica de Riscos e Portfólio.

A organização deverá definir uma política para o gerenciamento de riscos. Esta funcionalidade está relacionada com as tarefas da metodologia “Determinar Escopo da Gerência de Riscos” e “Definir Modelo de Plano do Gerenciamento de Riscos”, onde devem ser definidas as diretrizes do gerenciamento de projetos na organização e de forma genérica a todos os projetos.

A Figura 122 apresenta a tela desta funcionalidade, que deve armazenar o conteúdo da política organizacional através da importação de um arquivo, ou através da inserção manual de texto.

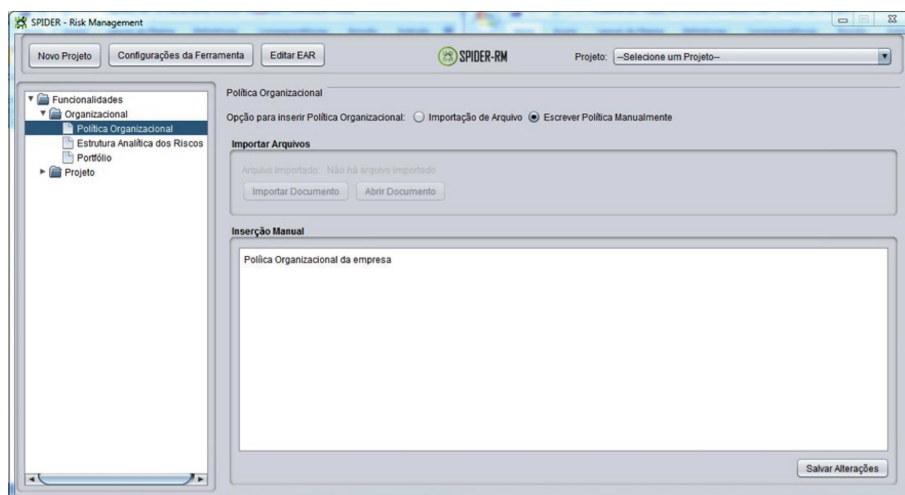


Figura 122. Tela de Política Organizacional

A funcionalidade Estrutura Analítica de Riscos relaciona-se com a tarefa “Definir Categoria de Riscos”, também da fase de planejamento da metodologia

sugerida, e com a tarefa “Identificar Novos Riscos Organizacionais” da fase de Avaliação. Como pode ser observado na Figura 123, é possível adicionar e remover itens utilizados em outros projetos à estrutura organizacional, além de consultar informações relacionadas a cada categoria, como descrição, possíveis origens, critérios para determinação de probabilidade e critérios para determinação de impacto.

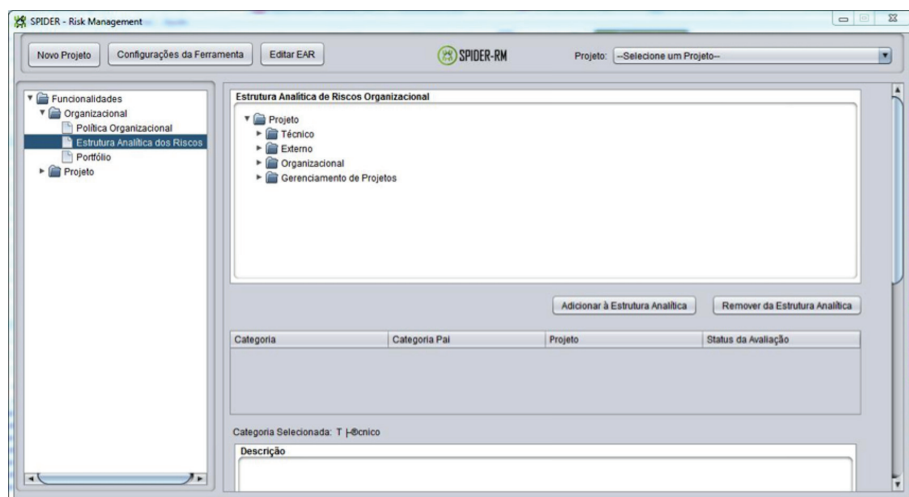


Figura 123. Tela de Estrutura Analítica de Riscos Organizacionais

A funcionalidade Portfólio originou-se a partir das tarefas “Revisar Processo de gestão de Riscos”, “Revisar tratamento de Riscos e sucessos” e “Identificar novos riscos organizacionais”, todas da fase de Avaliação da metodologia. A Figura 124 apresenta a tela de Portfólio, no qual resume os projetos cadastrados na ferramenta, identificando quais foram concluídos e podem ser avaliados.

A avaliação realizada nesta tela dá-se de duas formas, representadas pelas abas. A primeira refere-se à avaliação do projeto, que registra informações dos pontos fortes, pontos fracos, oportunidades de melhoria e outras informações adicionais que forem convenientes. Outra avaliação são das categorias adicionadas ao projeto que não constam na EAR institucionalizada,

no qual também são registradas as mesmas informações que a avaliação do projeto, para orientar a tomada de decisão na tarefa “Identificar Novos Riscos Organizacionais”.

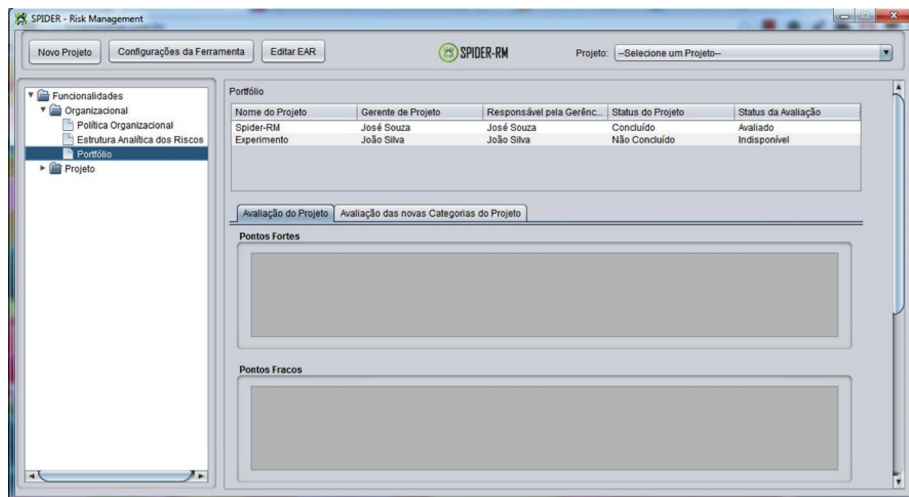


Figura 124. Tela de Portfólio

As demais funcionalidades estão agrupadas no submenu Projeto. Para acessar estas opções faz-se necessária a seleção de um projeto no menu combobox detalhado na visão geral da ferramenta. Estas funções podem ser separadas em três categorias: (i) as que são referentes ao planejamento, ou seja, as fases iniciais do projeto, que são Plano de Risco, Estrutura Analítica de Riscos do Projeto e Calendário; (ii) as funções relacionadas ao gerenciamento de riscos, que são Gerenciar Riscos, Priorizar Riscos e Riscos Ocorridos; e (iii) as funções que dizem respeito ao monitoramento de riscos, que são Analisar Riscos, Planos Pendentes e Planos Realizados.

A funcionalidade Plano de Risco está alinhada à tarefa “Definição do Plano de Gerenciamento de Riscos” da metodologia e, de forma semelhante à política organizacional, tem o intuito de registrar informações a respeito das ações que serão tomadas para administrar os riscos durante o projeto, podendo ser armazenada em um arquivo de qualquer formato que deverá ser

importado para a ferramenta ou através de texto diretamente no campo identificado para inserção manual. A Figura 125 apresenta esta tela.

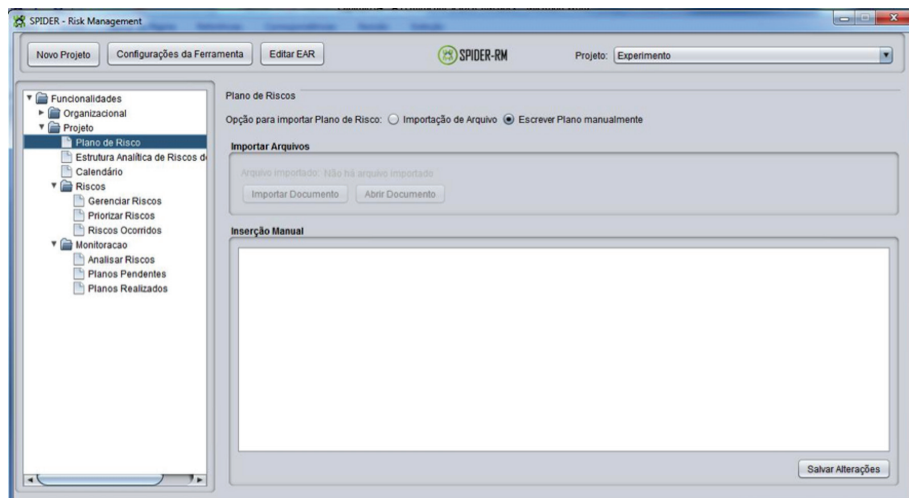


Figura 125. Tela de Plano de Risco

Associada também à tarefa “Definir Categorias de Riscos”, a funcionalidade Estrutura Analítica de Riscos do Projeto permite gerenciar a EAR instanciada de um projeto. Ao ser criado um novo projeto, automaticamente é feita uma cópia da EAR organizacional para do projeto, que tornam-se então independentes. Esta tela, observada na Figura 126, assemelha-se com a tela de Estrutura Analítica de Riscos do módulo organizacional, porém possui duas abas específicas: uma para adicionar um nova categoria avulsa não institucionalizada; e outra para adicionar novamente uma categoria da EAR organizacional que havia sido apagada após a instância da mesma.

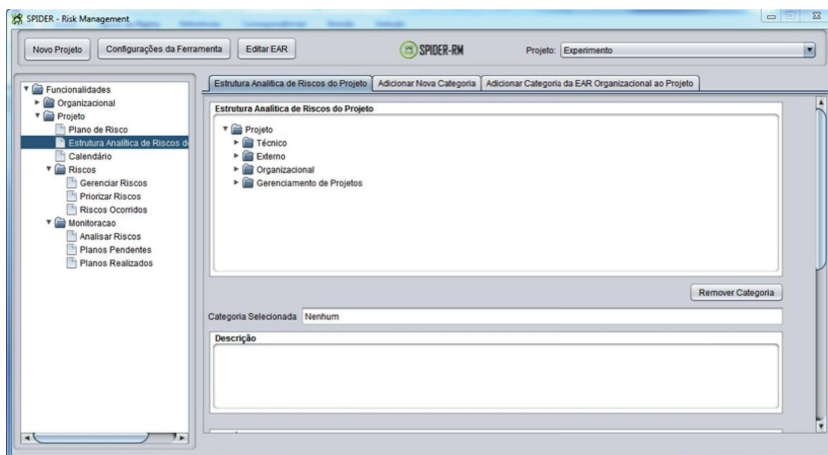


Figura 126. Tela de Estrutura Analítica de Riscos do Projeto

A funcionalidade Calendário não possui relação direta com alguma tarefa da metodologia, porém foi necessária sua inclusão para estratégias de monitoramento de forma mais clara, registrando assim deadlines para avaliar mudanças nos riscos ou executar os planos de mitigação. A Figura 127 exhibe esta tela, que possui um calendário marcando as datas importantes (Marcos ou Pontos de Controle cadastrados) e um formulário para registro, alteração, exclusão e consulta de Marcos ou Pontos de Controle.

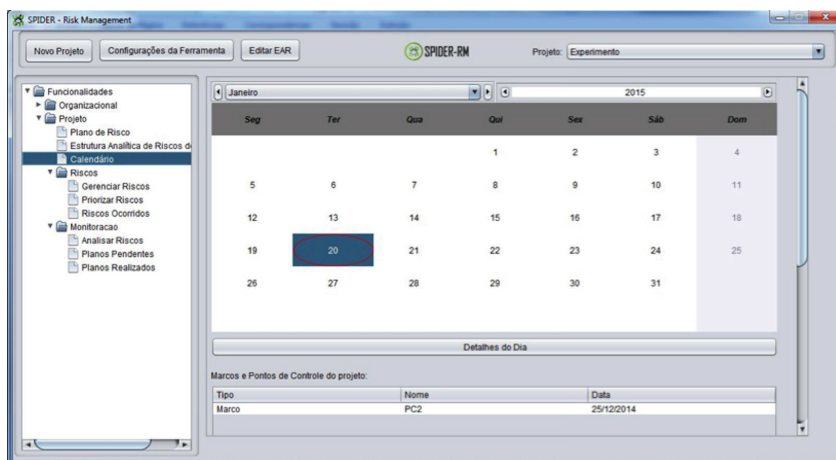


Figura 127. Tela de Calendário

A Figura 128 apresenta a tela da funcionalidade Gerenciar Riscos, que foi baseada nas tarefas “Identificação dos Riscos”, “Detalhamento de Riscos” e “Elaboração de Planos para Riscos Prioritários”, sendo responsável tanto pelo registro básico do risco, contendo identificação, descrição e categoria, quanto pelos dados relacionados à análise de riscos, evidenciados não somente pela combinação de probabilidade e impacto, mas também pelas relações entre riscos, pelas condições de ocorrência do risco, e pelo desenvolvimento de planos de mitigação e contingência, dispostos nas demais abas desta tela.

SPIDER - Risk Management

Novo Projeto | Configurações da Ferramenta | Editar EAR | SPIDER-RM | Projeto: Experimento

Funcionalidades:

- Organizacional
 - Projeto
 - Plano de Risco
 - Estrutura Analítica de Riscos
 - Calendário
 - Riscos
 - Gerenciar Riscos**
 - Priorizar Riscos
 - Riscos Ocorridos
 - Monitoracao
 - Analisar Riscos
 - Planos Pendentes
 - Planos Realizados

Abas: Informações Gerais | Relações | Condições | Plano de Mitigação | Plano de Contingência | Histórico de Alterações

Identificação de Risco: [Campo]

Data de Identificação: [Campo]

Emissor: [Campo]

Categoria de Risco: Técnico

Probabilidade: 0 %

Impacto: --Selecione--

Estado Atual: [Campo]

Descrição de Risco: [Campo]

Botões: Limpar Campos | Salvar alterações do risco | Adicionar Novo Risco | Remover Risco Selecionado

Grau de Severidade: 0

Tabela de Riscos

Identificação	Probabilidade	Impacto	Grau de Severidade	Prioridade	Estado
R1	20	Alto	15,0	0	Novo
R2	45	Alto	33,75	0	Novo
R3	20	Médio	10,0	0	Novo
R4	50	Crítico	50,0	0	Novo

Figura 128. Tela de Gerenciar Riscos

A priorização de riscos baseia-se na tarefa da metodologia denominada “Definição de Riscos Prioritários”, e como pode ser visto na Figura 129, sua tela dispõe de uma lista de riscos cadastrados, ordenados pelo grau de severidade calculado automaticamente, através do produto entre impacto e probabilidade. Caso seja conveniente para o usuário é possível alterar a ordem de prioridade manualmente.

Na tela responsável pela priorização, também há uma funcionalidade para selecionar quais destes riscos serão monitorados, uma vez que os

recursos dos projetos são limitados e pode ser inviável o monitoramento de todos os riscos de forma simultânea. Ao apertar o botão “Selecionar risco para monitorar” surge uma caixa de diálogo apresentada na Figura 130, que permite ao usuário marcar quais riscos serão monitorados durante o projeto, desde que já possuam um plano de mitigação definido. A qualquer momento durante o projeto ambas as telas podem ser acessadas para repensar a priorização e consequentemente reavaliar os riscos que necessitam ser monitorados.



Figura 129. Tela de Priorizar Riscos

Os riscos que foram selecionados para monitoramento passam a ter seu status alterado para “Mitigando”. Todo risco, após ser cadastrado, passa a ter seu status definido como “Novo”, e a primeira alteração deste valor ocorre na situação aqui especificada.

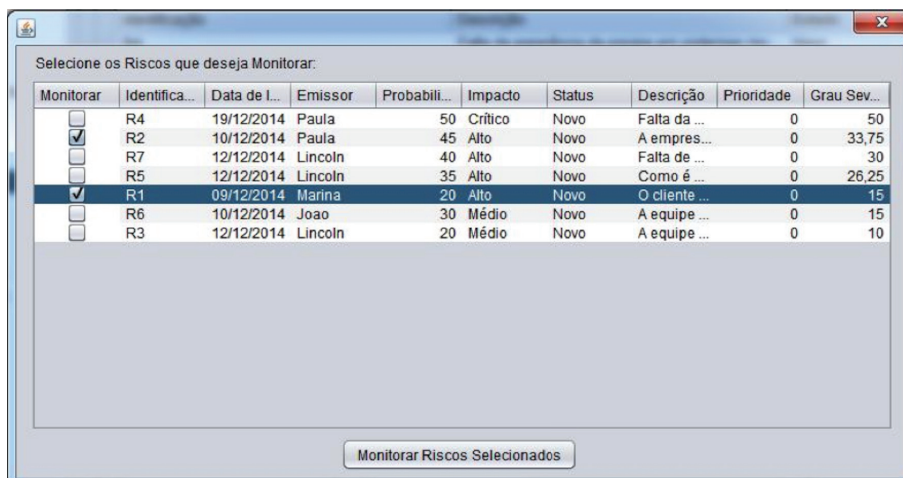


Figura 130. Tela da Seleção de Riscos para Monitoramento

A funcionalidade Riscos Ocorridos, apresentada na Figura 131, é basicamente uma tela para consulta de riscos que tiveram sua ocorrência deliberada durante o projeto, gerando assim um relatório de fácil acesso às suas informações.

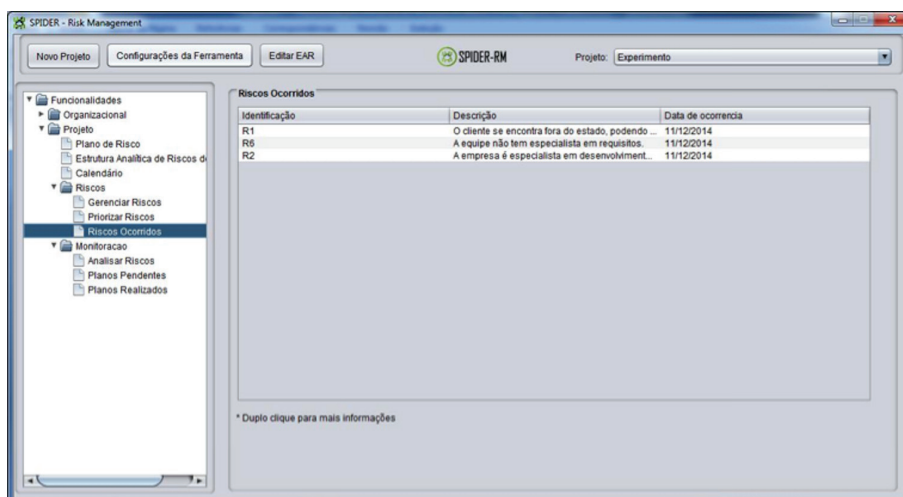


Figura 131. Tela de Apresentação de Riscos Ocorridos

A Figura 132 exibe a funcionalidade de Seleção de Risco para Análise, onde os riscos selecionados para monitoramento ficam pendentes até que sejam analisados antes do marco ou ponto de controle ao qual foi definido. Esta funcionalidade está relacionada à tarefa “Avaliação e Monitoramento de riscos” da metodologia sugerida.

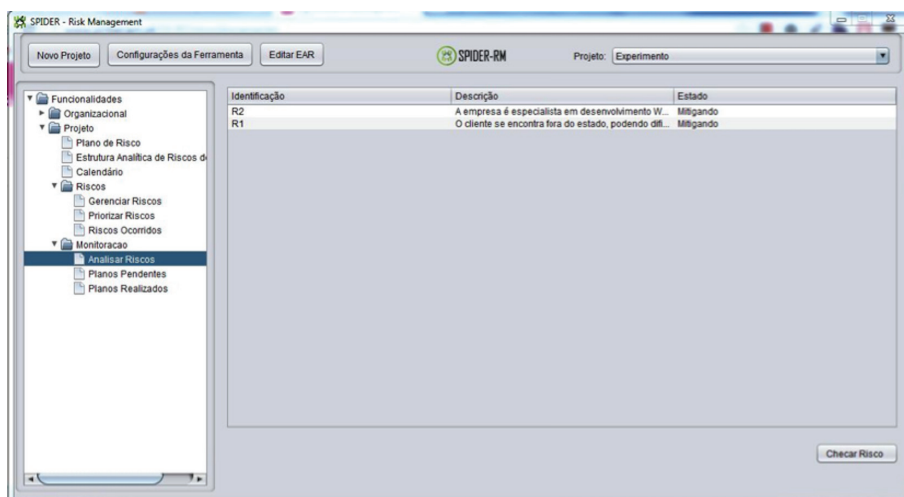


Figura 132. Tela de Seleção de Risco para Análise

Após selecionado o risco a ser analisado e o botão “Checar Risco” ser pressionado, é aberta uma caixa de diálogo especificada na Figura 133, que apresenta dados importantes a respeito do risco envolvido, e é onde é registrado o monitoramento do risco. Nesta tela é possível alterar novamente o status do risco que neste momento está com o valor “Mitigando”. As opções para alteração são os valores “Contingenciando”, quando o risco ocorreu, ou “Novo”, quando o risco não ocorreu. Neste último caso o risco volta a ficar aguardando ser selecionado novamente para um novo monitoramento.

Identificação do Risco: R1

Descrição do Risco:

O cliente se encontra fora do estado, podendo dificultar a interação com a equipe.

Condições para a ocorrência do risco:

Status	Id	Identificação	Descrição
<input type="checkbox"/>	3	Condição1	Caso a equipe ou cliente não possam viajar para realizar ...

Risco ocorreu: Não

Próximo status do Risco: Novo

Confirmar Ocorrências

Figura 133. Tela da Análise de Risco

A tela da funcionalidade de Execução de planos pendentes (Figura 134) está relacionada com a tarefa “Execução do Plano de Mitigação ou Contingência” e apresenta uma lista de planos que precisam ser executados antes da data limite especificada. Após a mudança de status de um risco um plano é acionado, de Mitigação, caso o status seja “Mitigando” ou de Contingência, caso seja “Contingenciando”.

Ao ser selecionado um plano, abrirá uma caixa de diálogo com os detalhes cadastrados dos planos, identificando entre outras informações, como deverá ser realizado.

Os planos executados devem ser marcados na coluna “Realizar” da tela, e em seguida deve ser pressionado o botão “Salvar realização de planos”, para que as alterações sejam registradas e o plano selecionado seja omitido da lista.

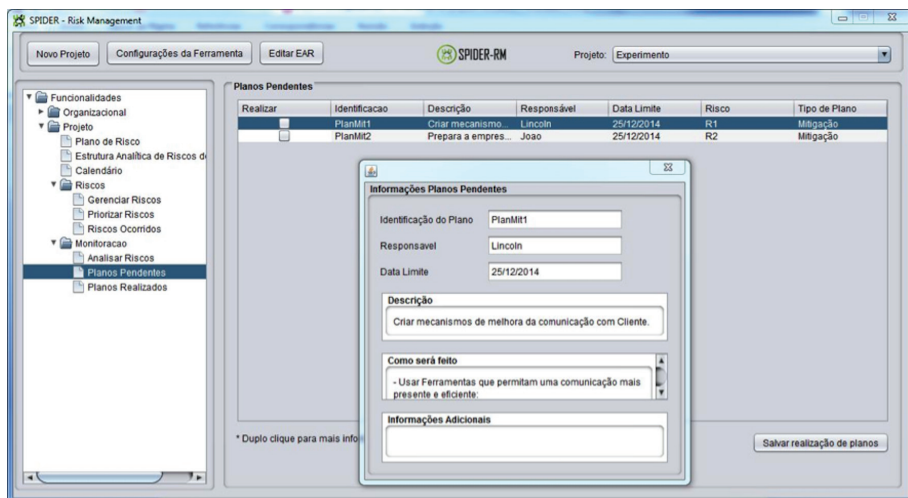
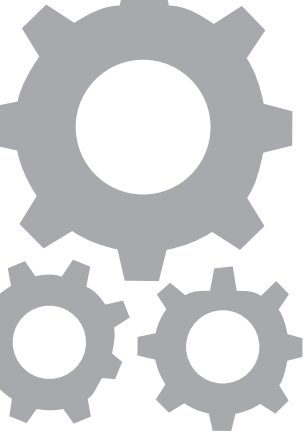


Figura 134. Tela da Execução de Planos Pendentes

A tela da funcionalidade de Planos Realizados, apresentada na Figura 135, é basicamente uma tela para consulta dos planos já executados e omitidos da tela anterior, gerando assim um relatório de fácil acesso às suas informações.



Figura 135. Tela da Apresentação dos Planos Realizados



PARTE IV

ABORDAGENS DE APOIO À IMPLEMENTAÇÃO INDIRETA DOS PROCESSOS CONSTANTES NOS NÍVEIS GERENCIADO E DEFINIDO

Sandro Ronaldo Bezerra Oliveira

Alexandre Marcos Lins de Vasconcelos

Além dos processos constantes nos níveis de maturidades tratados nas Partes II e III deste livro, um conjunto de outras abordagens devem ser levadas em consideração quando da implementação dos programas de melhoria presentes em modelos de qualidade. Assim, esta parte do livro tratará de algumas destas abordagens, retratando como estas podem orientar na efetiva implementação dos processos discutidos nos capítulos anteriores.

Assim, esta parte do livro fornecerá:

- orientações para a implementação de Estimativas dos Projetos (Capítulo 18);
- recomendações para a implementação de Critérios Objetivos (Capítulo 19);
- sugestões para a implementação de Simulação para o Ensino de Melhoria do Processo de Software (Capítulo 20);
- apoio para a Avaliação dos Processos de Software seguindo os métodos constantes nos modelos de qualidade (Capítulo 21);

- sugestões de abordagens para a implementação de rastreabilidade de requisitos (Capítulo 22);
- apoio à execução do processo de software (Capítulo 23);
- uma metodologia ágil de apoio à implementação do processo de Gerência de Riscos (Capítulo 24);
- apoio a partir de um *framework* para a implementação dos processos de Gerência de Portfólio de Projetos (Capítulo 25) e Gerência de Recursos Humanos (Capítulo 26).



Capítulo 18

Implementação de Estimativas de Projetos

em [Yoshidome et al., 2011], [Macedo e Oliveira, 2011], [Baldez et al., 2010]

Ewelton Yoshio Chiba Yoshidome

Gleise Pinheiro Baldez

Kleverton Oliveira Macedo

Maurício Ronny de Almeida Souza

Sivaldinho de Oliveira Barbos

Wallace Michel Pinto Lira

Sandro Ronaldo Bezerra Oliveira

No contexto da Engenharia de Software, a Estimativa é a disciplina que busca determinar a quantidade de esforço, recursos, dinheiro e tempo necessários para o desenvolvimento de um sistema ou produto de software específico [Pressman, 2006]. Neste sentido, visa evitar problemas clássicos no desenvolvimento de produtos de software, como prazos estendidos, alto gasto dos recursos e abandono de projetos.

No contexto do MPS.BR [SOFTEX, 2012a], estimar projetos em termos de tamanho, esforço e custo faz parte do escopo do processo de Gerência de Projetos (GPR), bem como a definição de uma base histórica de estimativas. No CMMI-DEV (SEI, 2010), estimativas de projeto constituem uma meta específica da área de processo *Project Planning* (PP), a SG 1 *Establish Estimates*.

Portanto, este trabalho apresenta um conjunto de ferramentas de software livre (gratuitas e de código aberto) que implementam técnicas e métodos de estimativas de tamanho (APF - Análise de Ponto por Função e UCP - Pontos por Casos de Uso) e esforço e custo (com o método CoCoMo – *Constructive Cost Model*). Adicionalmente, é proposta uma metodologia de uso para as ferramentas aderentes aos resultados esperados do processo de GPR do MR-MPS-SW e práticas específicas da área de processo de PP do CMMI-DEV, e analisada a conformidade aos modelos.

18.1. Trabalhos Relacionados

Foram encontrados na literatura pesquisas e estudos de caso envolvendo estimativas utilizando análise de Ponto por Função, Ponto por Caso de Uso e o método CoCoMo.

Kusumoto *et al.* (2010) descrevem como classificar automaticamente a complexidade dos atores e casos de uso originados de uma modelagem de caso de uso a partir da utilização de uma ferramenta de apoio desenvolvida para atender esta demanda. A ferramenta criada foi desenvolvida para o idioma japonês, limitando a sua utilização. No entanto, a técnica desenvolvida pode ser aplicada em conjunto com qualquer ferramenta que suporte à estimativa utilizando Pontos por Caso de Uso. Anda *et al.* (2001), em seu estudo, sugerem que a aplicação de Pontos por Caso de Uso deve ocorrer em conjunto com outros métodos que provêm estimativas *bottom-up*, como o CoCoMo.

Com relação ao método de Pontos por Função, sua utilização é ampla no mercado atual, devido à sua utilização na negociação de contratos. Isto ocorre pois a unidade de medida desta métrica, o ponto de função, mensura fragmentos do trabalho que serão entregues pela organização desenvolvedora. Neste contexto, Bagheri (2010) apresenta uma abordagem para estimar ajustes de custo em projetos que foram negociados utilizando pontos por função.

Há ferramentas que apresentam suporte à estimativa de Pontos por Caso de Uso e Pontos por Função, geralmente proprietárias, tais como: Enterprise Architect (disponível em <http://www.devdirect.com/>), Estimate Easy Use Case (disponível em <http://www.duversa.com>) e APFplus (disponível em <http://www.ivanmecenas.ecn.br/apf.htm>), o que dificulta a sua disseminação e a possibilidade de customização funcional. O método CoCoMo apresenta ferramentas livres de apoio, alguns exemplos incluem: USC-COCOMOII (disponível em http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.html) e Costar (disponível em <http://www.softstarsystems.com/demo.htm>). No entanto, estas ferramentas não apresentam integração do CoCoMo com os métodos APF ou UCP, como requerido pelos modelos MR-MPS-SW e CMMI-DEV.

18.2. Ferramentas de Estimativas

As ferramentas apresentadas constituem pesquisas do Projeto [Oliveira *et al.*, 2011]. As ferramentas, propostas neste trabalho, possuem o objetivo de prover, ao Gerente de Projeto, soluções sistematizadas, simples e de fácil utilização, e que permitam a manutenção de uma base histórica de estimativas dos projetos da organização, substituindo o uso convencional de planilhas eletrônicas, que não atendem as necessidades gerenciais por completo.

A opção por ferramentas de Software Livre (gratuitas e de código aberto) foi tomada por permitir liberdade de executar, distribuir, estudar e alterar o código sem custos financeiros, garantindo, assim, a disponibilização do código fonte para que as ferramentas definidas possam ser adaptadas para diversos contextos de organizações ou estudos.

As ferramentas Spider-APF e Spider-UCP [Baldez *et al.*, 2010] têm o objetivo de prover ao Gerente de Projeto uma forma sistematizada de estimar projetos por APF [Albrecht e Gaffney, 1983] e por UCP [Karner, 1993], respectivamente, assim como fornecer uma base histórica dos dados dos projetos desenvolvidos, com os seus valores estimados. A Spider-CoCoMo [Macedo

e Oliveira, 2011] é uma ferramenta de software livre concebida para apoiar de forma sistematizada estimativas de custo e esforço de projetos de software, a partir do modelo CoCoMo [Boehm, 1981]. A escolha pelas ferramentas resultou no atendimento às características propostas pelo Projeto SPIDER e pela integração funcional que as ferramentas possuem entre os métodos usados, aspecto requerido pelos modelos MR-MPS-SW e CMMI-DEV. As ferramentas e seus respectivos Manuais de Usuário, encontram-se disponíveis em <http://www.spider.ufpa.br>.

18.3. Metodologia de Uso das Ferramentas

Entende-se metodologia como sendo um conjunto de práticas, procedimentos e regras utilizadas por aqueles que trabalham em uma disciplina [PMI, 2013]. Neste trabalho, portanto, é proposto um conjunto de práticas utilizando ferramentas de software livre. Estas práticas procuram agregar recomendações para o uso das ferramentas apresentadas na Seção 18.2 e melhor discutidas nas Seções 18.4 e 18.5.

Vale ressaltar que esta metodologia não pretende descrever detalhes de uso das ferramentas, os quais se encontram disponíveis nos seus respectivos manuais de usuário. A metodologia colabora com o trabalho de Yoshidome *et al.* (2010), que retrata a implementação do processo de Gerência de Projetos do MPS.BR.

18.3.1. Dimensionando Tarefas e Produtos de Trabalho do Projeto a partir de Métodos Apropriados

Conforme o Guia do MPS.BR [SOFTEX, 2012a], a estimativa de tamanho é a dimensão das funcionalidades sob o ponto de vista do usuário. Podem ser contadas tabelas internas e externas ao sistema, classes, objetos, relatórios,

telas, consultas a banco de dados, cálculos, transações e atores dos casos de uso, linhas de código, entre outros. Para esta metodologia de implementação, as técnicas de APF e UCP foram adotadas para estimar este tamanho, justificando a utilização da Spider-APF e Spider-UCP.

O primeiro passo da utilização das ferramentas é a definição de um "Padrão" para a organização, isto é, um parâmetro que define os pesos de complexidade dos objetos que serão estimados. No caso da Spider-UCP, esses pesos são definidos para Atores, Casos de Uso, Fatores Técnicos e Fatores Ambientais. Na Spider-APF, são aplicados para definir a complexidade das funções. Uma vez definidos estes "Padrões", eles estarão disponíveis como parâmetros para estimar os projetos da organização.

A próxima tarefa é registrar o projeto que será dimensionado, cadastrando suas informações gerais: o nome do projeto; o padrão que deve ser usado; e a data e hora, recuperados do sistema. Na Spider-APF, devem ser adicionadas as informações sobre o tipo de contagem (Aplicação, Manutenção ou Projeto em desenvolvimento), necessárias para a Análise de Pontos de Função.

Posteriormente, deve-se quantificar os fatores. Para a estimativa por APF, a tarefa posterior ao registro do projeto na ferramenta Spider-APF é o registro de fatores de ajuste. Esta consiste em atribuir um nível de influência que as características não-funcionais podem exercer no projeto. Para a análise dos pontos por casos de uso, na Spider-UCP são levados em consideração dois fatores: os técnicos e os ambientais. Ambas as ferramentas permitem registrar novos fatores ou reutilizar outros previamente registrados em outros projetos.

Por fim, é necessário o registro das informações do projeto que serão utilizadas para o cálculo de estimativa: (i) No caso da Spider-UCP deve-se registrar os Atores e/ou Casos de Uso; ou (ii) no caso da Spider-APF são definidas Funcionalidades do projeto que podem ser do tipo [Albrecht e Gaffney, 1983]: Arquivos Lógicos Internos (ALI); Arquivos Lógicos Externos (ALE); Entrada Externa (EE); Saída Externa (SE); e Consulta Externa (CE).

Uma vez registrados todos os detalhes do projeto, podem ser realizados os cálculos da Análise por Ponto de Função (Spider-APF) ou Pontos por Casos de Uso (Spider-UCP). Os resultados da estimativa podem ser visualizados a partir de relatórios gerados pelas ferramentas. Estes relatórios apresentam gráficos gerados na Spider-UCP, os quais são semelhantes na Spider-APF, que ilustram a evolução dos cálculos de estimativa ao longo do projeto, procurando convergir o dimensionamento estimado para o real tamanho, conforme o cálculo é refinado e atualizado. Esta funcionalidade provê o atendimento ao requisito da base histórica de estimativas requerido pelos modelos MR-MPS-SW e CMMI-DEV.

18.3.2. Estimando Esforço e Custo para a Execução das Tarefas e dos Produtos de Trabalho

As estimativas de esforço e custo são, normalmente, baseadas nos resultados de análises utilizando modelos e/ou dados históricos aplicados ao tamanho, atividades e outros parâmetros de planejamento [SOFTEX, 2012a]. Desta forma, esta metodologia utiliza o método CoCoMo para estimar custo e esforço, sendo utilizada a ferramenta Spider-CoCoMo.

O primeiro passo é a criação de um projeto na Spider-CoCoMo. Cada projeto da organização deverá ser criado na ferramenta seguindo os parâmetros do método CoCoMo [Boehm, 1981]: orgânico, semi-destacado e encaixado.

Em seguida, é necessário inserir, como entrada, o tamanho estimado do projeto, que pode ser importado da Spider-UCP ou da Spider-APF (também é possível informá-lo manualmente, caso a organização não adote as técnicas APF ou UCP). Com isto, é possível realizar o cálculo do CoCoMo. A ferramenta dá suporte aos dois tipos de cálculo do CoCoMo, básico e intermediário [Macedo e Oliveira, 2011].

Para cada cálculo aplicado em um determinado projeto será criado um novo nó filho para o projeto, identificado pela data e hora do cálculo e o tipo de CoCoMo aplicado. Isto garante a definição de uma base histórica consistente baseada nos projetos já realizados pela organização, organizada com o objetivo de facilitar futuras estimativas e o entendimento do comportamento dos projetos da organização.

É importante, então, manter e comunicar os resultados. Para isto, a Spider-CoCoMo permite a geração de relatórios de dois tipos: (i) um mostrando os dados de um cálculo, como os valores dos parâmetros utilizados e os resultados do cálculo; e (ii) outro mostrando um gráfico de linha contendo a evolução do projeto em relação aos valores estimados, que permite a observação do refinamento da estimativa do projeto, apontando alterações de esforço e custo (decorrente de variações no escopo do projeto, atualizações no seu dimensionamento ou ajuste dos cálculos). Esta funcionalidade prevê o atendimento ao requisito da base histórica de estimativas requerido pelos modelos MR-MPS-SW e CMMI-DEV.

18.3.3. Análise da Metodologia

O processo de Gerência de Projetos do MR-MPS-SW, constante no Nível de Maturidade G, possui dois Resultados Esperados (práticas) que abordam diretamente a estimativa de tamanho (GPR 2), e custo e esforço (GPR 4) [SOFTEX, 2012a]. Vale salientar que o resultado esperado GPR 4 sofre evolução no Nível de Maturidade E, a partir do qual a realização de estimativas e do planejamento do projeto é feita com base em um repositório de estimativas e no conjunto de ativos de processo organizacional. Um resultado esperado é um resultado observável do sucesso do alcance do propósito do processo, ou seja, é um dos passos necessários a ser evidenciado para que um determinado processo seja implementado.

Semelhantemente, o CMMI-DEV define na área de processo *Project Planning* a meta específica SG 1 – *Establish Estimates*. Meta específica trata de características que descrevem o que deve ser implementado para satisfazer uma área de processo. Contidas na meta específica SG1, duas práticas específicas atendem diretamente as estimativas de tamanho (SP 1.2 – *Establish Estimates of Work Product and Task Attributes*) e de esforço e custo (SP 1.4 – *Estimate Effort and Cost*) [SEI, 2010]. Práticas específicas são atividades consideradas importantes na satisfação de uma meta específica associada.

O principal objetivo da proposta de metodologia de uso das ferramentas é fornecer uma alternativa sistematizada para estimar projetos, em conformidade com os resultados esperados e as práticas específicas mencionados. Para sintetizar e explicitar este atendimento, a Tabela 19 apresenta a relação entre os resultados esperados e as práticas específicas com as ferramentas e suas respectivas práticas e funcionalidades, que apoiam sua implementação.

A ferramenta Spider-CoCoMo apenas apoia o atendimento ao GPR 4 no que tange a definição de um repositório de estimativas, não se baseando no conjunto de ativos de processo organizacional, como esperado pela prática.

Tabela 19. Aderência da Metodologia de Uso das Ferramentas aos Resultados Esperados do MR-MPS-SW e às Práticas Específicas do CMMI-DEV

MR-MPS-SW	CMMI-DEV	Ferramenta(s) de Apoio	Funcionalidades/Práticas
GPR 2	SP 1.2	Spider-UCP Spider-APF	<ul style="list-style-type: none"> • Registro de projetos; • Dimensionamento de projetos através do cálculo por Análise por Pontos de Função ou Pontos por Caso de Uso; • Geração de relatórios.
GPR 4 (Até o Nível F)	SP 1.4	Spider-CoCoMo	<ul style="list-style-type: none"> • Estimativa de custo e esforço a partir do cálculo do método CoCoMo; • Geração de relatórios.

MR-MPS-SW	CMMI-DEV	Ferramenta(s) de Apoio	Funcionalidades/Práticas
GPR 4 (a partir do Nível E)	SP 1.4	Spider-CoCoMo	<ul style="list-style-type: none"> Definição de uma base histórica das estimativas dos projetos já realizados ou em desenvolvimento para apoiar futuras estimativas.

18.4. Uma Visão das Funcionalidades das Ferramentas Spider-APF e Spider-UCP

Como já mencionado, as ferramentas Spider-APF e a Spider-UCP têm o objetivo de prover ao Gerente de Projeto uma forma sistematizada simples e fácil de estimar projetos por ponto de função e por pontos de caso de uso, assim como fornecer uma base histórica dos dados dos projetos desenvolvidos, com os seus valores estimados. Levando em consideração, ainda, que nem toda empresa possui uma ferramenta específica para estimar tamanho de software e que em muitas destas são usadas planilhas eletrônicas para realizar tal estimativa, e que estimar tamanho de software em planilhas eletrônicas não permite a geração de uma base histórica.

A seguir um detalhamento de uso das principais funcionalidades existentes nas ferramentas Spider-APF e Spider-UCP.

18.4.1. Cadastrar Padrão

A atividade de identificar se existe um Padrão para estimar o projeto inicia quando o usuário identifica que não há um parâmetro que defina os pesos de complexidade dos objetos que serão estimados. Assim, o usuário acessa a área de registro de padrão e, dependendo da ferramenta, informa os dados para cadastro.

Para a ferramenta Spider-UCP o usuário acessa a área de registro do padrão e informa os pesos para: Atores, Casos de Uso, Fatores Técnicos e Fatores Ambientais. Em seguida o usuário confirma este padrão, deixando o mesmo disponível, para utilização em projetos. Caso já exista um padrão que atenda a estimativa do projeto, o fluxo é identificado na próxima atividade de fornecer informações do Projeto a ser estimado, como pode ser visto na Figura 136.

Para a ferramenta Spider-APF deve-se informa os pesos dos pontos de funções não ajustados, deixando o mesmo disponível para utilização em projetos. Caso exista um padrão que atenda ao projeto, o fluxo é identificado na próxima atividade de fornecer informações do projeto, como pode ser visto na Figura 137.

	Peso		Peso
Sistema distribuído	2.0	Portabilidade	2.0
Tempo de resposta	2.0	Facilidade de mudança	1.0
Eficiência	1.0	Concorrência	1.0
Processamento complexo	1.0	Recursos de segurança	1.0
Código reusável	1.0	Acessível por terceiros	1.0
Facilidade de instalação	0.5	Requer treinamento especial	1.0
Facilidade de uso	0.5		

Figura 136. Definição do Padrão na Spider-UCP

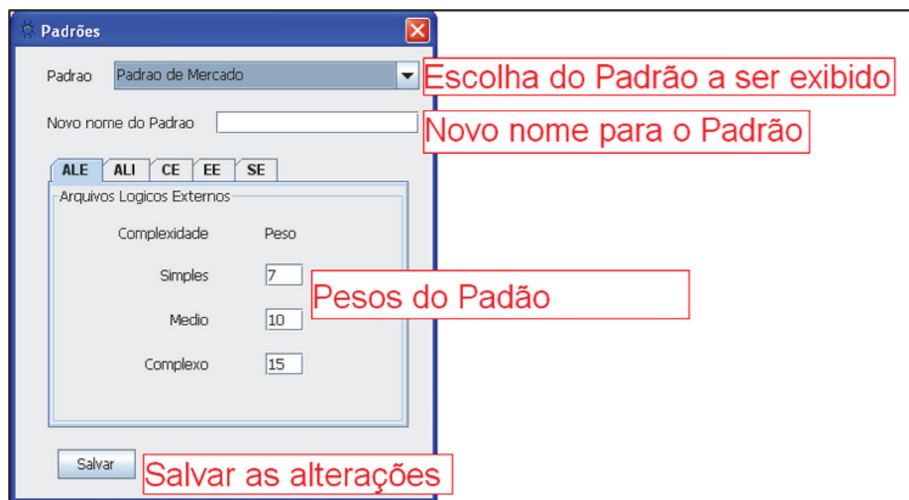


Figura 137. Definição do Padrão na Spider-APF

18.4.2. Cadastrar Projeto

O cadastro de projetos está presente em ambas as ferramentas, esta funcionalidade define informações do projeto para qual serão realizados os cálculos das estimativas. Para o cadastro de projeto na ferramenta Spider-UCP o usuário deve informar: o nome do projeto; o padrão que deve ser usado; e a data e hora, recuperados do sistema; como pode ser visto na Figura 138. De forma semelhante a Spider-APF cadastra os mesmos campos e adiciona uma informação adicional sobre o tipo de contagem (Aplicação, Manutenção ou Projeto em desenvolvimento), levando em consideração a análise por pontos de função. Todos os projetos cadastrados ficam exibidos em forma de árvore a esquerda do sistema, como pode ser visto na Figura 139.

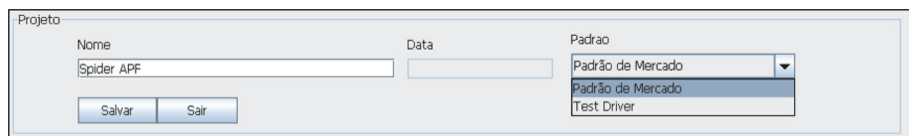


Figura 138. Cadastro de Projeto na Spider-UCP

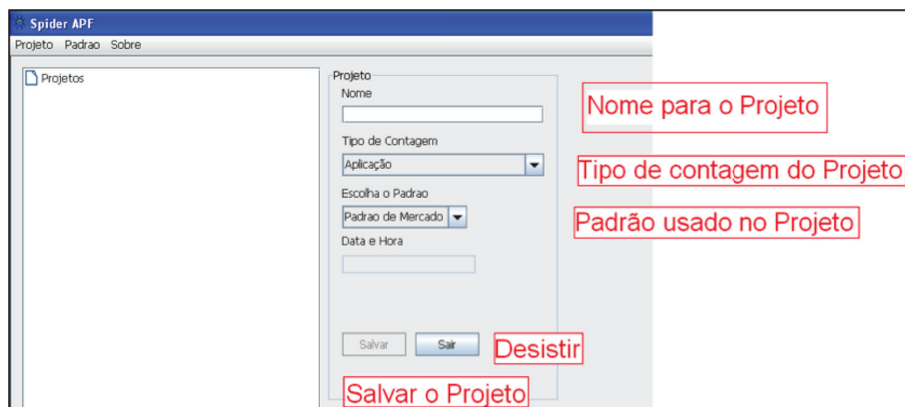


Figura 139. Cadastro de Projeto na Spider-APF

18.4.3. Quantificar e Registrar Fatores

Posteriormente, deve-se quantificar os fatores. Para a análise dos pontos por casos de uso, que é realizado pela Spider-UCP são levados em consideração dois fatores: os técnicos e os ambientais; como pode ser visto na Figura 140.

Primeiramente são cadastrados os Fatores Técnicos (TFactor), que permitem que sejam quantificados os fatores técnicos do projeto, como: Sistema Distribuído, Performance, Portabilidade, entre outros; podendo ser realizada de duas formas:

- registrando os pesos dos fatores, que podem variar de 0 a 5, onde 0 significa que o fator é irrelevante e 5 significa que o fator é essencial para o projeto;
- ou usando um recurso do sistema que visa o reuso de fatores já cadastrados, carregando assim estas informações de um projeto já existente que atenda as necessidades deste projeto. Caso estes fatores sejam carregados, o usuário ainda tem a possibilidade de editá-los. Após quantificar os fatores, o usuário confirma o registro e a ferramenta realiza o cálculo do total de Fatores de Complexidade

Técnica a partir da seguinte fórmula: $FCT = 0.6 + (0.01 * TFT)$ [Karner, 1993], onde TFT equivale ao somatório dos pesos dos fatores técnicos.

A próxima atividade é a quantificação dos Fatores Ambientais (EFactor), que permite que sejam quantificados os fatores ambientais do projeto, como: Experiência da Equipe, Motivação, Capacidade, Domínio, entre outros; podendo ser realizada de duas formas:

- registrando os pesos dos fatores;
- ou carregando de um projeto já existente que atenda as necessidades deste projeto. Caso estes fatores sejam carregados, o usuário ainda tem a possibilidade de editá-los. Após quantificar os fatores ambientais o usuário confirma o registro e a ferramenta realiza o cálculo dos fatores ambientais por meio da fórmula: $FA = 1,4 + (-0,03 * TFA)$ [Karner, 1993], onde o TFA é o total de pesos dos fatores ambientais definido.

Já na análise de pontos por função, realizada pela ferramenta Spider-APF, posteriormente ao cadastro de projeto a atividade registrar fatores de ajuste é realizada, como pode ser visto na Figura 141. Esta consiste em atribuir um nível de influência que estes podem ter na estimativa, ou seja, caracteriza as características não-funcionais do projeto: Comunicação de Dados, Performance, Atualização On-line, entre outros.

Cálculo

Data:

Fator Técnico

Fatores Técnicos

Carregar de outro Cálculo

Sistema distribuído:

Tempo de resposta:

Eficiência:

Processamento complexo:

Código reusável:

Facilidade de instalação:

Facilidade de uso:

Portabilidade:

Facilidade de mudança:

Concorrência:

Recursos de segurança:

Acessível por terceiros:

Requer treinamento especial:

Fator Ambiental

Fatores Ambientais

Carregar de outro Cálculo

Familiaridade com o RUP ou outro processo formal:

Experiência com a aplicação em desenvolvimento:

Experiência em orientação a objetos:

Presença de analistas experientes:

Motivação:

Requisitos estáveis:

Desenvolvedores em meio expediente:

Linguagem de programação difícil:

Pontos Por Caso de Uso

Atores e Casos de Uso

Carregar de outro Cálculo

Atores

Qtde

Simples:

Médio:

Complexo:

Carregar de outro Cálculo

Casos de Uso

Qtde

Simples:

Médio:

Complexo:

Salvar Sair

Figura 140. Quantificar Fator na Spider-UCP

Projeto

Nome:

Tipo de Contagem:

Escolha o Padrão:

Data e Hora:

Cálculo

Pontos PF:

Carregar Fatores de Outro Cálculo

Fatores de Ajuste

Comunicação de Dados	<input type="checkbox"/>	Performance	<input type="checkbox"/>
Processamento de Dados Distribuído	<input type="checkbox"/>	Volume de Transação	<input type="checkbox"/>
Configuração de Equipamento	<input type="checkbox"/>	Interface com Usuário	<input type="checkbox"/>
Processamento Complexo	<input type="checkbox"/>	Atualização On-line	<input type="checkbox"/>
Facilidade de Implantação	<input type="checkbox"/>	Reusabilidade	<input type="checkbox"/>
Facilidade de Mudança	<input type="checkbox"/>	Facilidade Operacional	<input type="checkbox"/>
Entrada de dados On-line	<input type="checkbox"/>	Múltiplos Locais	<input type="checkbox"/>

Data do Cálculo:

Fator de Ajuste:

Salvar Sair

Figura 141. Quantificar Fator na Spider-APF

Estas informações podem ser carregadas (reusadas) de outro projeto, caso haja um que satisfaça a condição do novo projeto. Caso o usuário escolha a opção de carregar os fatores de outro projeto, ele ainda tem a opção de editá-los, caso seja necessário, e em seguida confirmar o uso. Caso contrário, ele pode apenas registrar novos fatores que satisfaçam o projeto. Em

ambos os casos, após a confirmação do usuário, a ferramenta deve: realizar o cálculo dos fatores de ajuste ($FA = (NIT \times 0,01) + 0,65$ [Albrecht e Gaffney, 1983]); e armazenar a data e a hora deste cálculo, com a finalidade de gerar uma *baseline* da estimativa do projeto. A variável NIT representa o nível de influência total, ou seja, o somatório dos níveis de influência atribuídos a todos os fatores de ajuste.

18.4.4. Registrar Atores e/ou Casos de Uso

Após quantificar os fatores técnicos e ambientais na ferramenta Spider-UCP, deve-se registrar os Atores e/ou Casos de Uso, permitindo que esta possa ser realizada de duas maneiras: carregando um ator e/ou caso de uso, que atenda as necessidades do novo projeto e depois de carregado o usuário ainda pode editá-los; ou registrando um ator e/ou casos de uso novo. Caso o usuário registre um novo ator e/ou caso de uso, ele deve informar o nome e a complexidade para o projeto, como pode ser visto na Figura 142.

Calculo
 Data: 28/10/2010 12:25:09

Fator Técnico: 0,855
 Fatores Técnicos:
 Carregar de outro Cálculo

- Sistema distribuído: 0
- Tempo de resposta: 2
- Eficiência: 2
- Processamento complexo: 1
- Código reusável: 3
- Facilidade de instalação: 3
- Facilidade de uso: 4
- Portabilidade: 4
- Facilidade de mudança: 1
- Concorrência: 0
- Recursos de segurança: 1
- Acessível por terceiros: 1
- Requer treinamento especial: 1

Fator Ambiental: 0,815
 Fatores Ambientais:
 Carregar de outro Cálculo

- Familiaridade com o RUP ou outro processo formal: 2
- Experiência com a aplicação em desenvolvimento: 3
- Experiência em orientação a objetos: 3
- Presença de analistas experientes: 0
- Motivação: 5
- Requisitos estáveis: 3
- Desenvolvedores em meio expediente: 5
- Linguagem de programação difícil: 3

Pontos Por Caso de Uso: 43,900

Atores e Casos de Uso
 Carregar de outro Cálculo

Atores
 Qtde

- Simples: 0
- Médio: 0
- Complexo: 1

Casos de Uso
 Qtde

- Simples: 12
- Médio: 0
- Complexo: 0

Alterar Sair

Figura 142. Quantificar Atores e Casos de Uso na Spider-UCP

18.4.5. Definir Funcionalidades

Na ferramenta Spider-APF a próxima atividade consiste em definir Funcionalidades, a qual permite ao usuário a possibilidade de carregar funcionalidades provenientes de outro projeto, caso exista algum que atenda as mesmas necessidades, ou registrar novas funcionalidades, que podem ser do tipo [Albrecht e Gaffney, 1983]: Arquivos Lógicos Internos (ALI); Arquivos Lógicos Externos (ALE); Entrada Externa (EE); Saída Externa (SE); e Consulta Externa (CE); como pode ser visto na Figura143. Da mesma forma, caso o usuário escolha a opção carregar funcionalidade, ele ainda pode alterar a quantidade atribuída a cada uma delas ou apenas confirmar. Caso contrário, ele deve: inserir uma nova funcionalidade; quantificar as funcionalidades; e selecionar a complexidade.

Funcionalidades

Nome da funcionalidade

ALE		ALI		CE		EE		SE	
	Qtde		Qtde		Qtde		Qtde		Qtde
<input type="checkbox"/> Simples	0	<input type="checkbox"/> Simples	0	<input type="checkbox"/> Simples	0	<input type="checkbox"/> Simples	0	<input type="checkbox"/> Simples	0
<input type="checkbox"/> Medio	0	<input type="checkbox"/> Medio	0	<input type="checkbox"/> Medio	0	<input type="checkbox"/> Medio	0	<input type="checkbox"/> Médio	0
<input type="checkbox"/> Complexo	0	<input type="checkbox"/> Complexo	0	<input type="checkbox"/> Complexo	0	<input type="checkbox"/> Complexo	0	<input type="checkbox"/> Complexo	0

Figura 143. Quantificar Funcionalidades na Spider-APF

18.4.6. Realizar Cálculos

Por fim, após registrar atores e/ou casos de uso na Spider-UCP, realiza-se a atividade Calcular UCP, que realiza o cálculo final da análise por pontos de casos de uso, a partir da seguinte fórmula $UCP = PCUNA * FCT * FA$ [Karner, 1993], onde: PCUNA são os Pontos de Casos de Uso Não Ajustados; o FCT é o Fator de Complexidade Técnica; e FA é o Fator Ambiental; como visto na Figura 142.

Já na ferramenta Spider-APF, como atividade final, tem-se a atividade Calcular FPA, que realiza o cálculo final da análise por pontos de função que consiste em calcular os pontos de funções ajustados por meio da seguinte fórmula $AFD = ADD * FA$, onde: ADD é a contagem não ajustada das funções do projeto; e FA é o fator de ajuste da contagem APF; como pode ser visto na Figura 144.

Fatores de Ajuste	
Comunicação de Dados	1.0
Processamento de Dados Distribuído	0.0
Configuração de Equipamento	1.0
Processamento Complexo	0.0
Facilidade de Implantação	2.0
Facilidade de Mudança	0.0
Entrada de dados On-line	5.0
Performance	0.0
Volume de Transação	0.0
Interface com Usuário	3.0
Atualização On-line	3.0
Reusabilidade	0.0
Facilidade Operacional	0.0
Múltiplos Locais	0.0

Figura 144. Cálculo na Spider-APF

18.5. Uma Visão das Funcionalidades da Ferramenta Spider-CoCoMo

A Spider-CoCoMo é uma ferramenta oriunda das pesquisas do projeto SPIDER. A ferramenta, bem como o seu Manual de Usuário, encontram-se disponíveis em <http://www.spider.ufpa.br>.

Sua concepção foi dada pela necessidade de uma forma sistematizada e simples para realizar estimativas de projetos de software, pois a maioria das empresas utilizam planilhas eletrônicas e com as mesmas não é possível atender as necessidades dos gerentes por completo, tendo em vista que não é possível obter uma base histórica dos valores estimados nos projetos da organização.

A Spider-CoCoMo está inserida no contexto do processo Gerência de Projetos do MR-MPS-SW, auxiliando nas estimativas de custo, prazos e número de pessoas. A ferramenta está indiretamente ligada com o resultado esperado GPR 2 e diretamente ligada com o GPR 4, dois dos resultados esperados deste processo.

O primeiro deles, o GPR 2, visa garantir que as tarefas e os produtos de trabalho sejam dimensionados utilizando métodos apropriados. A ferramenta necessita que este resultado esperado seja cumprido, pois ele serve de parâmetro para o cálculo do CoCoMo. Em particular para este resultado esperado, o projeto SPIDER possui duas ferramentas para medir tamanho de projeto baseado no método de análise de pontos por função e pontos por casos de uso, que são a Spider-APF e a Spider-UCP [Baldez *et al.*, 2010]. Caso seja adotado qualquer outro tipo de método que não os citados anteriormente, pode ser inserido manualmente o valor na Spider-CoCoMo sem que haja perda nos valores estimados.

O segundo resultado esperado, o GPR 4, requer que o esforço e os custos sejam estimados. Esse resultado esperado é totalmente atendido com a utilização da Spider-CoCoMo, tanto nos níveis G e F, tendo em vista que o método CoCoMo é utilizado para estimativas, como dito anteriormente, como nos níveis superiores ao nível F, pois os valores estimados são armazenados em banco de dados, sendo feito um histórico dos mesmos. Este é o grande diferencial da Spider-CoCoMo sobre as planilhas eletrônicas.

18.5.1. Principais Funcionalidades

A Spider-CoCoMo possui três funcionalidades fundamentais e que servirão para evidenciar que o resultado esperado citado na seção anterior foi cumprido. A primeira delas é a de criação de projetos. Cada projeto da organização que necessita dos valores estimados deverão ser criados na ferramenta seguindo os parâmetros do método CoCoMo, a saber, orgânico, semi-destacado e

encaixado. Depois de criado, o projeto aparecerá como um novo nó na árvore de projetos, como a Figura 145 indica.

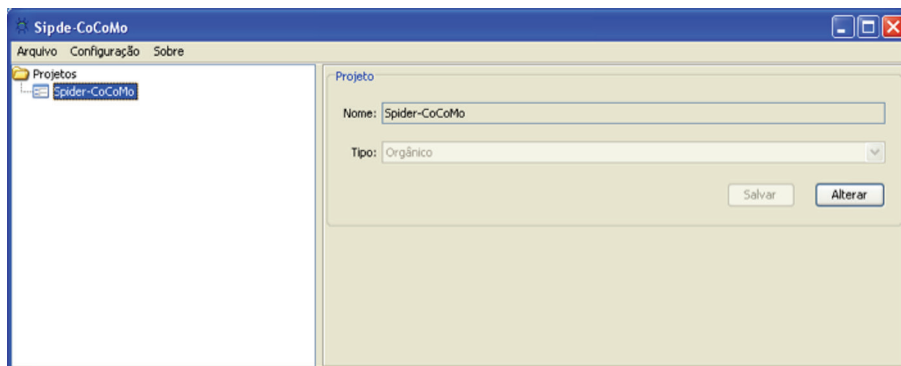


Figura 145. Projetos na Spider-CoCoMo.

Outra funcionalidade, a mais importante da ferramenta, é a de cálculo do método CoCoMo. A ferramenta dá suporte aos dois tipos de cálculo do CoCoMo, básico e intermediário. O que diferencia um cálculo do outro é o parâmetro EAF, destacado de vermelho na Figura 146, que se referem aos fatores influentes nos valores estimados.

Nesse momento do cálculo é necessário o tamanho do projeto, podendo ser em pontos por função ou de casos de uso. O usuário pode optar por importar esse valor a partir da Spider-UCP ou Spider-APF, ou ainda informar o valor manualmente, caso utilize uma outra técnica de estimativas. É importante enfatizar que a Spider-CoCoMo não utiliza o serviço das duas ferramentas anteriores, e sim acessa a base de dados das mesmas. Para cada cálculo aplicado em um determinado projeto será criado um novo nó filho para o projeto, identificado pela data e hora do cálculo e o tipo de CoCoMo aplicado. Os valores de esforço, tempo de desenvolvimento e número de pessoas estimados ficam sempre destacados de amarelo.

The screenshot displays the 'Sipde-CoCoMo' application window. On the left, a tree view shows 'Projetos' with 'Spider-CoCoMo' selected. The main area is divided into sections: 'Projeto' with fields for 'Nome' (Spider-CoCoMo) and 'Tipo' (Orgânico), and buttons 'Salvar' and 'Alterar'; 'Valores Estimados' with three yellow input fields for 'Esforço:', 'Tempo:', and 'Pessoas:'; and 'Cálculo' with radio buttons for 'Informar Manualmente', 'Importar de Spider-APF', and 'Importar de Spider-UCP', followed by 'Tamanho:' and 'Cálculo Utilizado:' fields, an 'Importar' button, and a 'Linhas de Código:' field. At the bottom, an 'EAF:' field is highlighted with a red rectangle, next to a 'Calcular EAF' button, and a 'Calcular CoCoMo' button is at the very bottom.

Figura146. Cálculo de CoCoMo.

A terceira funcionalidade da ferramenta, que merece destaque, refere-se ao mecanismo para geração de relatórios. O programa MPS.BR requer que evidências de um resultado esperado sejam geradas para garantir o atendimento da boa prática, dessa forma os relatórios da Spider-CoCoMo servem de evidências que o GPR 4 está sendo efetivamente implementado pela organização. Há dois tipos de relatórios, um mostrando um gráfico de linha contendo a evolução do projeto a nível dos valores estimados, e um mostrando os dados de um cálculo, como os valores dos parâmetros utilizados e os resultados do cálculo. Esses relatórios estão respectivamente representados nas Figuras 147 e 148.

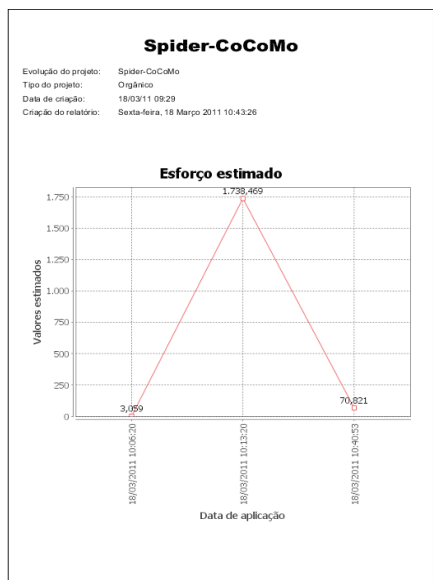


Figura147. Relatório de Evolução de Projeto

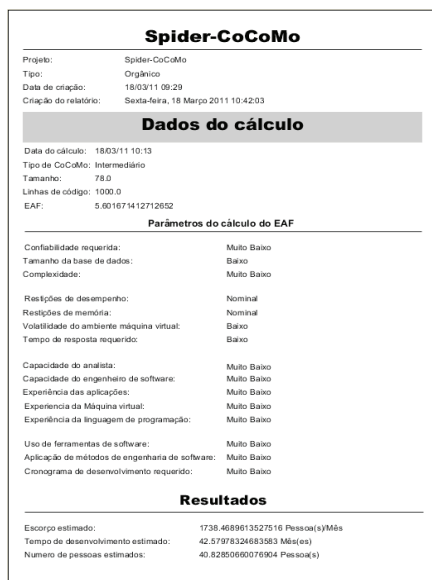


Figura148. Relatório do Cálculo

Para entender melhor o fluxo desde a criação do projeto até a realização de um cálculo de CoCoMo, as Figuras 149 e 150, representadas por um diagrama de atividades, mostra todo o fluxo respectivamente.

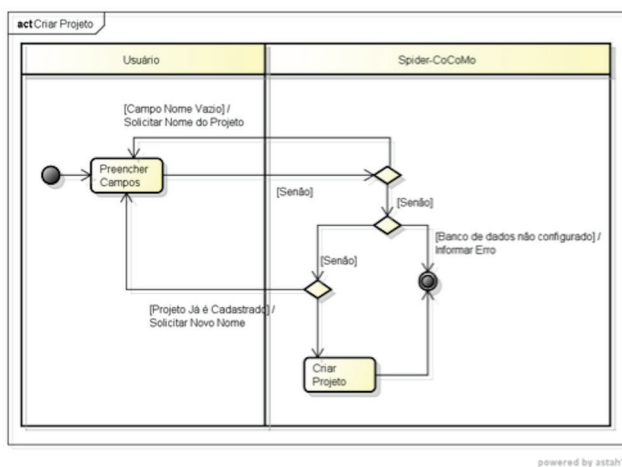


Figura149. Diagrama de Atividades para Cadastrar Projeto

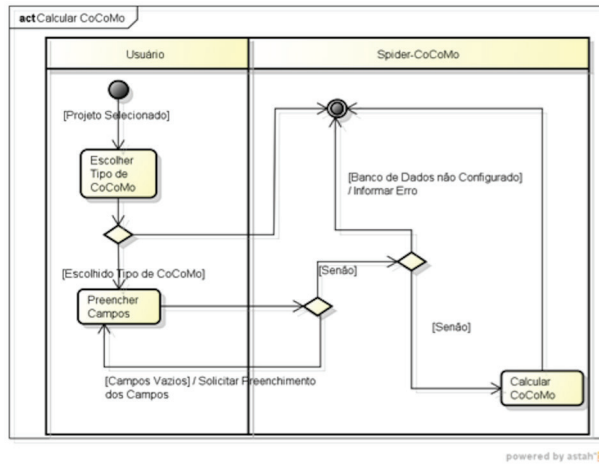


Figura150. Diagrama de Atividades para Realizar um Cálculo



Capítulo 19

Implementação de Critérios Objetivos

em [Barros e Oliveira, 2010a]

Renan de Sales Barros

Sandro Ronaldo Bezerra Oliveira

A Qualidade de Software é uma área de conhecimento da Engenharia de Software cuja finalidade é garantir que o software atenda a um determinado conjunto de padrões. O termo Qualidade de Software é bastante difundido na Engenharia de Software [IEEE, 2004] e está relacionado com teste de software e com a definição, padronização e normatização do processo de desenvolvimento de software e de seus produtos de trabalho.

Existem diversas técnicas para se realizar a qualidade de software. Algumas dessas técnicas possuem práticas que preveem a inspeção ou auditoria de algum produto de trabalho. Um mecanismo comumente adotado para a realização dessas inspeções e auditorias é a utilização de *checklists* [IEEE, 2004]. O uso de *checklists* é aconselhado uma vez que um *checklist* funciona como um guia para realização das inspeções ou auditorias [Smith, 1997].

Um *checklist* apresenta uma lista de critérios que devem ser avaliados em um dado produto de trabalho. Esses critérios podem ser abrangentes como, por exemplo, "Como foi descrito o ciclo de vida?". Esse tipo de critério está sujeito a interpretação do revisor e essa característica pode resultar em uma inspeção não objetiva. Por outro lado, os critérios podem ser menos

abrangentes como, por exemplo, “Existe uma descrição do ciclo de vida?”. Esse tipo de critério pode receber apenas um conjunto fechado de valores, para o exemplo dado os valores são “sim” e “não”. Esse tipo de critério é definido como critério objetivo. Os critérios objetivos são critérios que possuem uma subjetividade mínima ou inexistente e possibilitam para um revisor uma margem de interpretação mínima.

Modelos de maturidade como CMMI-DEV e MR-MPS-SW, recomendam a realização de avaliações de produtos de trabalhos a partir de critérios objetivos. A Spider-CL é uma ferramenta que foi desenvolvida para apoiar esse tipo de avaliação e possui funcionalidades relacionadas com o gerenciamento, aplicação e registro de *checklists* com critérios objetivos. Assim, as avaliações objetivas oriundas do CMMI-DEV, do MR-MPS-SW, ou de qualquer programa de melhoria podem ser sistematizadas com a Spider-CL. Além disso, a Spider-CL também permite a utilização de *checklists* com critérios objetivos para propósitos diferentes de avaliação de produtos de trabalho de processos de software.

19.1. Motivação

A concepção da Spider-CL deu-se durante as atividades do projeto SPIDER. Esse projeto tem como objetivo o levantamento ou desenvolvimento de ferramentas livres que possibilitem a criação de produtos de trabalho a partir dos resultados esperados descritos nos objetivos dos processos do MPS.BR. Durante esse levantamento, foi detectada a ausência de uma ferramenta gratuita que sistematizasse avaliações objetivas para os programas de melhoria de processos como o CMMI ou o MPS.BR.

A Spider-CL foi então desenvolvida com foco no atendimento de programas de melhoria de processos organizacionais tendo como contexto a concepção, especificação e aplicação de *checklists* que provejam inspeção de produtos de trabalho a partir de critérios objetivos. Nesse contexto,

constatou-se que a ferramenta deveria apresentar algumas características específicas como: **Portabilidade**, a ferramenta deve ser inserida em diferentes ambientes organizacionais, sendo que esses ambientes podem fazer uso de diferentes tipos de tecnologias; **Persistência das Informações**, os dados gerenciados pela ferramenta devem estar disponíveis para auditorias e inspeções; **Disponibilidade** via Web, em algumas organizações, os funcionários têm a possibilidade trabalhar remotamente; **Controle de Acesso**, como a ferramenta pode estar disponível através da intranet da organização ou mesmo na internet, a ferramenta deve possuir um mecanismo de controle de acesso.

Além da ausência de uma ferramenta com as características citadas, a construção da Spider-CL também foi motivada pela crescente utilização das metodologias ágeis. Uma das práticas descritas por esse tipo de metodologia é a redução no número de artefatos gerados no decorrer do processo de software. Com o uso da Spider-CL, as organizações terão todas as informações relacionadas com as avaliações objetivas controladas e integradas na ferramenta o que deve reduzir o número de artefatos gerados mantendo a aderência ao CMMI ou ao MPS.BR. A ferramenta encontra-se disponível em <http://www.spider.ufpa.br>.

19.2. Principais Funcionalidades

A seguir são descritas algumas das principais funcionalidades disponíveis na Spider-CL, organizadas no menu conforme pode ser visto na Figura 151.



Figura151. Tela Inicial da Spider-CL

19.2.1. Controle de Usuários

A Spider-CL apresenta dois perfis de usuários: Administrador e Limitado. Esses perfis definem funcionalidades que um usuário poderá acessar na ferramenta. Além de um perfil, cada usuário apresenta também um *status* que pode ser Ativo ou Inativo. Usuários com *status* Ativo podem acessar as funcionalidades definidas de acordo com o seu perfil. Por outro lado, os usuários com *status* Inativo não podem acessar as funcionalidades disponibilizadas pela ferramenta. A tela de cadastro de usuários pode ser visualizada na Figura 152.

19.2.2. Definição de Critérios

Para que os *checklists* possam ser elaborados, é necessário antes definir os critérios que estarão presentes nos *checklists*. Os critérios devem apresentar no mínimo duas alternativas de resposta, como pode ser visto na Figura 153. Depois de criados, os critérios podem ser consultados e podem ter os textos das alternativas de respostas e do critério alterados.



Figura152. Cadastro de Usuário na Spider-CL

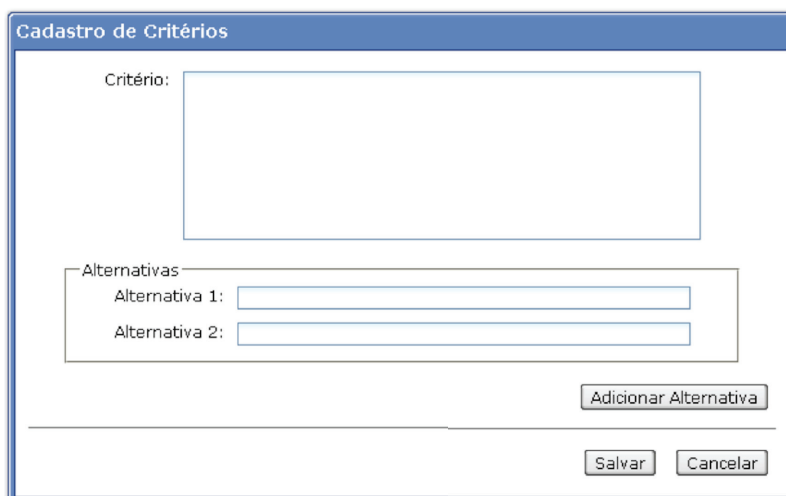


Figura153. Cadastro de Critério na Spider-CL

19.2.3. Elaboração de *Checklists*

Os *checklists* são elaborados a partir da organização de um conjunto de critérios. Isso permite que os critérios possam ser reutilizados por diferentes *checklists*. Durante a elaboração de um *checklist* também é possível determinar um conjunto de usuários com perfil Limitado que poderão aplicar o *checklist*. Essa lista de usuários pode ser modificada livremente após a criação do *checklist*, como pode ser visto na Figura 154.

Cadastro de Checklists

Dados do checklist:

Nome do checklist:

Descrição:

Lista de critérios do checklist:

Código do critério	Critério
--------------------	----------

Pesquisar critérios

Pesquisar em:

Palavra-chave:

Código do critério	Critério
--------------------	----------

Lista de usuário que podem acessar o checklist:

Código do usuário	Nome completo	Nome de usuário	Status
-------------------	---------------	-----------------	--------

Pesquisar usuários

Pesquisar em:

Palavra-chave:

Figura154. Elaboração de *Checklist* na Spider-CL

19.2.4. Aplicação de *Checklists*

Após a elaboração de um *checklist* o mesmo estará disponível para ser aplicado, como pode ser visto na Figura 155. O processo de aplicação consiste em atribuir valores aos critérios que o *checklist* apresenta. Para cada um desses critérios é apresentado um campo onde observações e informações adicionais referentes ao critério podem ser relatadas. Além disso, também é possível relatar observações e informações adicionais a respeito da aplicação do *checklist* em um campo disponível. No momento da conclusão de uma aplicação de *checklist*, além dos valores atribuídos aos critérios, também são armazenados pela Spider-CL: o horário e data da aplicação; e o usuário que realizou a aplicação.

A interface, intitulada "Preenchimento de Checklist", é organizada em seções. No topo, há um campo "Checklist 1" e uma seção "Descrição do checklist:" com um campo de texto. Abaixo, a seção "Descrição do uso do checklist:" também possui um campo de texto. O formulário contém duas seções de critérios, cada uma com um cabeçalho em azul. A primeira seção, "Critério 1:", mostra "Critério 1" com opções de resposta "☐ Sim" e "☐ Não", seguidas por um campo "Descrição:". A segunda seção, "Critério 2:", mostra "Critério 2" com as mesmas opções de resposta e um campo "Descrição:".

Figura155. Preenchimento de *Checklist* na Spider-CL

19.2.5. Visualização dos Resultados das Aplicações de *Checklists*

Após a aplicação de um *checklist*, os dados coletados nessa aplicação podem ser visualizados e exportados para um arquivo no formato PDF, como pode ser visto na Figura 156. Além disso, também é possível visualizar esses dados através de gráficos.

A interface de visualização de informações de uma aplicação de checklist na Spider-CL é apresentada em uma janela com o título "Visualização das Informações da Aplicação de um Checklist". A interface é dividida em duas seções principais: "Dados do Uso do Checklist" e "Respostas dos Critérios do Checklist".

Dados do Uso do Checklist:

- Código:** 3
- Checklist:**
 - Código:** 1
 - Título:** Checklist 1
- Usuário:**
 - Código:** 1
 - Nome:** admin
- Data:** 2010-05-20 11:34:31.0
- Descrição do uso checklist:** (área vazia para texto)

Respostas dos Critérios do Checklist:

- Critério 1:**
 - Critério 1
 - Resposta:** Não
- Critério 2:**
 - Critério 2
 - Resposta:** Não

Na base da janela, há três botões: "Excluir", "Baixar PDF" e "Cancelar".

Figura156. Visualização da Aplicação de *Checklist* na Spider-CL

Os gráficos são gerados com relação a um critério ou com relação a um *checklist*. Quando referente a um critério, o gráfico mostra a quantidade ou o percentual que uma alternativa foi marcada como resposta considerando

todos os *checklists* onde esse critério é utilizado. Por outro lado, quando referente a um *checklist*, para cada critério do *checklist* é exibido um gráfico mostrando a quantidade ou o percentual que uma alternativa foi marcada como resposta considerando apenas o *checklist* em questão. Cada gráfico exibido na ferramenta pode ser apresentado de duas formas, como um gráfico de barras ou de pizza ficando a escolha a critério do usuário, como pode ser visto na Figura 157.

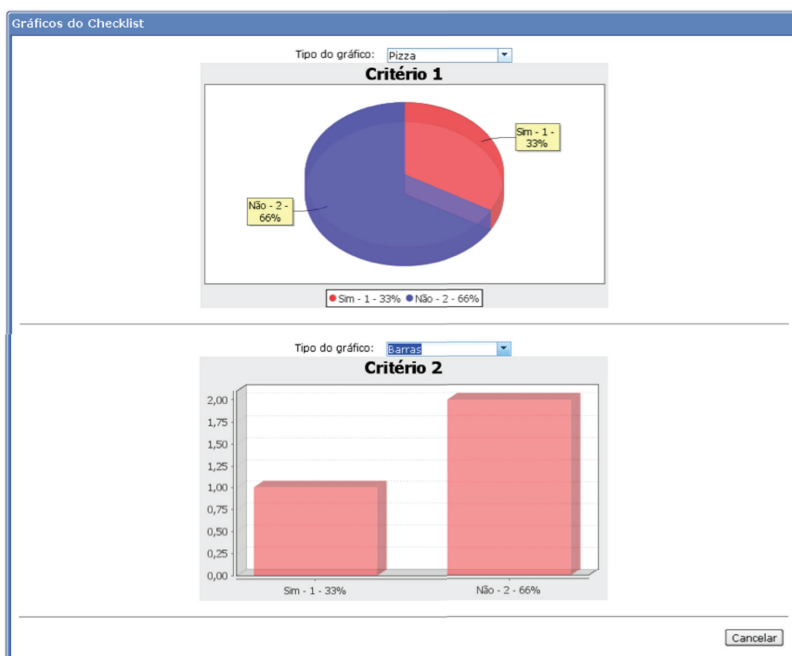


Figura157. Geração de Gráficos na Spider-CL



Capítulo 20

Implementação de Simulação para o Ensino de Melhoria de Processo de Software

em [Chaves et al., 2011a], [Chaves et al., 2010], [Chaves et al., 2011b]

Rafael Oliveira Chaves

Tales C. Miranda

Emanuel M. da C. Tavares

Walter A. da L. Lobato

Sandro Ronaldo Bezerra Oliveira

Elói Luiz Favero

As metodologias ágeis possuem métodos, práticas e técnicas que podem aumentar a satisfação do cliente [Boehm e Turner, 2003], além de produzir um sistema com maior qualidade e em menor tempo [Anderson, 2003]. Elas também podem prover maior comunicação entre os membros da equipe, melhorando o processo de desenvolvimento como um todo, possibilitando que as respostas aos requisitos e às mudanças sejam mais rápidas.

A modelagem de processos é uma atividade chave para a melhoria do processo de software, auxiliando na identificação de deficiências e estimativas de sua execução [Becker-Kornstaedt, 2000]. Entre os seus principais objetivos estão ajudar a conhecer, controlar e gerenciar as atividades de um processo de software [Acuña e Juristo, 2004]. Dentro dessa perspectiva, Rai e Al-Hindi (2000), Krishnan *et al.* (2000) e Armenise *et al.* (1993) sugerem ainda que existe uma relação positiva entre a Modelagem de Processos de Software (MPS) e

a maior qualidade do produto e do processo de software. A partir disso, a MPS tem sido associada com melhores produtos de software e com a qualidade do processo de software, ambos estes que dependem consideravelmente da maturidade do processo de software [Garcia-Borgoñon *et al.*, 2013].

Apesar da importância do conhecimento da MPS, foi encontrado que os profissionais têm aprendido mais sobre processos de software em suas atividades de trabalho do que nos seus cursos universitários/educação [Lethbridge, 2000]. Isso pode ser porque, no momento, a MPS geralmente representa apenas uma pequena parte do que é ensinado nos cursos de graduação em computação [ACM/AIS/IEEE-CS, 2005], ou porque a maioria dos currículos universitários considera processo de software como sendo um conhecimento marginal da engenharia de software (ES) [Kuhrmann *et al.*, 2013].

O Curriculum IEEE-CS/ACM (2004) recomenda que a área de conhecimento sobre processo de software deve ser uma parte do ensino de graduações em ES. Um dos objetivos de aprendizagem nesta área do conhecimento é a modelagem e especificação de processos de software.

Objetivos de aprendizagem podem ser alcançados através da adoção de estratégias de ensino [Saskatchewan Education, 1991] que definem a abordagem de ensino que deve ser usada. As estratégias educacionais podem ser classificadas em 5 categorias: instrução direta, instrução interativa, instrução indireta, estudo independente e aprendizagem experiencial [Saskatchewan Education, 1991]. No contexto de aprendizagem experiencial, têm-se os jogos de computador em que os alunos aprendem fazendo ou experimentando situações reais ou simuladas [Itin, 1999]. Tais jogos podem ser mais uma opção para o ensino de MPS.

A fim de se ter um panorama sobre o ensino de modelagem de processos de software no ambiente universitário, analisou-se as disciplinas [Jaccheri, 1997; Groth e Robertson, 2001; Hawker, 2009; Wangenheim e Hauck, 2010] em relação às estratégias instrucionais que elas usam para o ensino de MPS. Como resultado, percebeu-se que a instrução direta, por meio de aulas expositivas, é usada para a apresentação de conceitos iniciais sobre MPS. Todas

as disciplinas têm como principal estratégia a instrução indireta, por meio de projetos acadêmicos. Outras estratégias instrucionais com menor uso nas disciplinas são: aprendizagem experiencial, por meio de simulação, mas sem apoio computadorizado; e instrução interativa por meio de aprendizagem em pares. Não se observou a utilização de jogos como recurso de uma estratégia instrucional para ensino de MPS.

Com o intuito de desenvolver um jogo específico para ensinar MPS é necessário evidenciar as vantagens de jogos sérios em face as estratégias de ensino tipicamente usados para o ensino MPS, ou seja, a instrução direta e instrução indireta. A instrução direta, por meio de aulas expositivas, é centrada no professor, o estudante permanece passivo na maior parte do tempo, e o conteúdo tende a ser rapidamente esquecido [Saskatchewan Education, 1991]. Por outro lado, os jogos podem diminuir a carga do professor, permitir uma maior autonomia do aluno e, geralmente, a atenção do aluno é mantida por um tempo mais longo. Em relação à instrução indireta, o projeto acadêmico é uma abordagem que tenta imitar projetos reais em um ambiente acadêmico. Estes projetos se destinam a iniciar os alunos na prática da ES e, também, podem se concentrar em habilidades não técnicas, tais como comunicação, trabalho em grupo, habilidades interpessoais, gestão de projetos, e resolução de problemas [Goold e Horan, 2002]. No entanto, o contexto acadêmico muitas vezes impõe restrições de tempo e escopo para esses projetos [Navarro, 2006]. Jogos podem permitir flexibilidade na criação de diferentes escopos e cenários de ES, e as durações das partidas são mais curtas, portanto jogos podem ser facilmente utilizados durante as aulas.

Um jogo para apoiar o ensino da MPS deve superar as limitações da instrução direta e instrução indireta, e também deve tirar proveito dos benefícios que são típicos de jogos (aprendizagem experiencial). Neste contexto, e em virtude da falta de jogos específicos para o ensino MPS, um jogo chamado DesigMPS foi desenvolvido para este fim, baseando-se em trabalhos sobre avaliação adaptativa usando Mapas Conceituais (MC), e com as seguintes características: permitir vários escopos (flexíveis), pois novos

modelos de processo de software podem ser criado por um especialista em MPS; e o tempo de jogo ajustado para durar cerca de uma aula típica das universidades brasileiras.

Embora haja pouca evidência de que jogos sérios são eficazes no ensino ES quando comparado com outros métodos de ensino [Wangenheim e F. Shull, 2009], existem algumas vantagens significativas para uso destes jogos sérios:

- Podem tornar o aprendizado mais divertido [Kafai, 2001], por permitir o “aprender fazendo” em situações realistas, produzindo resultados positivos em termos de aquisição de conhecimento e compreensão dos conteúdos [Connolly *et al.*, 2012];
- Possibilitam os alunos trabalhar em seu próprio ritmo, requerendo pouca ou nenhuma presença de um instrutor, nem interação com seus pares [Wangenheim *et al.*, 2009];
- Oferecem atividades que estão em conformidade com modernas teorias de aprendizagem, estas que sugerem que a aprendizagem é mais eficaz quando é empírica, baseada em problemas, e fornece um *feedback* imediato [Boyle *et al.*, 2011].

20.1. O Jogo DesigMPS

DesigMPS é um jogo de computador que visa apoiar o ensino da MPS, por meio do reforço de conceitos pertinentes e do exercício de Modelagem do Processo de Software. Em termos de objetivos de aprendizagem, o jogo foca os níveis cognitivos de conhecimento, compreensão e aplicação de acordo com a taxonomia revisada de objetivos educacionais de Bloom [Anderson e Krathwohl, 2001].

Ao final da aplicação do jogo espera-se que os alunos tenham maior capacidade de: 1) nomear os elementos de processo de software (atividade, papel,

artefato e ferramenta), lembrar seus respectivos usos, assim como lembrar as regras de modelagem – nível de conhecimento; 2) reconhecer e descrever os elementos do processo de software e seus relacionamentos de dependência (e.g. *start_to_start*, *end_to_start*), a partir de uma descrição textual do processo - nível de compreensão e; 3) modelar um processo de software usando a SPIDER_ML em um contexto de melhoria de processos do MPS.BR – nível de aplicação.

O jogo tem como público alvo estudantes de cursos de graduação ou profissionais já atuantes no mercado. O jogo foi projetado para ser jogado individualmente durante aulas presenciais com duração máxima de 90 minutos.

As seguintes estratégias foram usadas no DesigMPS: 1) o uso de Linguagem de Modelagem de Processo de Software gráfica, SPIDER_ML, como ferramenta de apoio à modelagem; 2) a usabilidade é centrada no aluno, não no especialista em modelagem; e 3) não sobrecarrega a atividade de modelagem com variáveis de simulação.

20.1.1. *Game Flow*

No jogo o aluno assume o papel de um engenheiro de processos que deve modelar um processo de software com base no modelo MR-MPS-SW, usando a SPIDER_ML. Para tal, o DesigMPS conduz o aluno em atividades de modelagem de processos, sendo estruturado em quatro estágios em ordem crescente de dificuldade: A, B, C e D (vide Figuras 158 a 161).

O DesigMPS é baseado nos trabalhos de [Anohina *et al.*, 2007; Anohina *et al.*, 2010] sobre avaliação adaptativa de conhecimento usando Mapas Conceituais (MC), esta avaliação ocorre por meio de 6 tarefas em nível crescente de dificuldade. Os estágios A, B, C e D do DesigMPS são equivalentes às tarefas no. 1, no. 2, no. 4 e no. 6 propostas por Anohina *et al.* (2010). As tarefas

no. 3 e no. 5 não foram implementadas porque elas eram muito similares as tarefas no. 2 e no. 6, respectivamente.

Para que o aluno possa completar os estágios, o DesigMPS fornece as seguintes informações apresentadas na Figura 158: a) uma descrição textual de um processo de software do MR-MPS-SW; b) a estrutura do modelo do processo, esta é a configuração de como os elementos estão relacionados, por meio de associações e transições; e c) uma lista com os elementos e relacionamentos do processo.

O aluno deve modelar o processo colocando os elementos e relacionamentos da lista (vide Figura 158(c)) em suas posições corretas na estrutura do modelo de processo (vide Figura 158(b)). Os elementos e relacionamento da lista são extraídos de um modelo de gabarito (modelo de referência), criado por um especialista em MPS e baseado no MR-MPS-SW.

Nos estágios mais fáceis é fornecido ao aluno a estrutura do modelo de processo com uma maior quantidade de elementos preenchidos e relacionamentos rotulados nas suas posições corretas, esta é a abordagem *fill-in-the-map* (estágios A, B e C). À medida que os estágios vão se tornando mais difíceis, é fornecida uma quantidade cada vez menor de elementos e relacionamentos preenchidos em suas posições na estrutura. No estágio mais difícil, os alunos precisam definir o modelo de processo por eles mesmos, do princípio, sem o auxílio da estrutura do modelo do processo, esta é a abordagem *construct-the-map* (estágio D).

A Figura 158 mostra o mais fácil, o estágio A: é fornecida a estrutura do modelo do processo com todos os relacionamentos rotulados (*END_TO_START*, *accepted*, *refused*) e alguns elementos preenchidos nas suas posições corretas. Os elementos que devem ser preenchidos pelos alunos estão marcados com “???????”. Nesse exemplo os elementos preenchidos são: Design Doc, Code e Tests.

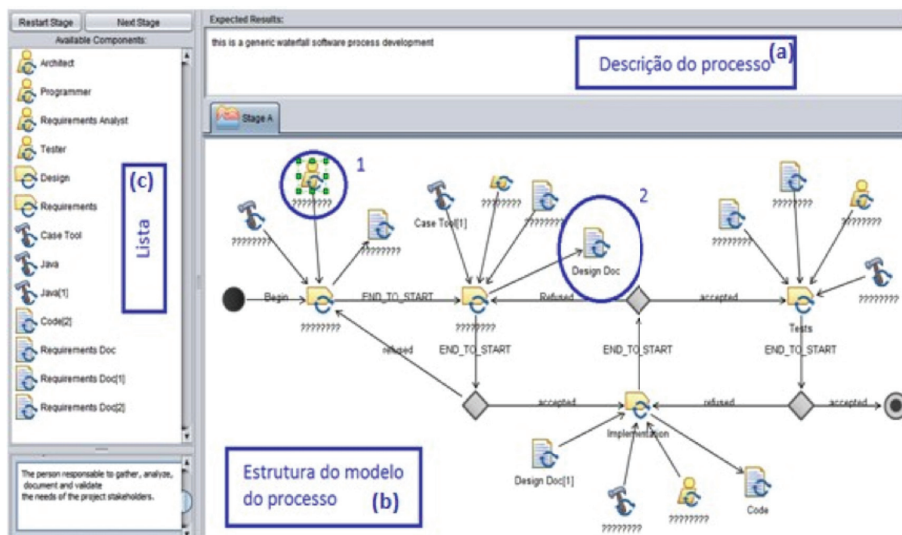


Figura 158. Estrutura do modelo do estágio A apresentando 4 elementos preenchidos e todos os relacionamentos rotulados.

A Figura 159 mostra o estágio B: é fornecida a estrutura do modelo do processo com todos os relacionamentos rotulados corretamente, porém todos os elementos estão marcados com “????????”, requerendo o preenchimento por parte do aluno. A Figura 160 mostra o estágio C: em que é fornecida a estrutura do modelo do processo, no entanto, não é disponibilizada nenhuma informação acerca dos elementos e relacionamentos, estes devem ser preenchidos pelo aluno, tanto relacionamentos e elementos estão marcados com “????????”. Por fim, a Figura 161 mostra o estágio D, o mais difícil: nele a estrutura do processo não é fornecida, portanto os, alunos devem criar o modelo do principio, sem nenhuma informação a priori sobre a estrutura do processo.

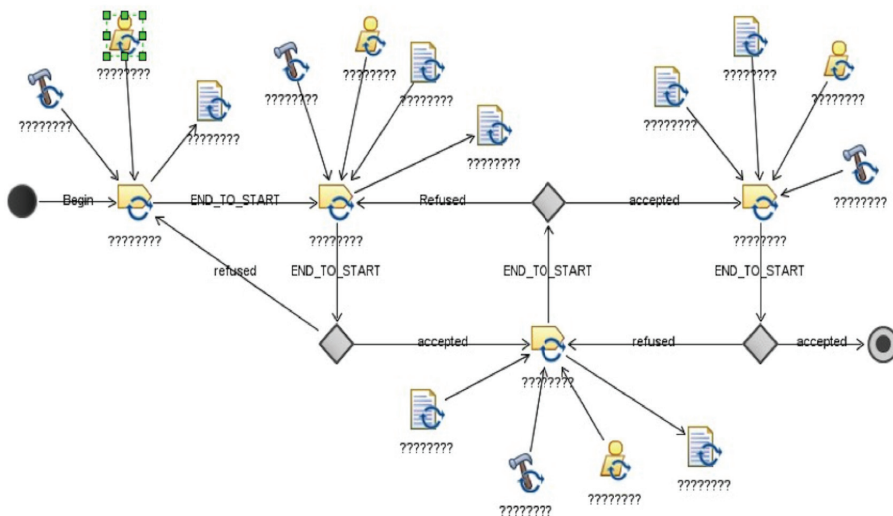


Figura 159. Estágio B: apresentando todos os elementos marcados como “?????” e os relacionamentos rotulados.

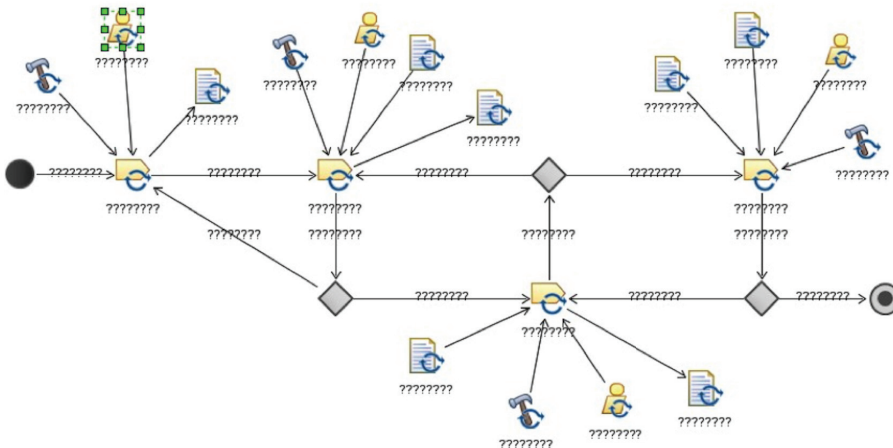


Figura 160. Estágio C: nenhum elemento preenchido ou relacionamento rotulado estão disponíveis na estrutura.

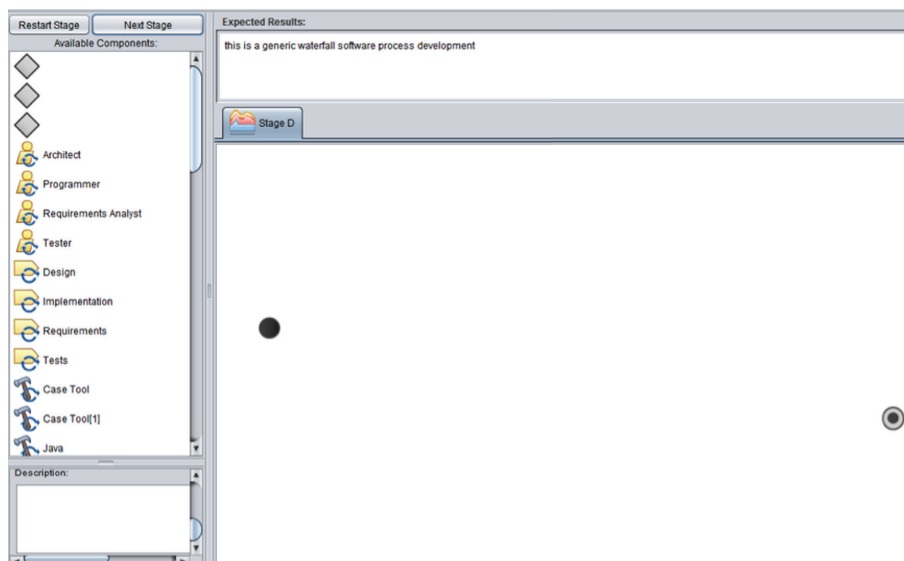


Figura 161. Estágio D: nenhuma parte da estrutura está disponível.

A comparação entre o modelo do aluno e o modelo de gabarito (modelo de referência) é realizada ao final de cada estágio, de acordo com critérios de similaridades (ver Tabela 20), gerando uma pontuação de forma automática. A pontuação – diferenças nos critérios de similaridade entre os modelos do aluno e de referência –, juntamente com a indicação dos erros constituem o *feedback* aos alunos (vide Figura 162). Se a pontuação for maior do que um limiar mínimo o aluno passa para o próximo estágio, caso contrário permanece no mesmo.

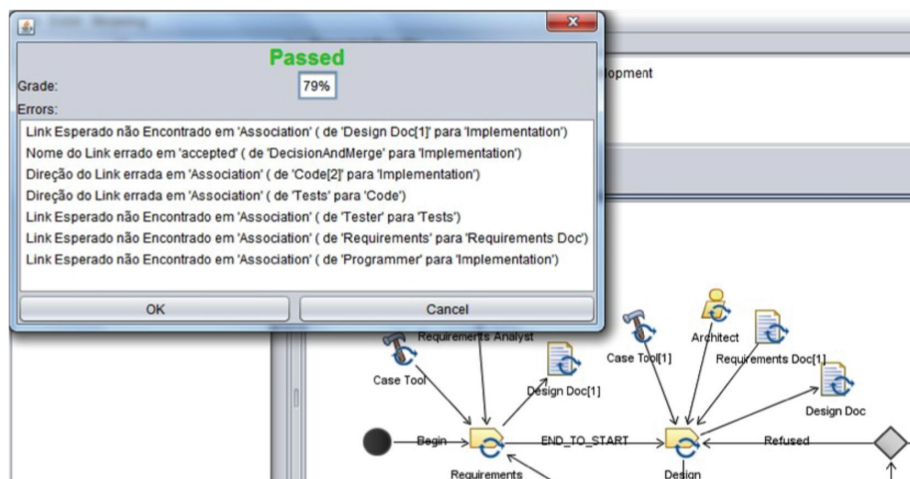


Figura 162. Pontuação e a indicação dos erros como *feedback* aos alunos no final de cada estágio

A Tabela 20 sumariza os estágios e as respectivas informações disponíveis para os alunos.

Tabela 20. Sumário dos componentes dos estágios

Estrutura do Modelo de Processo			
Estágio	Vazio	Com alguns elementos preenchidos	Com todos os relacionamentos rotulados
A		X	X
B			X
C	X		
D			

Para os estágios A, B e C, o DesigMPS fornece uma ajuda opcional para os alunos: quando clicarem em alguma posição na estrutura do processo em que um elemento não está preenchido, ao mesmo tempo, o *frame* inferior esquerdo mostra a descrição do elemento. Esta descrição é uma breve

declaração sobre a função de elemento no processo de software, como mostrado na Figura 163.

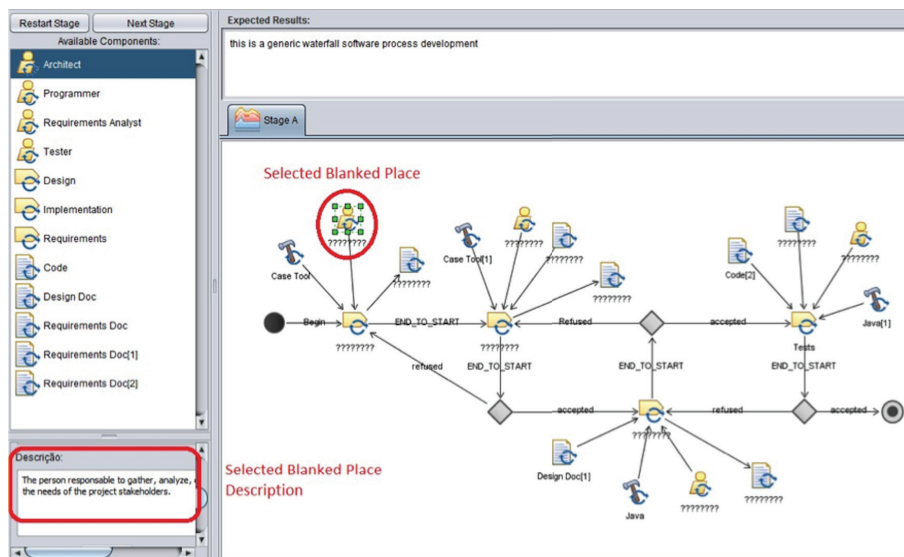


Figura 163. Seleção de uma posição não preenchida e sua descrição

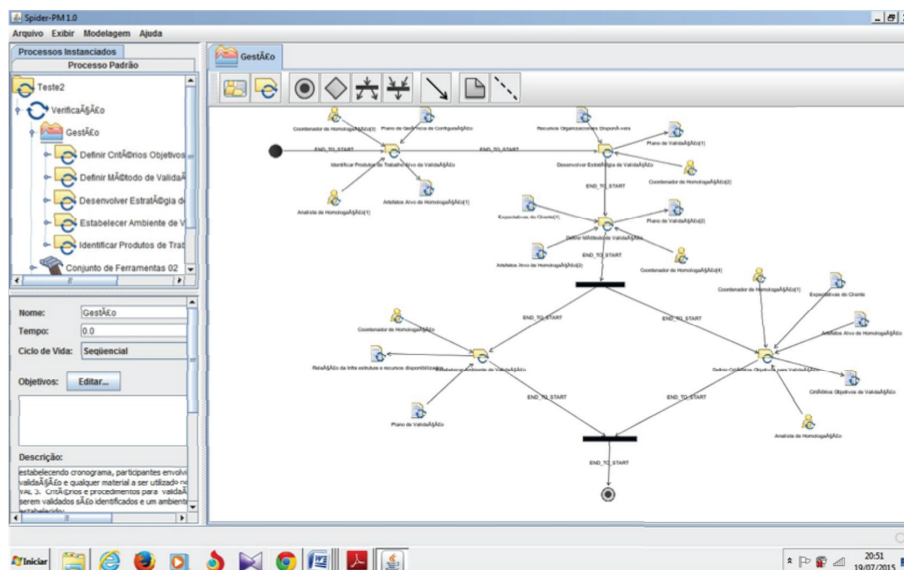


Figura 164. Primeiro modelo de referência usado na atividade de aprendizagem para o grupo experimental

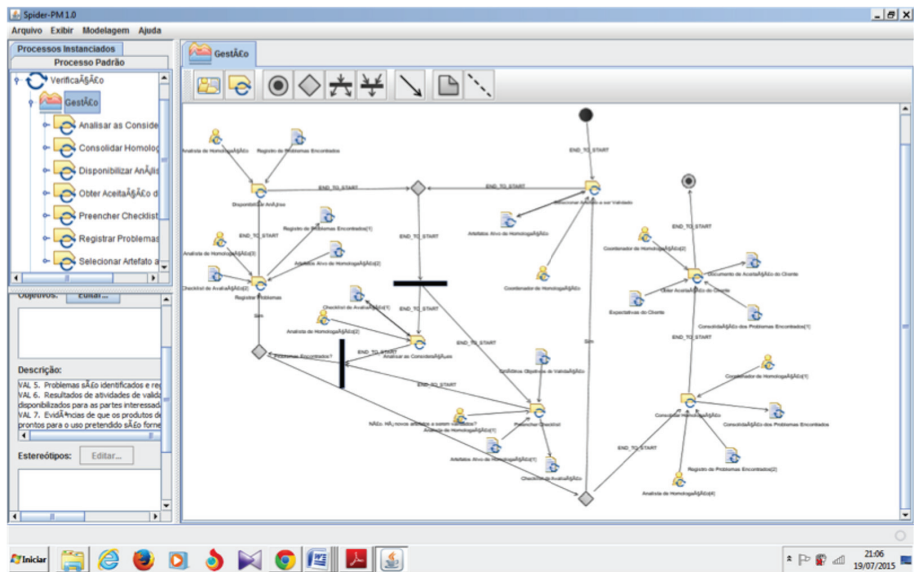


Figura 165. Segundo modelo de referência usado na atividade de aprendizagem para o grupo experimental



Capítulo 21

Avaliação de Processos de Software

em [Neiva et al., 2011]

Jñane Neiva Sampaio de Souza

Pedro Afonso Aviz

Sandro Ronaldo Bezerra Oliveira

É visível que, de forma rápida e constante, as empresas estão aumentando sua dependência tecnológica e isso significa que suas operações internas estão sendo conduzidas e direcionadas por um conjunto cada vez maior de sistemas informatizados [Bartié, 2002]. Ao mesmo tempo, no ambiente de alta tecnologia do século XXI, quase todas as organizações estão envolvidas no desenvolvimento de produtos e serviços cada vez mais complexos [SEI, 2010]. Tendo em vista atender bem a essa demanda e para se manter competitiva, é crescente a preocupação das organizações desenvolvedoras de software com a qualidade de seus produtos e serviços prestados.

Nesse contexto, há uma tendência na indústria de software para a implementação de modelos e normas para a melhoria da qualidade dos produtos e dos processos, com a pretensão de elevar a maturidade no desenvolvimento de software e, consequentemente, alcançar maior credibilidade e visibilidade no mercado [Staples *et al.*, 2007]. Além disso, na atualidade, são recorrentes as iniciativas de melhoria de processo com a adoção não de apenas um, mas de múltiplos modelos, a depender dos objetivos de negócio da organização, como constatado no trabalho de Mello (2011).

Para medir a capacidade dos processos de software de uma empresa, deve ser realizada a avaliação dos seus processos para diagnosticar a situação em que se encontra e estabelecer o quanto e onde precisa melhorar [ISO/IEC, 2004a]. As avaliações são realizadas comparando-se os processos implementados em uma organização com as boas práticas definidas em um modelo de referência. Para tanto, deve-se utilizar um método de avaliação formal, o qual define as regras e passos para a execução de uma avaliação [ISO/IEC, 2004a].

Geralmente a avaliação de processos de software é feita em conformidade com apenas um modelo/norma por vez, ou seja, para saber se uma organização está aderente a mais de um modelo ou norma, é necessário realizar um processo de avaliação completo para cada modelo/norma, repetindo-se a coleta e a análise das mesmas informações relativas aos processos de software da organização. Isso pode despendar mais tempo, recursos humanos e financeiros [Neiva e Oliveira, 2010]. Em decorrência desses problemas, quando se tratar de ambientes de melhoria multi-modelos, pode ser mais viável a realização de avaliações integradas.

De acordo com Neiva e Oliveira (2010), existem algumas metodologias e ferramentas que executam a avaliação de processos de software com base em um método de avaliação específico de um modelo e, ao final, geram resultados da aderência em relação a mais de um modelo de referência. Contudo, ainda não há uma metodologia que possibilite executar uma avaliação alinhada a mais de um método de avaliação de diferentes modelos de referência em um mesmo processo avaliativo, de modo que seja evidenciado o mapeamento entre as atividades de avaliação inerentes a cada método de avaliação.

Diante do exposto, o objetivo deste trabalho é a definição de uma abordagem de avaliação de processos de software que seja alinhada aos métodos de avaliação MA-MPS – Método de Avaliação para Melhoria de Processo de Software [SOFTEX, 2015] e SCAMPI A – *Standard CMMI Appraisal Method for Process Improvement* [SEI, 2011], de maneira que na condução de uma avaliação integrada possa ser observada e compreendida a correspondência

existente entre as atividades de avaliação dos métodos envolvidos. Essa abordagem serviu de base para a especificação e desenvolvimento de uma ferramenta de apoio à avaliação de processos de software que considera ambos os métodos de avaliação de forma integrada – até então uma proposta não formal, uma vez que os organismos responsáveis pelos métodos citados necessitam avaliar previamente a completude e a corretude do fluxo sistematizado pela ferramenta.

21.1. A Ferramenta Spider-Appraisal

A ferramenta Spider-Appraisal é uma ferramenta de licença GPL voltada para apoiar a avaliação de processos de software em duas abordagens: simples, seguindo-se apenas um método de avaliação, ou integrada (conjunta). Na abordagem integrada, seu principal objetivo é possibilitar que em uma mesma avaliação dos processos de software de uma organização possa ser verificada a aderência ao MR-MPS-SW e CMMI-DEV, através da realização conjunta de atividades constantes em ambos os métodos de avaliação: MA-MPS e SCAMPI A. A ferramenta pode ser utilizada em diferentes organizações, independentemente do tamanho da mesma, e encontra-se disponível em http://www.spider.ufpa.br/projetos/spider_appraisal/Spider-Appraisal.zip.

Haja vista o papel fundamental da avaliação de processos de software, pode-se destacar que a utilização de ferramenta para apoiar o processo de avaliação é de grande importância, como ratificado por Hunter, Robinson e Woodman (1997):

Ferramentas de [...] avaliação de software são, portanto, de valor tanto para os produtores de software que desejam melhorar seus processos em comparação com seus concorrentes, como para compradores de software que desejam avaliar os processos de potenciais contratantes [...].

Atualmente, a maioria das avaliações de processos de software é realizada apenas registrando-se as informações em documentos de texto e em planilhas eletrônicas. Isso torna o trabalho bem manual, com mais custo de tempo e com maior dificuldade para manipular as informações da avaliação. Como já discutido, a realização de uma avaliação conjunta demanda maior esforço da equipe de avaliação e acrescenta mais complexidade na análise das evidências coletadas e na geração dos resultados da avaliação.

Dentro deste contexto, a Spider-Appraisal visa a auxiliar as organizações interessadas na avaliação de processos dos modelos de referência MR-MPS-SW e CMMI-DEV, provendo um acompanhamento sistematizado de todas as fases do processo de avaliação. Apesar do apoio sistêmico, é imprescindível enfatizar que a ferramenta não tende a eliminar o trabalho intelectual do avaliador necessário em uma avaliação, por conta da subjetividade existente, mas sim dar suporte à análise e ao registro dos resultados obtidos neste processo.

21.2. As Principais Funcionalidades da Spider-Appraisal

A Spider-Appraisal possui as seguintes características:

- Cadastro das Estruturas dos Modelos de Referência: permite cadastrar e manter as estruturas de diferentes versões para os modelos de referência MR-MPS-SW e CMMI-DEV;
- Cadastro do Mapeamento dos Modelos: permite cadastrar e manter o mapeamento existente entre as estruturas dos modelos MR-MPS-SW e do CMMI-DEV, de forma a possibilitar a avaliação integrada (conjunta) desses modelos;
- Atribuição de Pontuação: possibilita atribuir pontuação para os ativos dos modelos com base na análise das evidências da avaliação. Quando se tratar de avaliação integrada, permite cadastrar a

pontuação para ambos os modelos de referência a partir do mapeamento cadastrado entre os ativos dos modelos;

- **Visualização do Mapeamento dos Métodos de Avaliação:** em se tratando de avaliações integradas, para cada atividade da ferramenta que possua correspondência com o framework, possibilita visualizar o mapeamento existente entre as tarefas do MA-MPS com as atividades do SCAMPI A. Isso auxilia na condução de avaliações integradas, em que os processos de ambos os métodos de avaliação devem ser considerados.

A Spider-Appraisal divide-se nas seguintes áreas funcionais: a de **Administração**, que compreende a gerência dos cadastros básicos; e em outra com funcionalidades concernentes à **Execução** da Avaliação. Relativo ao módulo de **Administração**, destacam-se as seguintes funcionalidades: gerenciar as estruturas dos modelos de referência MR-MPS-SW e CMMI-DEV; gerenciar o mapeamento entre os resultados e práticas; e gerenciar o mapeamento entre os resultados de atributo de processo e as práticas genéricas. O módulo de **Execução** da avaliação provê funcionalidades de: planejamento da avaliação; cadastro de evidências objetivas; pontuação das práticas/resultados esperados; atribuição da maturidade e capacidade organizacional; emissão de relatórios; dentre. Nas seções seguintes são apresentadas algumas das principais funcionalidades da Spider-Appraisal.

21.2.1. Uma Visão Geral da Ferramenta

Para utilizar a ferramenta, o usuário necessita de autenticação por meio de *login* e senha, como ilustrado na Figura 166. De acordo com o perfil do usuário, ele terá permissão de acesso a determinadas funcionalidades e atividades que poderá executar.



Figura 166. Tela de Autenticação do Usuário

A Figura 167 apresenta a tela inicial da ferramenta Spider-Appraisal, possuindo: a) um menu superior, que agrupa funções de configurações da ferramenta, de acesso aos usuários, e geração de relatórios; e b) um menu lateral, que permite a navegação entres as funcionalidades dos módulos de administração e de execução.

A ferramenta também implementa as restrições definidas no fluxo de avaliação de processos, onde cada atividade tem que ser executada obedecendo a uma ordem de dependência, bem como as fases propostas no fluxo.

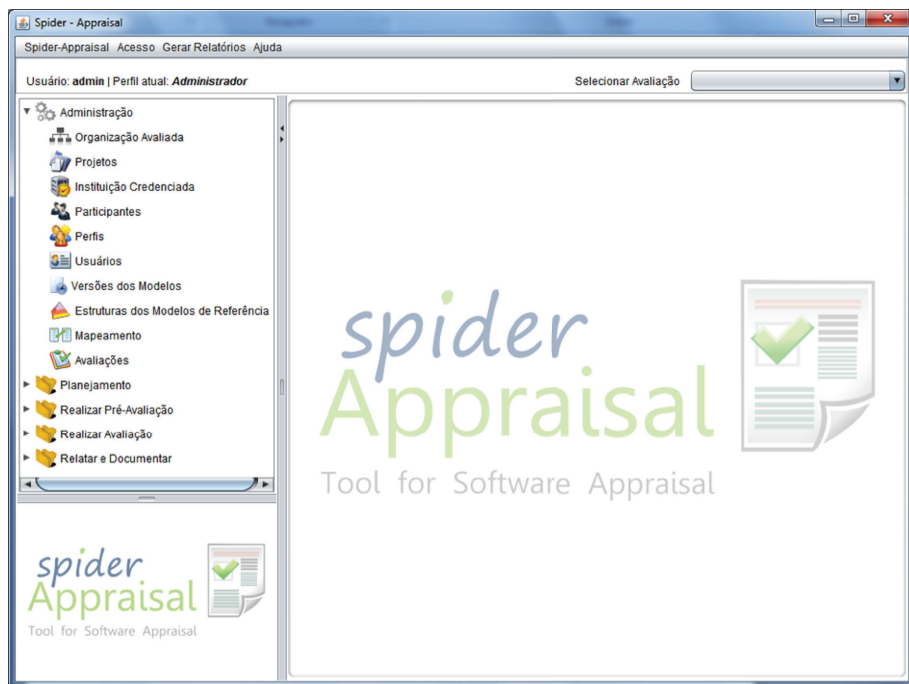


Figura 167. Tela Principal da Spider-Appraisal

21.2.2. Módulo de Administração

Antes de dar início à execução de uma avaliação, é necessário realizar alguns cadastros básicos que estão contidos na área de administração do menu lateral.

No cadastro da Organização Avaliada são inseridas informações que caracterizam a organização que será avaliada segundo um modelo de referência de processos, como mostra a Figura 168, na aba Organização.

Spider - Appraisal

Spider-Appraisal Acesso Gerar Relatórios Ajuda

Usuário: admin | Perfil atual: Administrador Selecionar Avaliação

Administração

- Organização Avaliada
 - Projetos
 - Instituição Credenciada
 - Participantes
 - Perfis
 - Usuários
 - Versões dos Modelos
 - Estruturas dos Modelos de Referência
 - Mapeamento
 - Avaliações
 - Planejamento
 - Realizar Pré-Avaliação
 - Realizar Avaliação
 - Relatar e Documentar

spider Appraisal
Tool for Software Appraisal

Organização Unidade Organizacional

Nome CNPJ

Site Ano de Fundação

Endereço

País Estado Cidade CEP

Nome do Contato E-mail

Telefone Fax

Informações Adicionais

Salvar Cancelar

Nome	Endereço	CNPJ	Telefone
Organização 2	Av. João Paulo II, S/N.	00.000.000/0000-00	() - /
Organização 1	Tv. Endereço, 1000.	00.000.000/0000-00	() - /

Editar Excluir

Figura 168. Tela Organização Avaliada - Aba Organização

O cadastro da Unidade Organizacional permite o registro de informações concernentes ao setor específico de desenvolvimento de software que será avaliado, com a indicação da Organização a qual pertence, conforme a Figura 169, na aba Unidade Organizacional. Podem ser cadastradas mais de uma unidade organizacional para uma mesma organização avaliada.

The screenshot displays the Spider-Appraisal software interface. The title bar reads 'Spider - Appraisal'. The menu bar includes 'Spider-Appraisal', 'Acesso', 'Gerar Relatórios', and 'Ajuda'. The status bar shows 'Usuário: admin | Perfil atual: Administrador' and a 'Selecionar Avaliação' dropdown.

The left sidebar contains a tree view under 'Administração' with the following items: 'Organização Avaliada' (selected), 'Projetos', 'Instituição Credenciada', 'Participantes', 'Perfis', 'Usuários', 'Versões dos Modelos', 'Estruturas dos Modelos de Referência', 'Mapeamento', 'Avaliações', 'Planejamento', 'Realizar Pré-Avaliação', 'Realizar Avaliação', and 'Relatar e Documentar'. At the bottom of the sidebar is the 'spider Appraisal' logo with the tagline 'Tool for Software Appraisal'.

The main window has two tabs: 'Organização' and 'Unidade Organizacional', with the latter being active. The form contains the following fields:

- 'Organização' dropdown menu
- 'Nome' text field
- 'Endereço' text field
- 'País', 'Estado', 'Cidade', and 'CEP' text fields
- 'Coordenador Local' and 'E-mail' text fields
- 'Telefone' and 'Fax' text fields with area codes and dashes
- 'Nº de Colaboradores' text field
- 'Descrição das Atividades' text area

 At the bottom right of the form are 'Salvar' and 'Cancelar' buttons. Below the form is a table with two columns: 'Unidade Organizacional' and 'Organização Avaliada'. It contains one row: 'Unidade Organizacional 1' and 'Organização 1'. At the bottom right of the table are 'Editar' and 'Excluir' buttons.

Figura 169. Tela Organização Avaliada - Aba Unidade Organizacional

No cadastro de Projetos são registrados os dados dos projetos de software que serão analisados pelos avaliadores em busca de evidências objetivas que comprovem a implementação dos resultados/práticas dos modelos de referência, de acordo com a Figura 170.

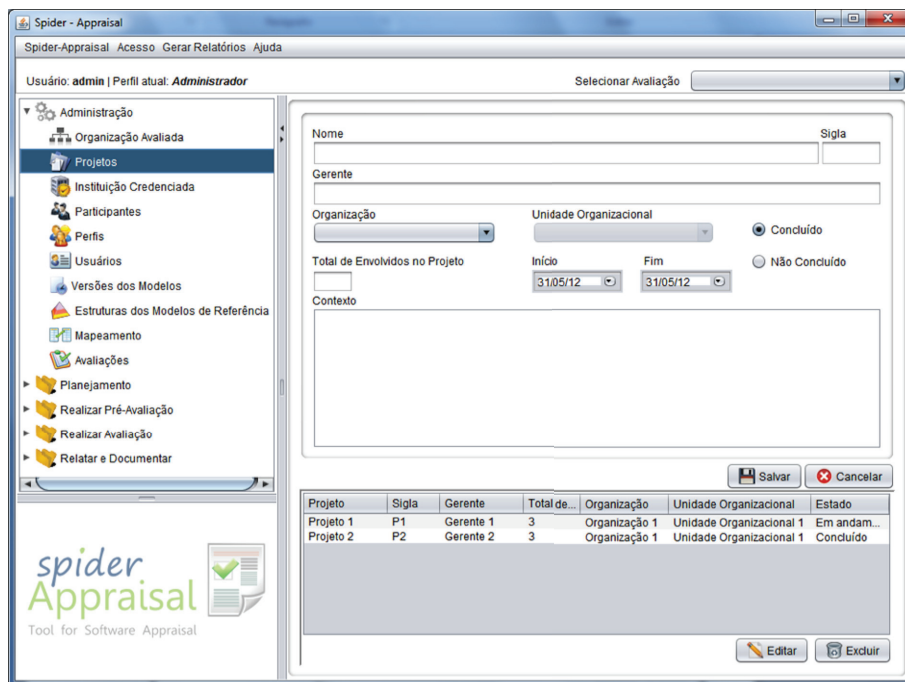


Figura 170. Tela Projetos

O cadastro de Instituição Credenciada permite inserir os dados acerca das instituições credenciadas (vide Figura 171), que podem ser de três tipos: Instituição Avaliadora (IA), Instituição Implementadora (II) e Instituição Organizadora de Grupo de Empresas (IOGE). Cada um desses tipos exerce um papel diferente no decorrer da avaliação. Uma instituição pode possuir mais de um tipo, e também pode estar credenciada para fazer avaliações em um ou ambos os modelos (CMMI e/ou MPS.BR).

Spider - Appraisal

Spider-Appraisal Acesso Gerar Relatórios Ajuda

Usuário: admin | Perfil atual: Administrador

Selecionar Avaliação

Administração

- Organização Avaliada
- Projetos
- Instituição Credenciada
- Participantes
- Perfis
- Usuários
- Versões dos Modelos
- Estruturas dos Modelos de Referência
- Mapeamento
- Avaliações
- Planejamento
- Realizar Pré-Avaliação
- Realizar Avaliação
- Relatar e Documentar

Nome CNPJ

Site

Endereço

País Estado Cidade CEP

Nome do Contato E-mail do Contato

Coordenador E-mail do Coordenador

Telefone Fax

Credenciado Para:

☐ CMMI ☐ MPS.BR

Tipos de Instituição:

Salvar Cancelar

Nome	Endereço	Tipos	CNPJ	Telefone
Instituição Credenciada 1	Av. Endereço	II, IA, IOGE	00.000.000/00...	() - /
Instituição Avaliadora 1	Av. Endereço	IA	00.000.000/00...	() - /
Instituição Implementadora 1	Av. Endereço	II	00.000.000/00...	() - /

Editar Excluir

spider Appraisal
Tool for Software Appraisal

Figura 171. Tela Instituição Credenciada

O cadastro de participantes permite registrar as pessoas que poderão ser entrevistadas pelos avaliadores posteriormente, sendo que essas necessariamente devem estar envolvidas nos projetos que serão avaliados (vide Figura 172).

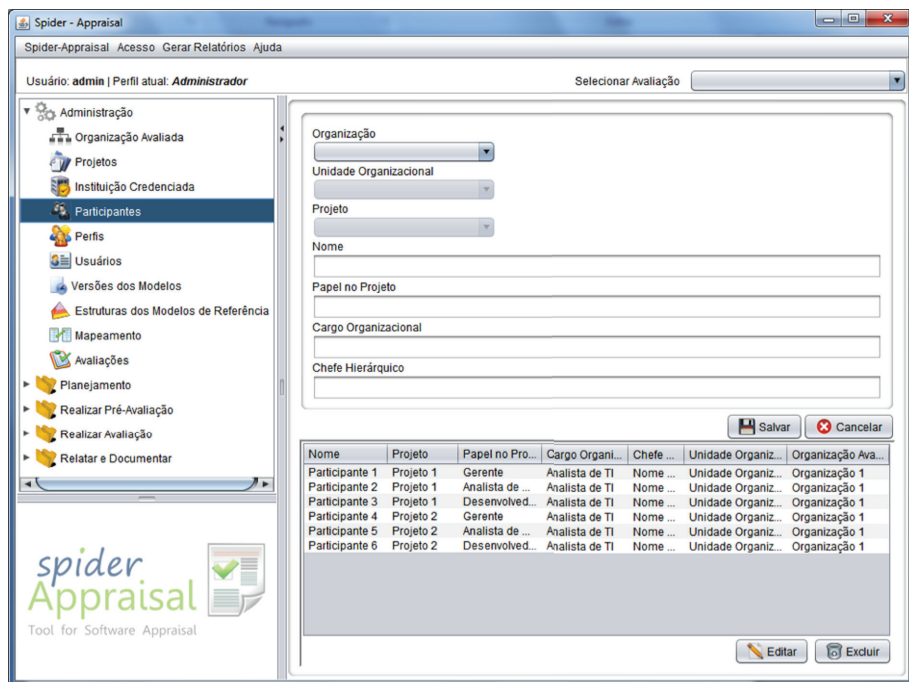


Figura 172. Tela Participantes

O cadastro de perfis possibilita que novos perfis de usuários sejam criados, além daqueles que veem previamente cadastrados na ferramenta, quais sejam: Administrador, Avaliador, Avaliado, Auditor, Implementador, Patrocinador e Coordenador da Instituição Avaliadora; sendo que tais perfis não podem ser editados, nem excluídos, por refletirem os atores principais do framework que são responsáveis por realizar uma avaliação. Para cadastrar um novo perfil deve ser preenchido um nome, uma descrição, e deve ser associado um Vínculo Institucional, que determina a qual tipo de instituição poderá pertencer um usuário cadastrado com esse novo perfil (vide Figura 173). O cadastro de usuários é explicado com mais detalhes a frente. Os vínculos institucionais possíveis são: Instituição Avaliadora, Organização Avaliada, Instituição Implementadora, IOGE e Administração do Sistema. Apenas para “Administração do Sistema” não existe atribuição

de instituição no momento do cadastro de usuário que tiver um perfil com esse vínculo.

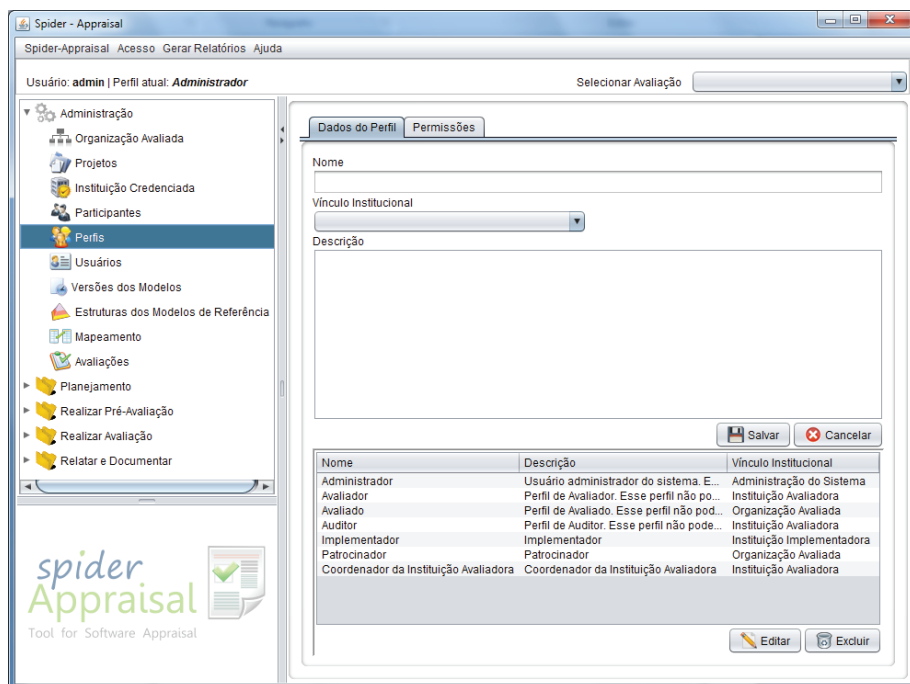


Figura 173. Tela Perfis – Aba Dados do Perfil

O cadastro de permissões possibilita que seja configurada a permissão de acesso às funcionalidades da ferramenta, sendo que existe uma configuração padrão de acesso atribuída para os perfis que já vem pré-cadastrados, mas que pode ser alterada conforme necessário. Os itens que estiverem marcados representam as funcionalidades que poderão ser acessadas por determinado perfil. Do contrário, os itens desmarcados indicam que tais funcionalidades não poderão ser executadas pelo perfil selecionado, conforme ilustrado na Figura 174. O botão “Padrão” executa a ação de restaurar as configurações de permissões do perfil selecionado.

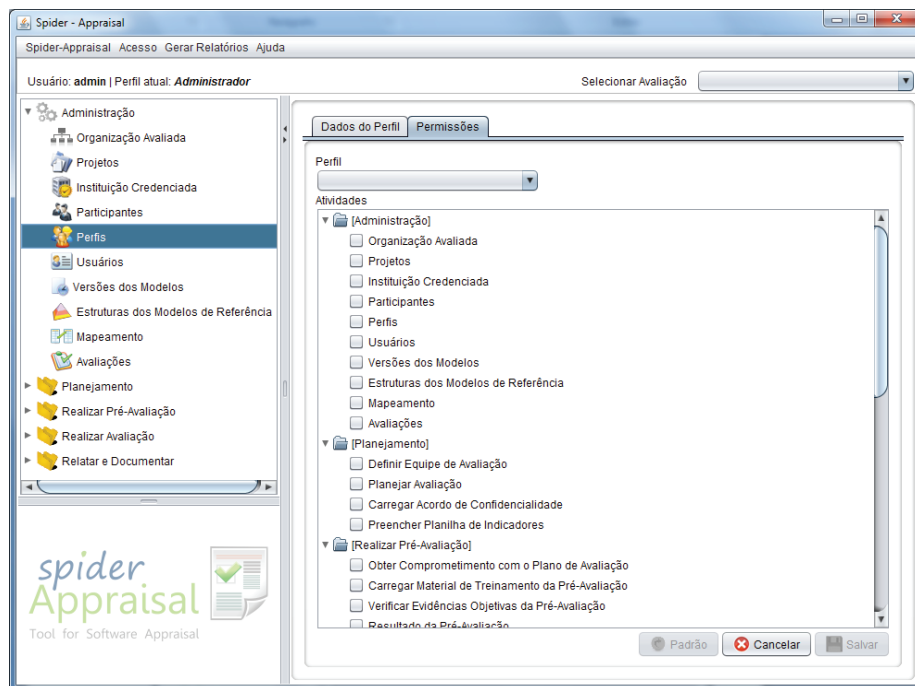


Figura 174. Tela Perfis - Aba Permissões

O cadastro de usuários permite registrar os dados das pessoas que poderão fazer *login* na ferramenta, a fim de executar as atividades de avaliação de processos. Na primeira aba “Dados Pessoais” são inseridas informações referentes aos dados pessoais (vide Figura 175), e na aba “Dados de Acesso” são informados o *login*, a senha, e os perfis que cada usuário poderá assumir.

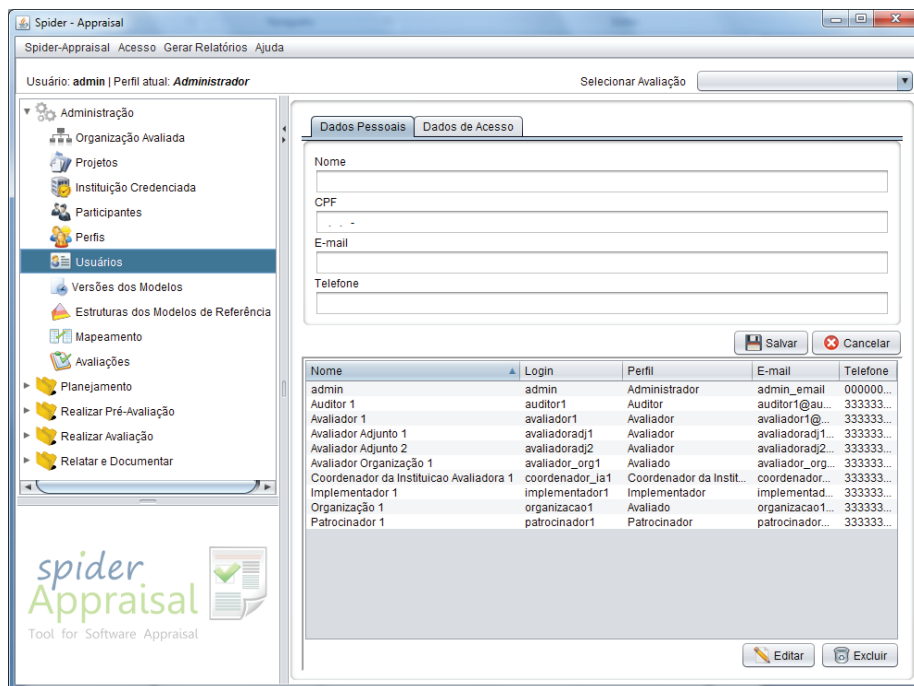


Figura 175. Tela Usuários

No cadastro de versões dos modelos são registradas as versões para os modelos de referência MR-MPS-SW e CMMI-DEV, caso seja necessário inserir na ferramenta a estrutura de novas versões dos referidos modelos. Para cadastrar uma nova versão, deve-se escolher o modelo (MR-MPS-SW ou CMMI-DEV) e preencher o nome da versão, como visto na Figura 176.

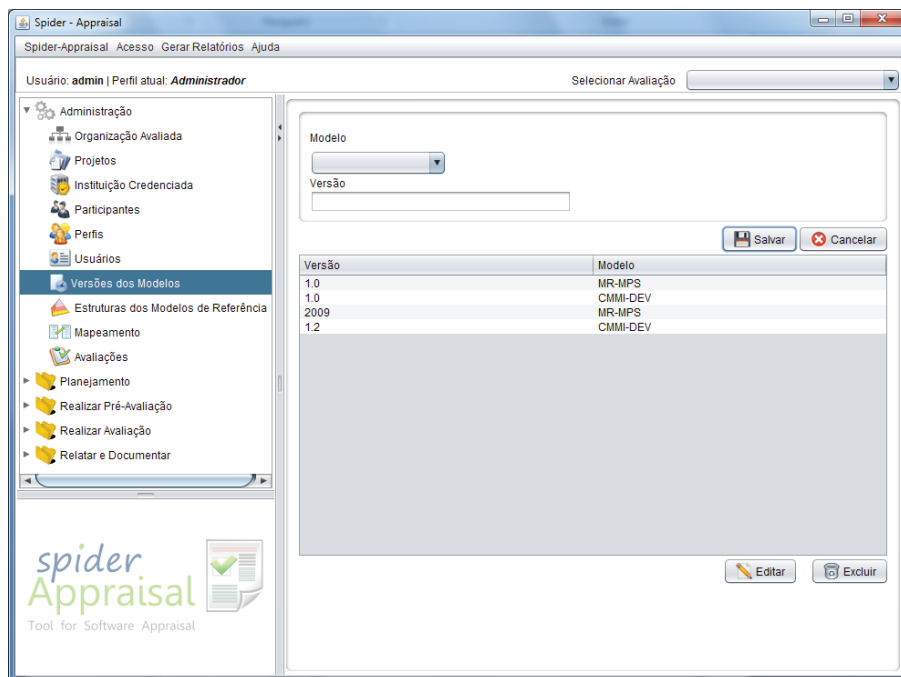


Figura 176. Tela Versões dos Modelos

No cadastro da estrutura dos modelos de referência são inseridos todos os ativos concernentes a uma versão de um modelo de referência (MR-MPS-SW ou CMMI-DEV), como por exemplo: níveis de maturidade, processos/ áreas de processo, resultados esperados (MR-MPS-SW), atributos de processo (MR-MPS-SW), objetivos específicos (CMMI-DEV), práticas específicas (CMMI-DEV), objetivos genéricos (CMMI-DEV), práticas genéricas (CMMI-DEV), dentre outros (vide Figura 177).

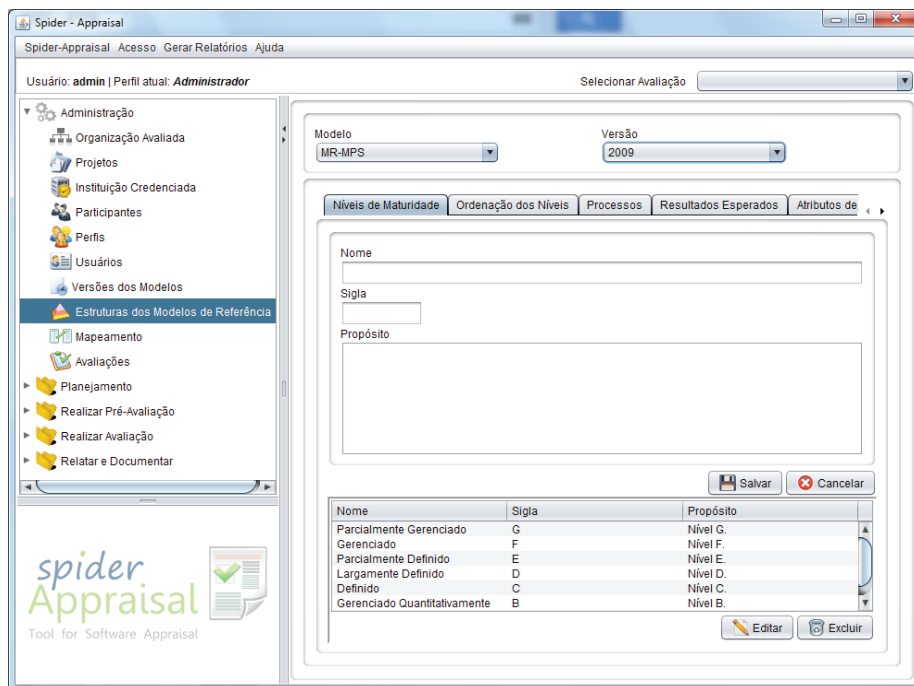


Figura 177. Tela Estrutura dos Modelos de Referência

O cadastro de mapeamento dos níveis permite indicar a equivalência existente entre os níveis de maturidade de uma determinada versão do CMMI-DEV e os níveis de uma versão do modelo MR-MPS-SW (vide Figura 178). Isso possibilita que na avaliação conjunta seja determinado o nível de maturidade em ambos os modelos de referência.

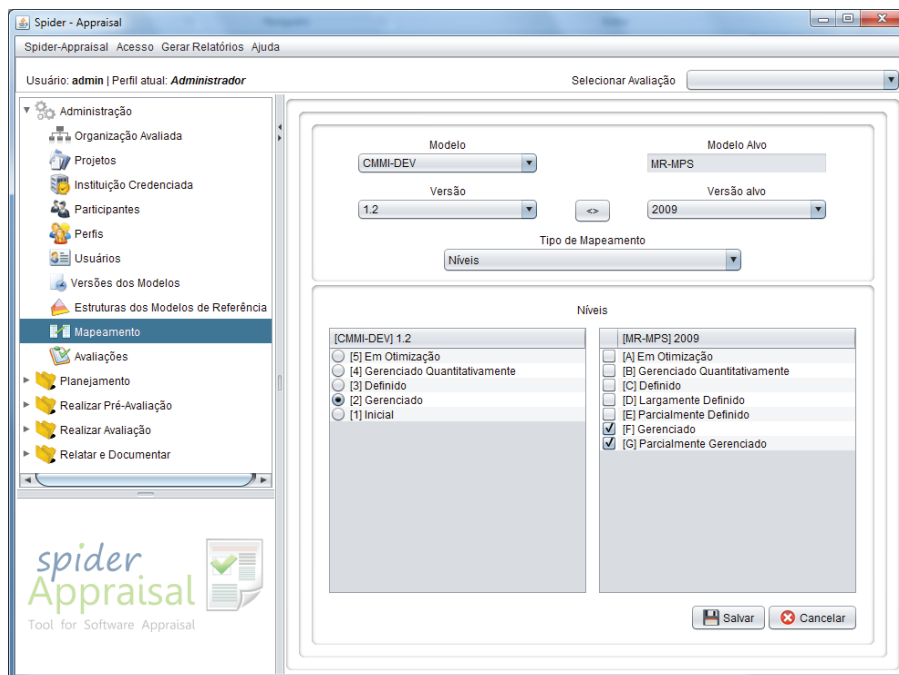


Figura 178. Tela Mapeamento de Níveis

O cadastro de mapeamento dos processos permite indicar a equivalência entre os processos de uma determinada versão do modelo MR-MPS-SW e as áreas de processo de uma versão do CMMI-DEV (vide Figura 179). Isso possibilita que na avaliação conjunta seja visualizada a equivalência existente entre as áreas de processo e os processos dos modelos de referência.

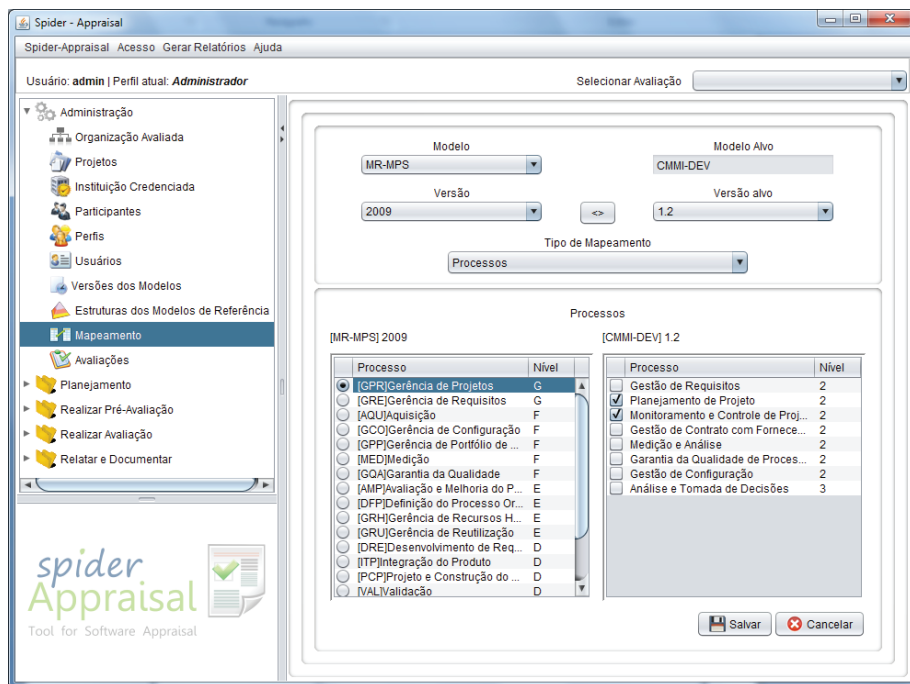


Figura 179. Tela Mapeamento de Processos

O cadastro de mapeamento do resultado esperado x prática específica permite indicar a equivalência entre os resultados esperados de uma determinada versão do MR-MPS-SW e as práticas específicas de uma versão do modelo CMMI-DEV (vide Figura 180). Isso possibilita que na avaliação conjunta seja visualizada a equivalência existente entre os resultados esperados e as práticas específicas de ambos os modelos de referência.

The screenshot shows the 'Spider - Appraisal' application window. The left sidebar contains a navigation tree with options like 'Administração', 'Organização Avaliada', 'Projetos', 'Instituição Credenciada', 'Participantes', 'Perfis', 'Usuários', 'Versões dos Modelos', 'Estruturas dos Modelos de Referência', 'Mapeamento', 'Avaliações', 'Planejamento', 'Realizar Pré-Avaliação', 'Realizar Avaliação', and 'Relatar e Documentar'. The main area is titled 'Resultado Esperado x Prática Específica' and contains several sections:

- Modelo:** A dropdown menu set to 'MR-MPS'.
- Modelo Alvo:** A dropdown menu set to 'CMMI-DEV'.
- Versão:** A dropdown menu set to '2009'.
- Versão alvo:** A dropdown menu set to '1.2'.
- Tipo de Mapeamento:** A dropdown menu set to 'Resultado Esperado x Prática Específica'.
- Resultado Esperado x Prática Específica:** This section contains two tables:
 - [MR-MPS] 2009:** A table with columns 'VAL', 'VER', 'GDE', 'DRU', 'GRI', 'GRH', 'GRU', 'DRE', 'ITP', 'PCP', 'GPP', 'MED', 'GQA', 'AMP', 'DFP', 'GPR', 'GRE', 'AQU', and 'GCO'. Below this is a list of 'Gerência de Projetos' (GPR 1 to GPR 9) with a 'Nível' column.
 - [CMMI-DEV] 1.2:** A table with columns 'PP', 'PMC', and 'Nível'. It lists various practices (SP 1.1 to SP 3.2) with their corresponding levels.
- Buttons:** 'Salvar' (Save) and 'Cancelar' (Cancel) buttons at the bottom right.

Figura 180. Tela Mapeamento de Resultado Esperado x Prática Específica

O cadastro de mapeamento do resultado de atributo de processo x prática genérica permite indicar a equivalência entre os resultados de atributo de processo (RAP) de uma determinada versão do MR-MPS-SW e as práticas genéricas de uma versão do modelo CMMI-DEV (vide Figura 181). Isso possibilita que na avaliação conjunta seja visualizada a equivalência existente entre os resultados de atributo de processo e as práticas genéricas de ambos os modelos de referência.

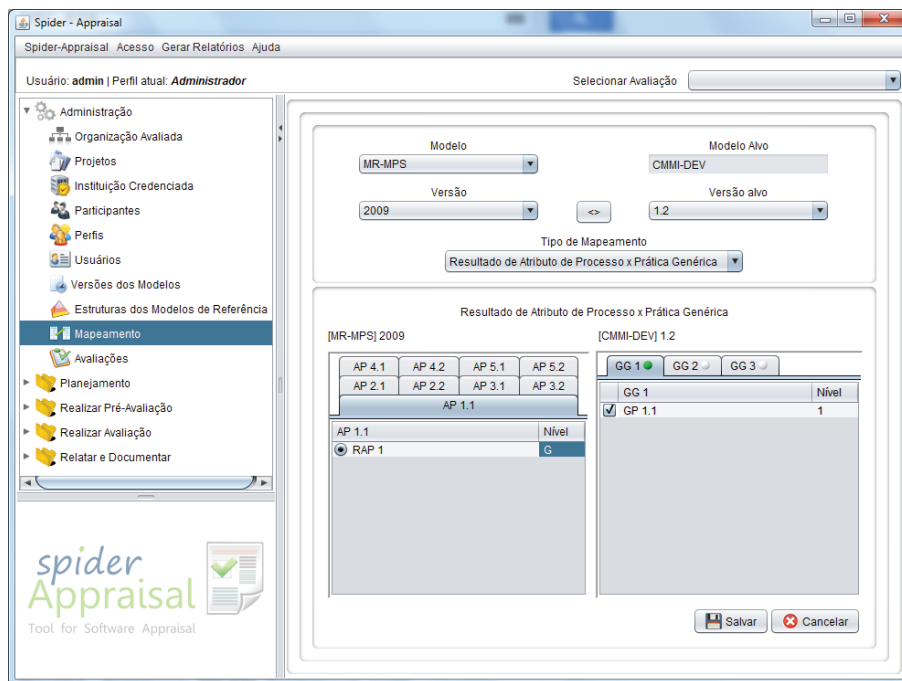


Figura 181. Tela Mapeamento de RAP x Prática Genérica

O cadastro de mapeamento do resultado de atributo de processo x prática específica permite indicar a equivalência entre os resultados de atributo de processo de uma determinada versão do MR-MPS-SW e as práticas específicas de uma versão do modelo CMMI-DEV (vide Figura 182). Isso possibilita que na avaliação conjunta seja visualizada a equivalência existente entre os resultados de atributo de processo do MR-MPS-SW e as práticas específicas do CMMI-DEV.

Esse mapeamento é necessário para os mais altos níveis de maturidade, já que nos níveis A e B do modelo MR-MPS-SW, os resultados de atributos de processos (RAP) destes níveis são comparados também com as áreas de processos do CMMI-DEV dos níveis 4 e 5 e suas respectivas práticas específicas [SOFTEX, 2012b].

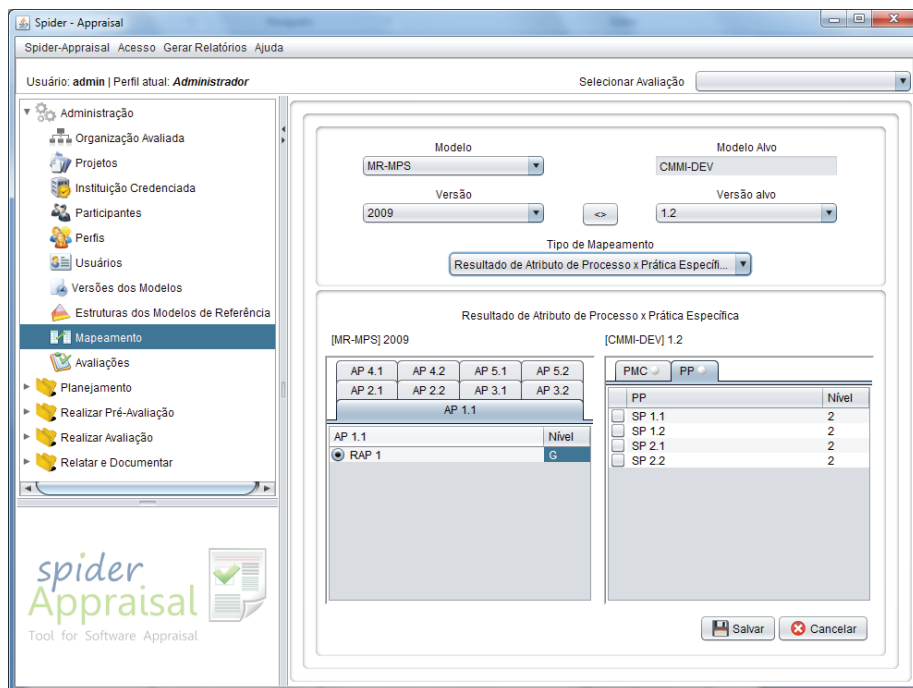


Figura 182. Tela Mapeamento de RAP x Prática Específica

No cadastro de avaliações são criadas as avaliações com a inserção de informações que a caracterizam, como: título, os modelos de referência envolvidos (CMMI e/ou MPS.BR), o tipo de avaliação, a organização avaliada, a unidade organizacional, a instituição avaliadora CMMI e/ou MPS.BR, avaliador líder CMMI e/ou MPS.BR, instituição implementadora, implementador, IOGE e auditor (vide Figura 183). Se forem selecionados ambos os modelos (CMMI e MPS.BR), isso significa que se trata de uma avaliação integrada (conjunta) e obrigatoriamente o seu tipo será uma avaliação estagiada.

Spider - Appraisal
Spider-Appraisal Acesso Gerar Relatórios Ajuda

Usuário: **admin** | Perfil atual: **Administrador** Selecionar Avaliação

Administração

- Organização Avaliada
- Projetos
- Instituição Credenciada
- Participantes
- Perfis
- Usuários
- Versões dos Modelos
- Estruturas dos Modelos de Referência
- Mapeamento
- Avaliações**
 - Planejamento
 - Realizar Pré-Avaliação
 - Realizar Avaliação
 - Relatar e Documentar

spider Appraisal
Tool for Software Appraisal

Título

Modelo

- ☒ CMMI
- ☒ MPS.BR

CMMI
1.2

MPS.BR
2009

Método
SCAMPI A & MA-MPS

Tipo de Avaliação

- ☐ Contínua
- ☒ Estagiada

☒ Ativo

Organização

Unidade Organizacional

Instituição Avaliadora CMMI

Avaliador Líder CMMI

Instituição Avaliadora MPS.BR

Avaliador Líder MPS.BR

Instituição Implementadora

Implementador

IOGE

Auditor

Salvar **Cancelar**

Título	Método	Organiza...	Unidade...	Institui...	Avali...	Institui...	Av...	Est...
Avaliação Conjunta 1...	SCAMPI A & MA-MPS	Organiza...	Unidade...	Institui...	Avali...	Institui...	Av...	Ativ...
Avaliação MPS.BR 1 ...	MA-MPS	Organiza...	Unidade...	-	-	Institui...	Av...	Ativ...

Editar **Excluir**

Figura 183. Tela Avaliações

21.2.3. Módulo de Execução da Avaliação

Para uma dada avaliação selecionada, a ferramenta possibilita que as atividades de avaliação sejam executadas de maneira sequencial. Sendo assim, para realizar uma atividade é preciso que a atividade imediatamente anterior tenha sido marcada como “Atividade concluída”. Essas atividades estão distribuídas em quatro fases do processo de avaliação, no menu lateral: Planejar e Preparar, Realizar Pré-Avaliação, Realizar Avaliação e Relatar e Documentar Resultados. Nas seções seguintes é explicada cada uma das atividades por fase. Para ilustrar o funcionamento da ferramenta, são mostradas as telas referentes a uma avaliação do tipo integrada, que envolve ambos os modelos, MR-MPS-SW e CMMI-DEV. Mas vale ressaltar que a ferramenta também

executa avaliações simples, que envolvem apenas um dos modelos (MR-MPS-SW ou CMMI-DEV).

A primeira fase do processo de avaliação sistematizado pela ferramenta chama-se **Planejar e Preparar**, que consiste em atividades de planejamento e preparação que precedem a realização da avaliação de fato. A seguir são explicadas as atividades dessa fase que estão presentes na ferramenta Spider-Appraisal.

Primeiramente o avaliador líder precisa definir os membros da equipe de avaliação. A equipe de avaliação deve ser composta por avaliador líder e avaliadores adjuntos (vinculados a uma instituição avaliadora) e por representantes da unidade organizacional (que são membros da organização que está sendo avaliada). Como pode ser visto na Figura 184, na tela de Definir Equipe de Avaliação é possível indicar os avaliadores adjuntos e os representantes da unidade organizacional que serão membros da equipe de avaliação. Nesse momento, os avaliadores líderes já foram previamente definidos no cadastro da avaliação, no módulo de administração da ferramenta, sendo que nesta tela eles apenas são exibidos para consulta. Em se tratando de uma avaliação integrada, a equipe de avaliação possuirá dois avaliadores líderes, um para o MPS.BR e outro para o CMMI. Também pode ser acionada a funcionalidade de gerenciar usuários, caso seja necessário cadastrar novos usuários do tipo avaliador que deverão fazer parte da equipe de avaliação.

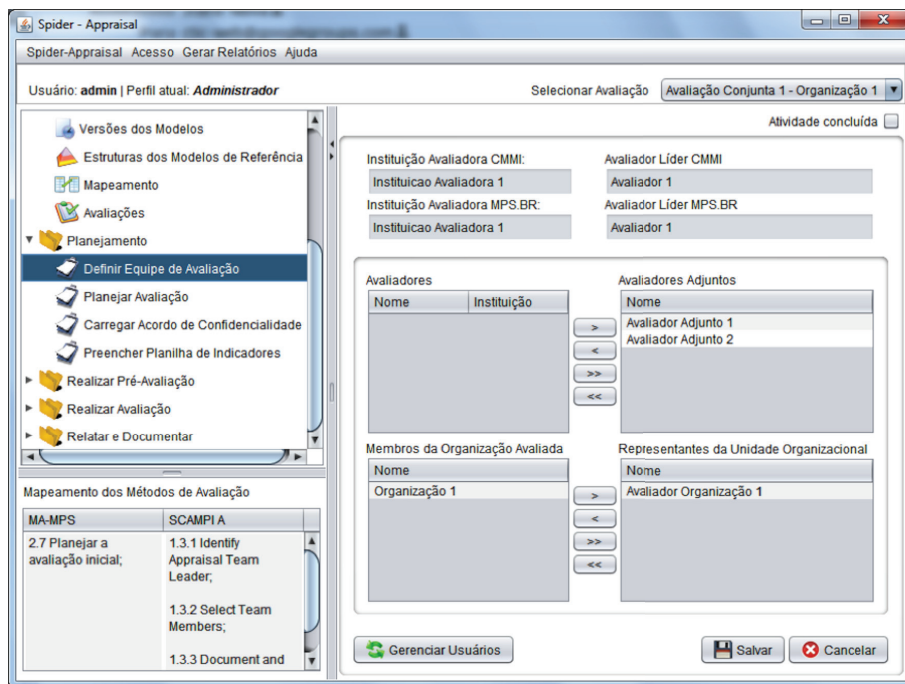


Figura 184. Tela Definir Equipe de Avaliação

Na atividade Planejar Avaliação, o avaliador líder deve preencher todas as informações referentes: a) à organização, indicando qual o patrocinador da avaliação; b) ao escopo (ver Figura 185), especificando o nível alvo de maturidade da avaliação e selecionando os processos que serão avaliados; c) aos projetos, selecionando os projetos de software em andamento e/ou concluídos que serão avaliados; d) à equipe de avaliação, possibilitando que sejam alterados os membros da equipe de avaliação alocados na atividade anterior, se necessário; e) aos participantes, listando as pessoas que serão entrevistadas durante a avaliação; e f) ao cronograma, informando as datas de início e fim da pré-avaliação e da avaliação final, bem como especificando a validade da avaliação. Todas essas informações consolidam o plano de avaliação, que pode ser gerado pela ferramenta na forma de relatório, através do menu Gerar Relatórios.

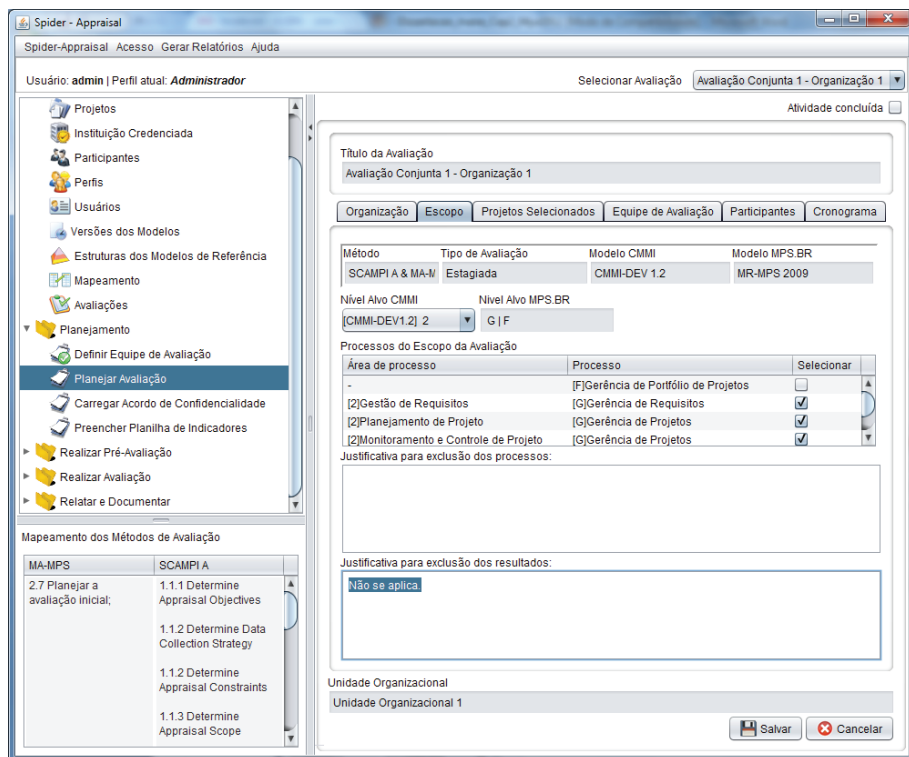


Figura 185. Tela Planejar Avaliação – Aba Escopo

Na atividade Carregar Acordo de Confidencialidade, podem ser carregadas várias versões do Acordo de Confidencialidade, sendo preciso preencher uma descrição e selecionar o arquivo correspondente, conforme visto na Figura 186. Após salvar, automaticamente será armazenada a data e hora que o arquivo foi carregado na ferramenta. A partir daí, as versões poderão ser consultadas a qualquer momento através do botão “Visualizar”, ser alteradas através do botão “Editar”, ou excluídas através do botão “Excluir”.

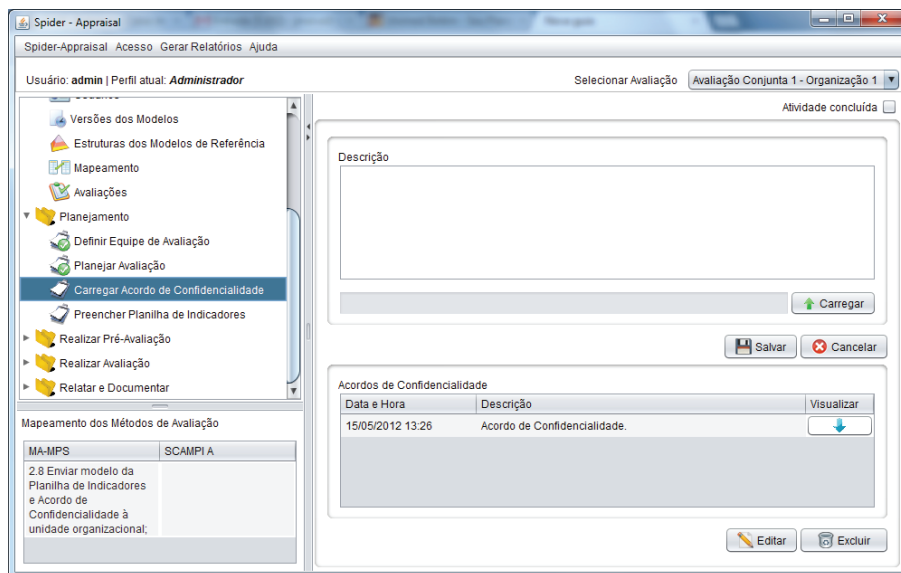


Figura 186. Tela Carregar Acordo de Confidencialidade

Na atividade Preencher Planilha de Indicadores, serão registradas as evidências objetivas iniciais obtidas a partir dos artefatos e dos processos da Unidade Organizacional, referentes à implementação de cada prática/resulta- do esperado do(s) modelo(s) de referência, respeitando-se o escopo da avalia- ção. Primeiro devem ser cadastrados os produtos de trabalho (ver Figura 187), preenchendo-se o nome, a fonte, e indicando se é organizacional e os projetos a que está associado. Para cada projeto selecionado, pode ser carregado um arquivo para ser consultado posteriormente.

Produto de trabalho trata-se de um resultado útil de um processo. Isso pode incluir arquivos, documentos, produtos, partes de um produto, serviços, descrições de processo, especificações e faturas [SEI, 2010].

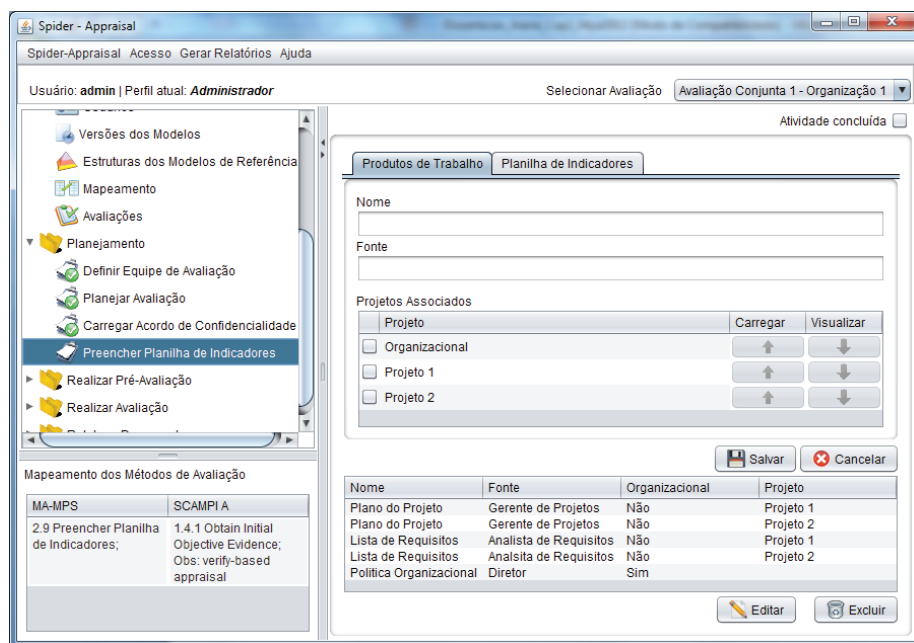


Figura 187. Tela Preencher Planilha de Indicadores – Aba Produtos de Trabalho

O próximo passo é o preenchimento da planilha de indicadores de fato, sendo que para cada processo dentro do escopo da avaliação deverão ser registradas para cada resultado esperado as evidências objetivas encontradas. Conforme a Figura 188, podem ser selecionados resultados esperados e resultados de atributos de processos. Os resultados esperados estão distribuídos em abas, e quando selecionada a aba de um resultado esperado é exibido seu propósito e podem ser acrescentadas as evidências através do botão “Adicionar”. O procedimento é idêntico para os resultados de atributos de processos. As informações que identificam uma evidência são: produto de trabalho, tipo da evidência, se é organizacional e os projetos associados, e uma descrição (ver Figura 189). Após cadastradas, as evidências são exibidas na forma de lista na área mais inferior da tela da planilha de indicadores, e podem ser editadas ou excluídas conforme necessário.

Quando se tratar de uma avaliação integrada, a planilha de indicadores exibirá áreas que indicam o mapeamento entre os processos e áreas de

processo, e entre os resultados esperados e práticas específicas, respectivamente do MR-MPS-SW e do CMMI-DEV (ver Figura 188). Essas informações auxiliam os envolvidos no processo de avaliação a visualizar a equivalência existente entre os modelos MR-MPS-SW e CMMI-DEV durante a realização da avaliação integrada.

Spider - Appraisal

Spider-Appraisal Acesso Gerar Relatórios Ajuda

Usuário: **admin** | Perfil atual: **Administrador**

Selecionar Avaliação: **Avaliação Conjunta 1 - Organização 1**

Atividade concluída ☐

Produtos de Trabalho | **Planilha de Indicadores**

Processo: **[MR-MPS2009] Gerência de Projetos**

Mapeamento do Processo:

- [CMMI-DEV - 1.2] Planejamento de Projeto
- [CMMI-DEV - 1.2] Monitoramento e Controle de Projeto
- [CMMI-DEV - 1.2] Gestão Integrada de Projeto +IPPD

☒ Resultados Esperados ☐ Resultados de Atributos de Processos

OPR 1 **OPR 2** **OPR 3** **OPR 4** **OPR 5** **OPR 6** **OPR 7** **OPR 8** **OPR 9** **OPR 10** **OPR 11**

Propósito: O escopo do trabalho para o projeto é definido.

Mapeamento:

Sigla	Prática Específica
SP 1.1	Estimar o Escopo do Projeto

Evidência	Tipo	Fonte	ORG	P1	P2
Plano do Projeto - Seção de E...	AD	Gerente de Projetos		<input type="radio"/>	<input type="radio"/>
Plano do Projeto - Seção de E...	AD	Gerente de Projetos		<input type="radio"/>	<input type="radio"/>
Plano do Projeto - Seção 3.	AD	Gerente de Projetos		<input type="radio"/>	<input type="radio"/>
Política Organizacional - Seção...	AI	Diretor		<input type="radio"/>	<input type="radio"/>

Legenda

☒ Adequado ☐ Tem Fraqueza ☐ Inadequado ☐ Não Avaliado

Mapeamento dos Métodos de Avaliação

MA-MPS	SCAMPIA
2.9 Preencher Planilha de Indicadores;	1.4.1 Obtain Initial Objective Evidence; Obs: verify-based appraisal

Figura 188. Tela Preencher Planilha de Indicadores – Aba Planilha de Indicadores

The image shows a software window titled "Spider - Appraisal". It contains two dropdown menus at the top: "Produto de Trabalho" (set to "Plano do Projeto") and "Tipo da Evidência" (set to "Artefato Direto"). Below these is a table with the heading "Projetos". The table has two columns: "Projeto" and "Visualizar". The first row shows "Projeto 1" with a checked checkbox in the "Projeto" column and a "Visualizar" button with a downward arrow in the "Visualizar" column. Below the table is a text area labeled "Descrição" containing the text "Seção 1.". At the bottom right are two buttons: "Salvar" (with a floppy disk icon) and "Cancelar" (with a red X icon).

Projeto	Visualizar
<input checked="" type="checkbox"/> Projeto 1	

Figura 189. Tela Adicionar Evidência

A fase **Realizar Pré-Avaliação** possui, sobretudo, atividades de validação para assegurar que as evidências iniciais coletadas são adequadas e suficientes, antes de prosseguir para a avaliação final. A seguir são explicadas as atividades dessa fase que são sistematizadas pela ferramenta Spider-Appraisal.

Na atividade Obter Comprometimento, após obtidas as assinaturas de comprometimento dos participantes da avaliação com o Plano de Avaliação e com o Acordo de Confidencialidade, é possível carregar estes documentos digitalizados para ficarem armazenados na ferramenta. Para tanto, basta preencher uma descrição e selecionar o arquivo que será carregado, separadamente para cada documento na aba Plano de Avaliação (ver Figura 190) e na aba Acordo de Confidencialidade.

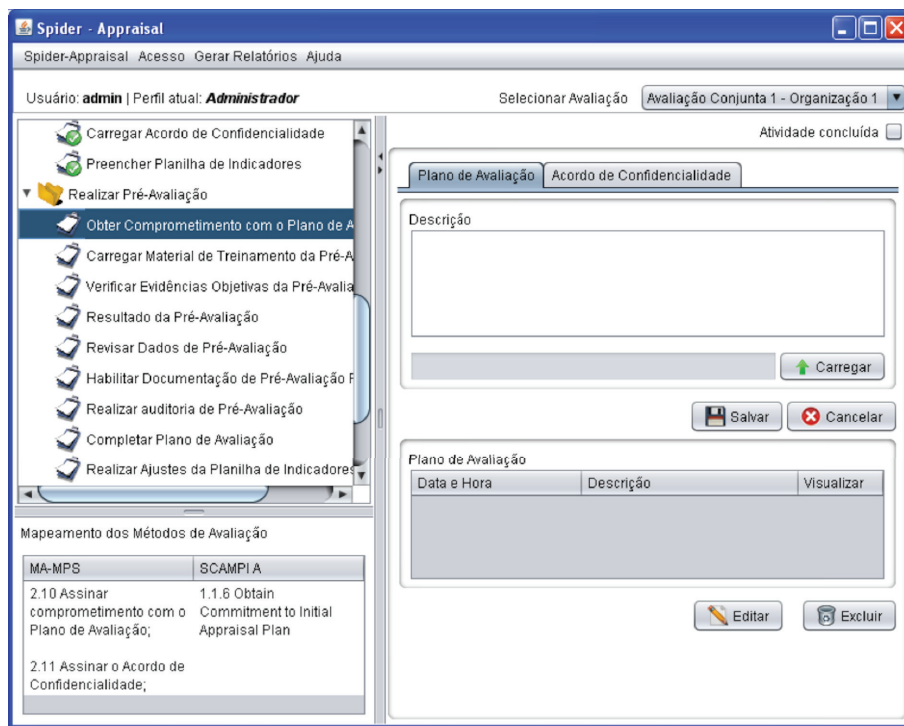


Figura 190. Tela Obter Comprometimento – Aba Plano de Avaliação

Na atividade Carregar Material de Treinamento da Pré-Avaliação é possível carregar o material de apoio utilizado para realizar o treinamento da equipe de avaliação para a pré-avaliação, sendo que normalmente o material é no formato de apresentação de slides. Ao clicar no botão “Carregar”, será aberta uma janela para pesquisar o arquivo no computador e, então, o arquivo selecionado será armazenado na ferramenta. A qualquer momento o material de treinamento poderá ser consultado pelos interessados, clicando-se no botão “Visualizar” (ver Figura 191).

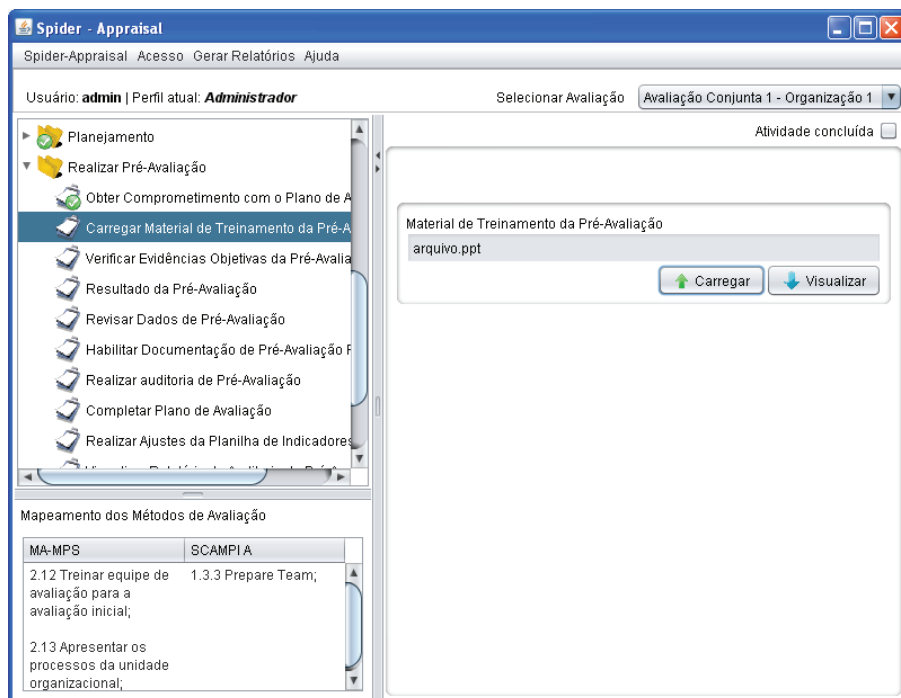


Figura 191. Tela Carregar Material de Treinamento da Pré-Avaliação

Na atividade Verificar Evidências Objetivas da Pré-Avaliação os avaliadores realizarão o exame das evidências objetivas iniciais contidas na Planilha de Indicadores, a fim de detectar se as evidências objetivas de cada resultado esperado/prática do escopo da avaliação estão adequadas e completas para prosseguir com a avaliação conforme planejado. Os avaliadores terão que classificar cada evidência para cada projeto como sendo Adequado (cor verde), Tem Fraqueza (cor amarela), Inadequado (cor vermelha), ou Não Avaliado (cor azul), conforme pode ser visto na Figura 192. A cor cinza significa que ainda não foi atribuída nenhuma classificação, portanto, simboliza o vazio. Essa classificação deverá ser realizada para todos os resultados esperados e resultados de atributos de processo de todos os processos pertencentes ao escopo da avaliação. Também é permitido adicionar, editar ou excluir evidências, conforme necessário.

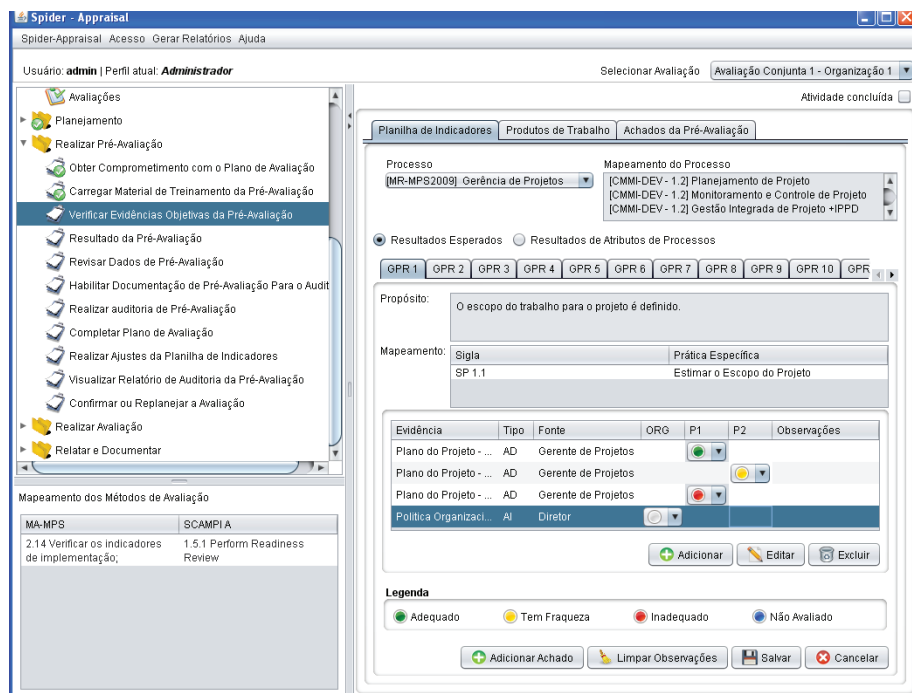


Figura 192. Tela Verificar Evidências Objetivas da Pré-Avaliação – Aba Planilha de Indicadores

Na aba Achados da Pré-Avaliação devem ser cadastrados os achados para cada processo. Na tela principal (vide Figura 193) deve ser selecionado um processo, e para tal devem ser cadastrados os achados através do botão “Adicionar”, o qual exibe uma tela para preencher os seguintes campos (vide Figura 194): selecionar o resultado esperado ou o resultado de atributo de processo, a classificação (ponto fraco ou oportunidade de melhoria), descrever os problemas e a sugestão para corrigir. Os achados cadastrados são exibidos na tela principal na forma de lista e podem ser editados ou excluídos, conforme necessário.

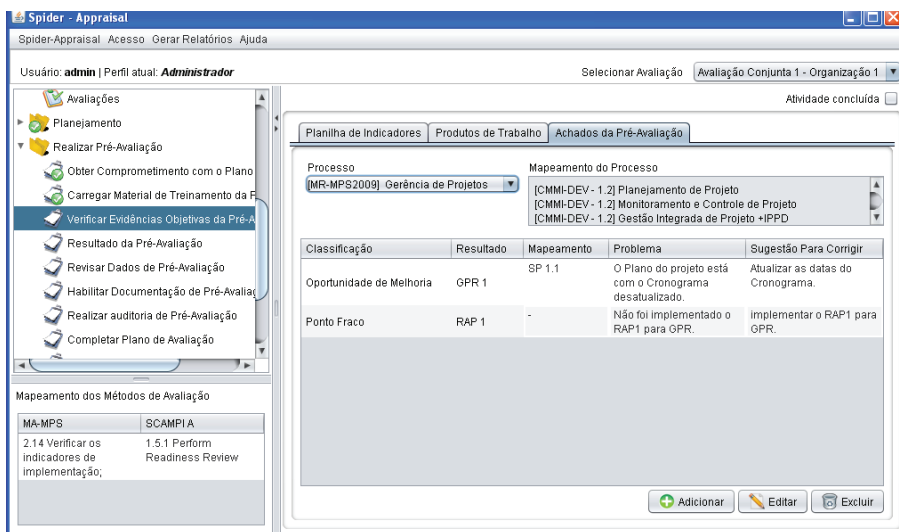


Figura 193. Tela Verificar Evidências Objetivas da Pré-Avaliação – Aba Achados da Pré-Avaliação

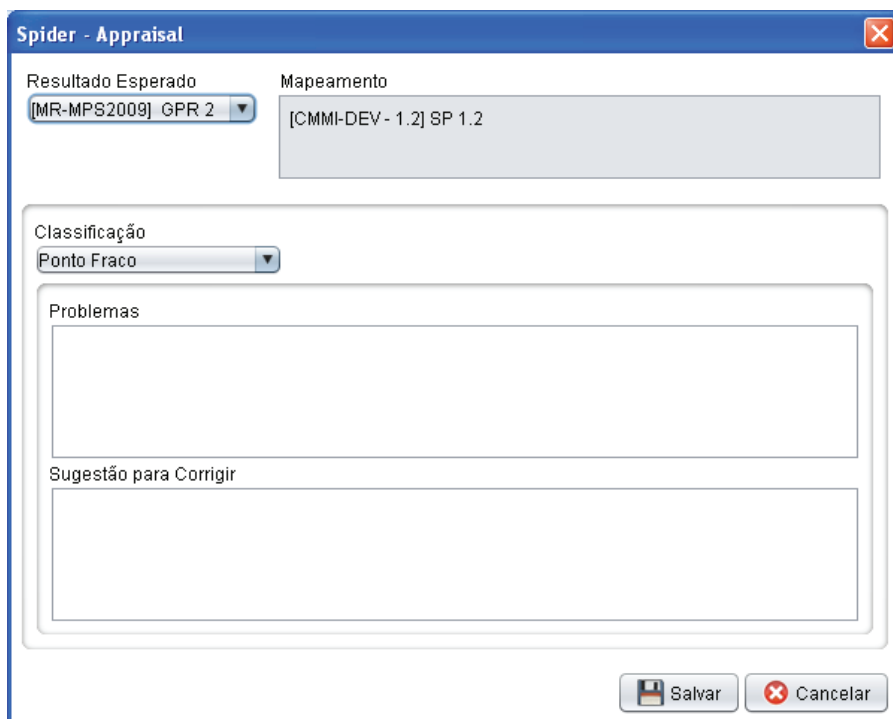


Figura 194. Tela Adicionar Achado

Na atividade Preencher Resultado da Pré-Avaliação é feito preenchimento do Resultado da Pré-Avaliação, que consiste em um relatório que indica o estado da avaliação até o momento da realização da pré-avaliação. O Resultado da Pré-avaliação contém o registro tanto de novas informações, como de alterações em informações planejadas anteriormente, como pode ser visualizado parcialmente na Figura 195 (já que é muito extenso e a tela possui uma barra de rolagem para consultar o relatório por inteiro). Após preencher e clicar no botão “Salvar”, é possível gerar o relatório em formato PDF através do menu “Gerar Relatórios”.

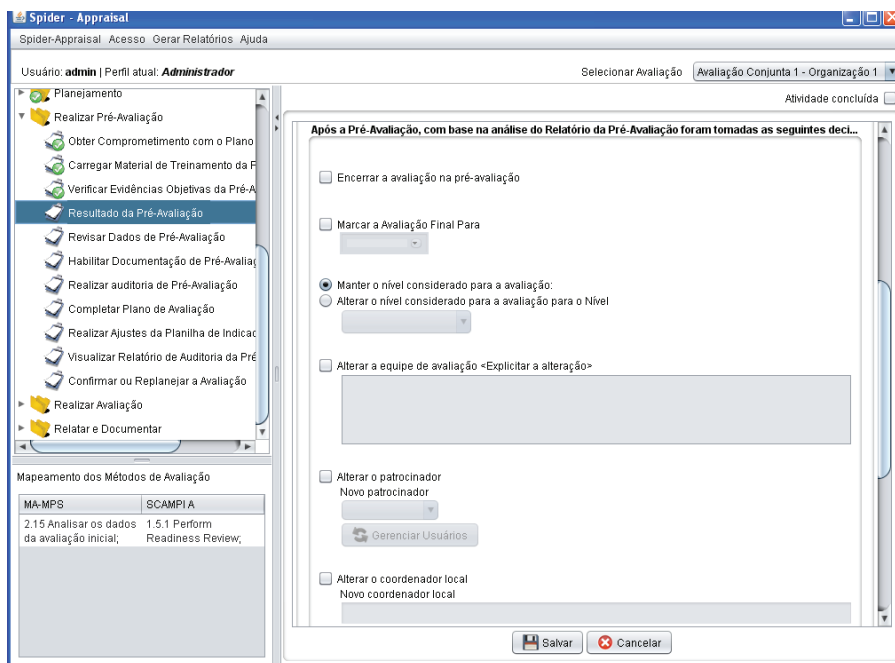


Figura195. Tela Preencher Resultado da Pré-Avaliação

A atividade Revisar Dados da Pré-Avaliação admite a revisão e a validação dos dados resultantes da verificação das evidências objetivas, incluindo-se a análise dos achados da pré-avaliação e do resultado da pré-avaliação, sendo que estes podem ser alterados, se necessário. Além disso, serve

de marco a partir do qual será decidido quanto à viabilidade de prosseguir como planejado, a necessidade de replanejar, ou mesmo de cancelar a avaliação; portanto, pode implicar em alteração do plano de avaliação também. A Figura 196 ilustra a tela de revisão que contém as abas Revisar Achados da Pré-Avaliação, Revisar Resultado da Pré-Avaliação e Revisar Plano de Avaliação.

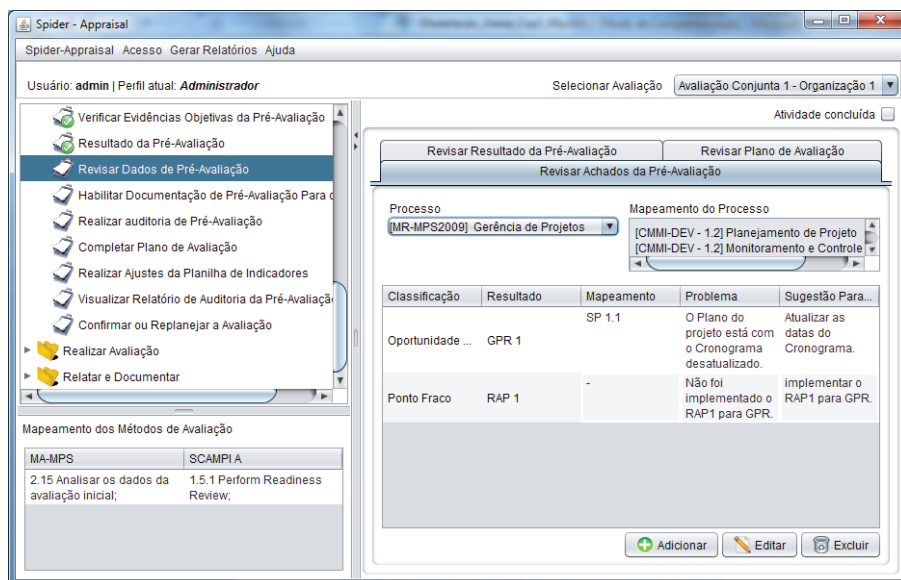


Figura 196. Tela Revisar Dados da Pré-Avaliação – Aba Revisar Achados

A atividade Habilitar para o Auditor a Documentação da Pré-Avaliação tem por finalidade habilitar a documentação da pré-avaliação para que o auditor responsável possa dar início ao trabalho de auditoria da avaliação. Como mostrado na Figura 197, devem ser selecionados os documentos que deverão estar disponíveis para o auditor e, então, clicar no botão “Salvar” e marcar a atividade como “Atividade concluída”. Após isso, o Auditor poderá executar a atividade “Realizar auditoria da Pré-Avaliação”, em que poderá acessar e visualizar apenas os documentos que foram habilitados. Importante ressaltar que essa e outras atividades de auditoria só

aparecerão no menu da ferramenta se no cadastro da avaliação tenha sido designado um auditor para a avaliação, caso contrário essa atividade será omitida do menu e não será executada.

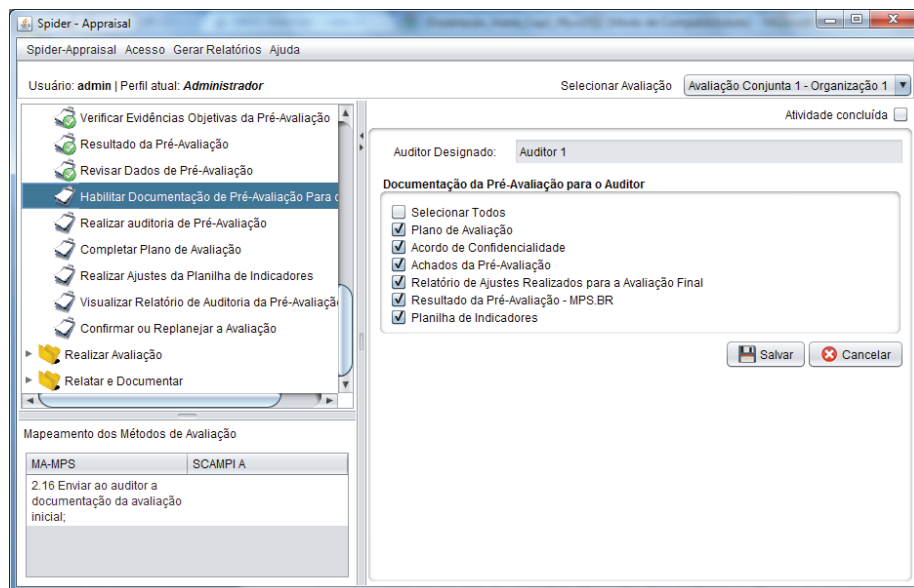


Figura 197. Tela Habilitar para o Auditor a Documentação da Pré-Avaliação

Na atividade Realizar Auditoria da Pré-Avaliação o auditor realizará a auditoria do trabalho feito pela equipe de avaliação até o momento da Pré-Avaliação. Como mostrado na Figura 198, o auditor poderá acessar os documentos que foram habilitados para a auditoria, listados na área “Documentos da Pré-Avaliação”, clicando-se em cima do nome do documento (Plano de Avaliação, Acordo de Confidencialidade, etc.) para que este seja aberto para consulta. Após a análise, o auditor deverá indicar se o documento está completo (sim ou não) e correto (sim ou não). O relatório da auditoria também possui os seguintes campos a serem preenchidos: descrição detalhada de todos os problemas encontrados; itens que necessitam de correção, apreciação sobre o resultado da avaliação (não há dúvidas ou há dúvidas); e, por fim, o parecer (aprovado, aprovado com modificações, ou não aprovado). Após preencher e

clicar no botão “Salvar”, é possível gerar o relatório em formato PDF através do menu “Gerar Relatórios”.

Spider - Appraisal
Spider-Appraisal Acesso Gerar Relatórios Ajuda

Usuário: admin | Perfil atual: Administrador Selecionar Avaliação Avaliação Conjunta 1 - Organização 1 Atividade concluída

Realizar Auditoria de Pré-Avaliação

AVALIADOR LÍDER MPS.BR: Avaliador 1
AVALIADOR LÍDER CMMI: Avaliador 1
INSTITUIÇÃO AVALIADORA MPS.BR: Institucao Avaliadora 1
INSTITUIÇÃO AVALIADORA CMMI: Institucao Avaliadora 1
ORGANIZAÇÃO AVALIADA: Organização 1
DATA DA PRE-AVALIAÇÃO: 24/05/2012
DATA DA AVALIAÇÃO: 01/06/2012

Documento	Completo?	Correto?
Plano de Avaliação	Sim	Sim
Acordo de Confidencialidade	Sim	Sim
Planilha de Indicadores	Sim	Sim
Relatório de Ajustes Realizados para a Avaliação Final	Sim	Sim

Descrição detalhada de todos os problemas encontrados

Itens que necessitam de correção

Apreciação sobre o resultado da avaliação na empresa

☒ Não há dúvidas sobre o resultado da avaliação
☐ Há dúvidas sobre o resultado da avaliação

Parecer

☐ APROVADO
☒ APROVADO COM MODIFICAÇÕES
☐ NÃO APROVADO

Salvar Cancelar

Figura 198. Tela Realizar Auditoria da Pré-Avaliação

A atividade Completar Plano de Avaliação possibilita aos avaliadores completar e/ou revisar o planejamento da avaliação, de acordo com os resultados obtidos na pré-Avaliação. Dessa forma, consiste apenas em revisar o plano de avaliação e realizar alterações se houver necessidade, conforme visto na Figura 199.

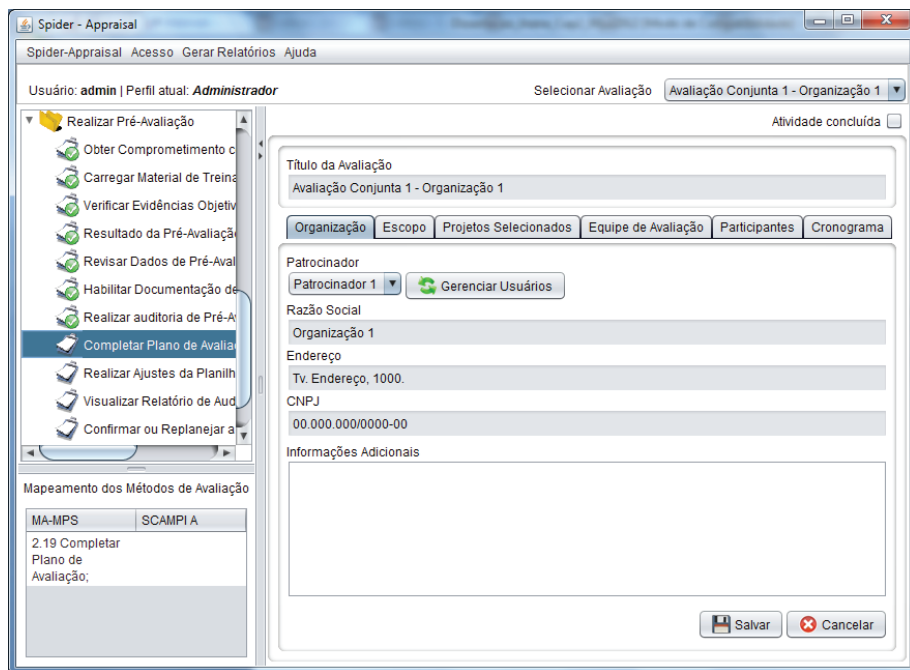


Figura 199. Tela Completar Plano de Avaliação

A atividade Realizar Ajustes da Planilha de Indicadores possibilita o ajuste da planilha de indicadores para resolver os problemas (pontos fracos ou oportunidades de melhoria) registrados pelos avaliadores como achados. A organização avaliada deverá examinar os achados cadastrados na aba Achados da Pré-Avaliação (ver Figura 200), consultando os problemas e as sugestões para corrigir e, então, realizar os ajustes necessários na aba Planilha de Indicadores e/ou na aba Produtos de Trabalho. Depois de realizar os ajustes, a organização avaliada deve preencher o campo “Correção Realizada” para cada achado na aba Achados da Pré-Avaliação, através do botão “Editar”, conforme Figura 201.

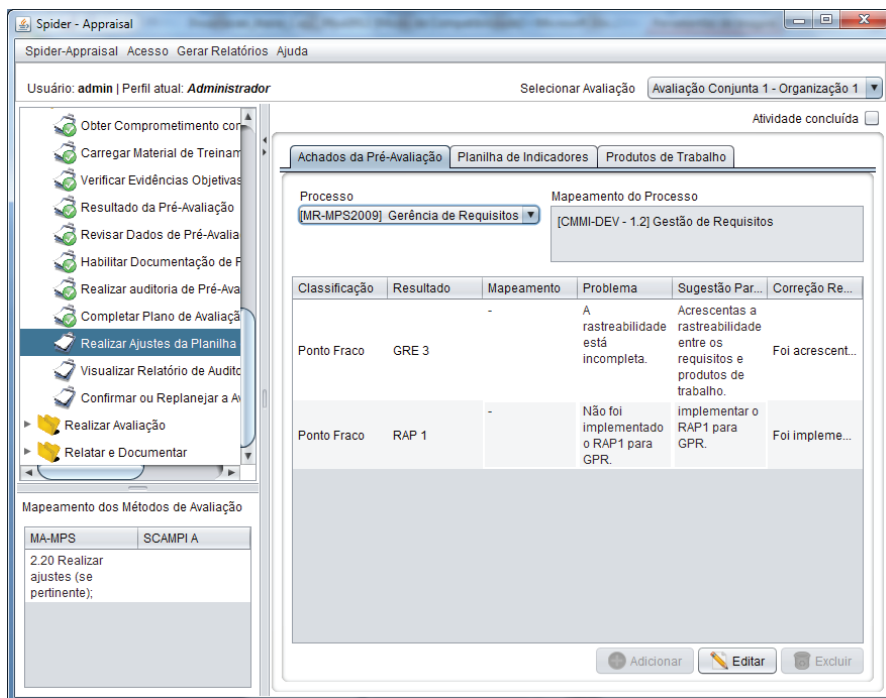


Figura 200. Tela Realizar Ajustes da Planilha de Indicadores – Aba Achados da Pré-Avaliação

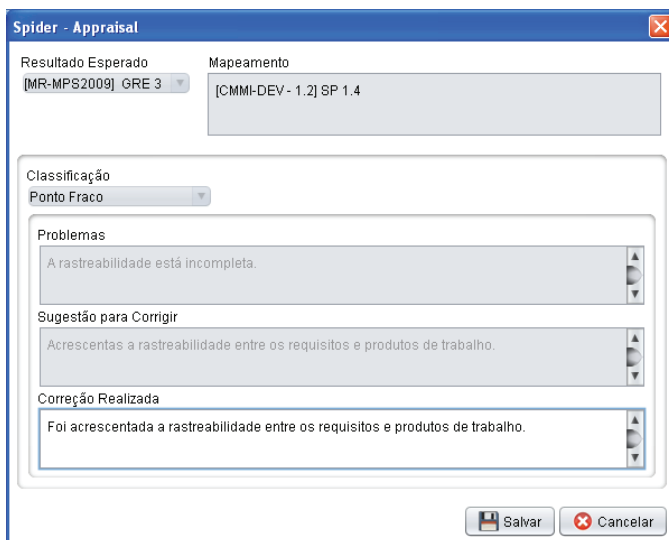


Figura 201. Tela Editar Achado

Na atividade Visualizar Relatório de Auditoria da Pré-Avaliação os avaliadores têm acesso ao relatório de auditoria da pré-avaliação que foi preenchido pelo auditor, no qual podem consultar todos os problemas encontrados na documentação da pré-avaliação, os itens que necessitam de correção, bem como o parecer final. Se o parecer tiver sido “aprovado”, nenhuma correção significativa deverá ser realizada. Caso seja “aprovado com modificações”, os avaliadores terão que realizar as correções apontadas pelo auditor. E se o parecer for “não aprovado”, foram detectadas inconsistências e irregularidades que não podem ser reparadas e, portanto, a avaliação deverá ser cancelada. A Figura 202 apresenta o relatório de auditoria, o qual pode ser acessado apenas para visualização e, sendo assim, a funcionalidade de edição está desabilitada para os avaliadores. É possível gerar o relatório em formato PDF através do menu “Gerar Relatórios”.

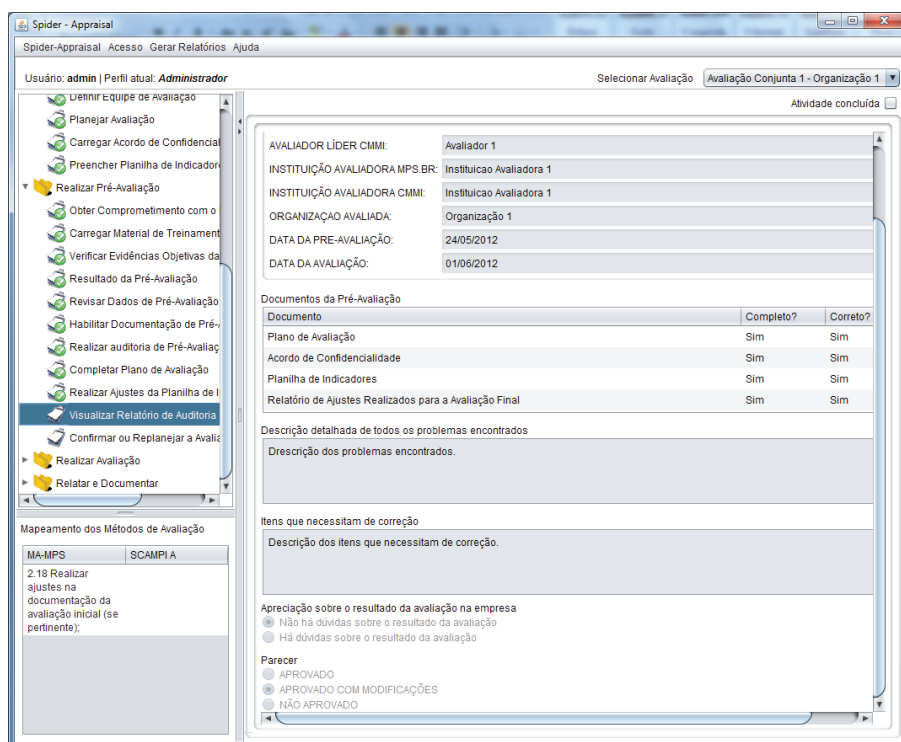


Figura 202. Tela Visualizar Relatório de Auditoria da Pré-Avaliação

Nesse momento os avaliadores devem confirmar a realização da avaliação ou, caso haja desvios e não se proceda como planejado, da renegociação do planejamento da avaliação com o patrocinador. Para tanto, essa atividade disponibiliza os achados da pré-avaliação, o resultado da pré-avaliação, e o plano de avaliação para serem revisados através de abas (ver Figura 203), os quais servem de subsídios para a decisão quanto ao prosseguimento da avaliação. Vale observar que essa funcionalidade na ferramenta é idêntica à atividade Revisar Dados da Pré-Avaliação; entretanto, o que difere é o momento em que são executadas e os desdobramentos das ações que podem ser diferentes em cada marco.

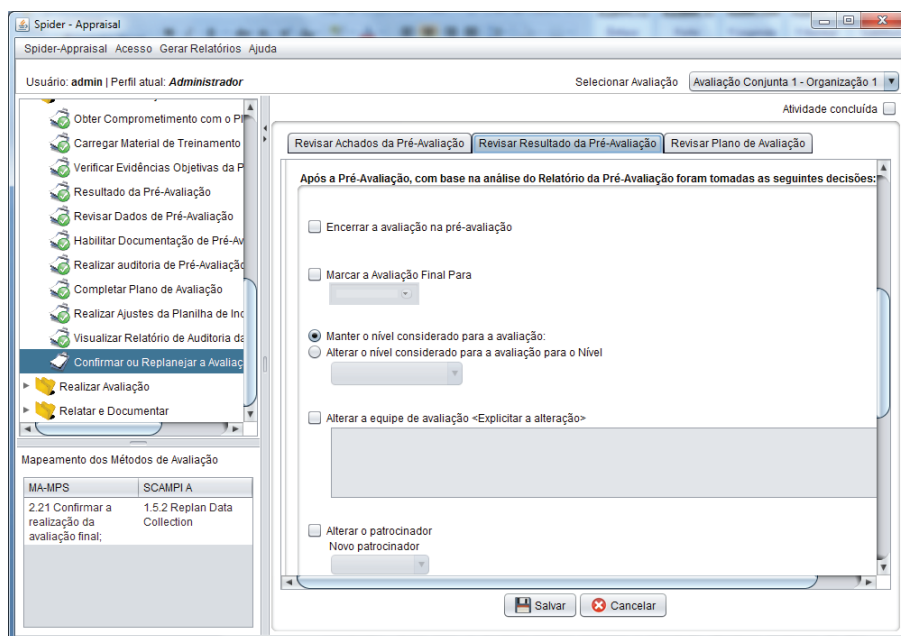


Figura 203. Tela Confirmar a Avaliação – Aba Revisar Resultado da Pré-Avaliação

A fase de **Realizar Avaliação** corresponde à avaliação formal e final, que possui essencialmente atividades de validação e pontuação, onde é verificada a presença de indicadores que comprovem a implementação das práticas/resultados dos modelos e são atribuídas pontuações. A seguir são

explicadas as atividades dessa fase que são sistematizadas pela ferramenta Spider-Appraisal.

Na atividade Carregar Material de Treinamento de Avaliação é possível carregar o material de apoio utilizado para realizar o treinamento da equipe de avaliação para a avaliação final, sendo que normalmente o material é no formato de apresentação de slides. Ao clicar no botão “Carregar”, será aberta uma janela para pesquisar o arquivo no computador e, então, o arquivo selecionado será armazenado na ferramenta. A qualquer momento o material de treinamento poderá ser consultado pelos interessados, clicando-se no botão “Visualizar” (ver Figura 204).

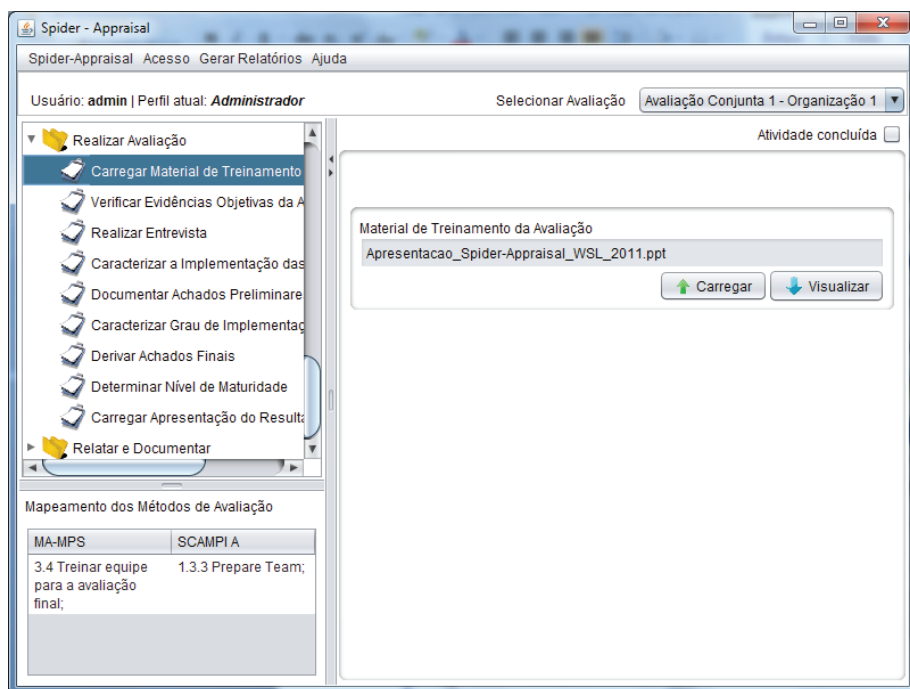


Figura 204. Tela Carregar Material de Treinamento da Avaliação

A atividade Verificar Evidências Objetivas da Avaliação compreende a verificação das evidências oriundas de artefatos da organização, que estão contidas na Planilha de Indicadores, a fim de detectar se as evidências

objetivas de cada resultado esperado/prática do escopo da avaliação estão adequadas e completas. Na aba Planilha de Indicadores, os avaliadores terão que classificar cada evidência para cada projeto como sendo Adequado (cor verde), Tem Fraqueza (cor amarela), Inadequado (cor vermelha), ou Não Avaliado (cor azul), conforme pode ser visto na Figura 205. A cor cinza significa que ainda não foi atribuída nenhuma classificação, portanto, simboliza o vazio. Essa classificação deverá ser realizada para todos os resultados esperados e resultados de atributos de processo de todo o processo pertencente ao escopo da avaliação. Também é permitido adicionar, editar ou excluir evidências, conforme necessário. Nessa atividade ainda pode ser feito o registro de anotações referentes às evidências, no campo “Observações” da tela “Editar Evidência”, conforme Figura 206.

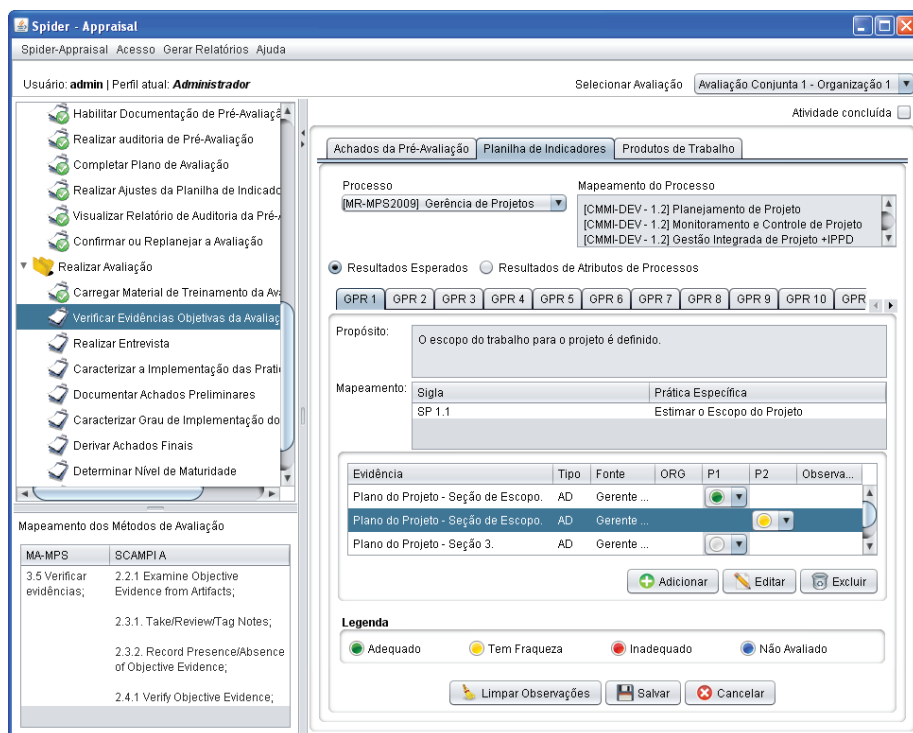


Figura 205. Tela Verificar Evidências Objetivas da Avaliação – Aba Planilha de Indicadores

The image shows a software window titled "Spider - Appraisal". It contains several input fields and buttons. At the top, there are two dropdown menus: "Produto de Trabalho" (set to "Plano do Projeto") and "Tipo da Evidência" (set to "Artefato Direto"). Below these is a section titled "Projetos" containing a table with one row: "Projeto" (checked) and "Projeto 1". To the right of the table is a "Visualizar" button with a downward arrow. Below the table is a large text area labeled "Descrição" containing the text "Seção de Escopo.". At the bottom is another large text area labeled "Observações". At the very bottom of the window are two buttons: "Salvar" (with a floppy disk icon) and "Cancelar" (with a red X icon).

Projeto	Visualizar
<input checked="" type="checkbox"/> Projeto 1	↓

Figura 206. Tela Editar Evidência

Na aba Achados da Pré-Avaliação (ver Figura 207), na tela principal são exibidos os achados cadastrados anteriormente para cada processo na forma de lista, os quais podem ser editados ou excluídos, conforme necessário.

Para cadastrar novos achados, na tela principal deve ser selecionado um processo e para tal devem ser cadastrados os achados através do botão “Adicionar”, o qual exibe uma tela para preencher os seguintes campos (ver Figura 208): selecionar o resultado esperado ou o resultado de atributo de processo, a classificação (ponto fraco ou oportunidade de melhoria), descrever os problemas e a sugestão para corrigir. Para editar um achado, basta selecioná-lo na lista da tela principal e clicar no botão “Editar”, o qual exibe uma tela para editar os mesmos campos de adicionar achado, e um campo adicional “Correção Realizada” (ver Figura 209).

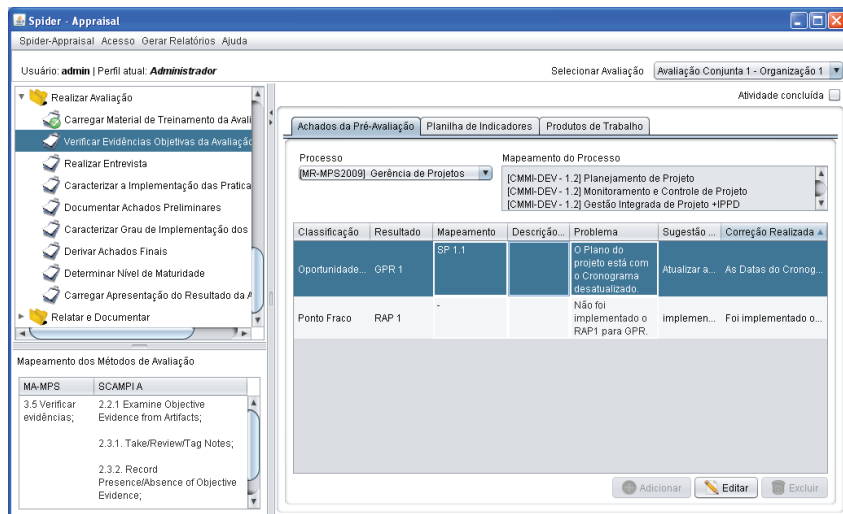


Figura 207. Tela Verificar Evidências Objetivas da Avaliação – Aba Achados da Pré-Avaliação

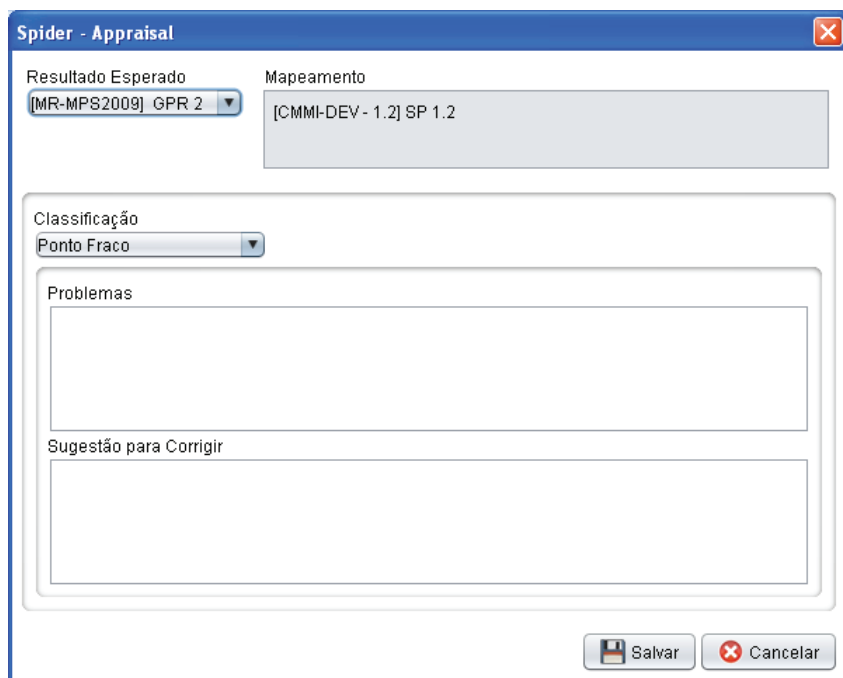


Figura 208. Tela Adicionar Achado

The image shows a software window titled "Spider - Appraisal". It contains several input fields and text areas for recording appraisal findings and corrections.

Resultado Esperado	Mapeamento
[MR-MPS2009] GPR 1	[CMMI-DEV - 1.2] SP 1.1

Classificação
Oportunidade de Melhoria

Problemas
O Plano do projeto está com o Cronograma desatualizado.

Sugestão para Corrigir
Atualizar as datas do Cronograma.

Correção Realizada
As Datas do Cronograma foram atualizadas.

Buttons: Salvar, Cancelar

Figura 209. Tela Editar Achado

A atividade Realizar Entrevista possibilita que sejam registradas as anotações tomadas durante as entrevistas com os membros da unidade organizacional que participam dos projetos que estão no escopo da avaliação. Na tela principal (ver Figura 210), primeiro devem ser cadastrados os grupos que serão entrevistados, preenchendo-se os seguintes campos: grupo entrevistado (gerente de projeto, desenvolvedores, etc.), número de participantes e data da entrevista. Após clicar no botão “Salvar”, os grupos cadastrados são listados na tela principal e pode-se realizar as operações de “Editar” ou “Excluir” para cada grupo selecionado.

Para iniciar uma entrevista, basta selecionar um grupo da lista e clicar no botão “Realizar Entrevista”, que abrirá uma nova janela (ver Figura 211) contendo as abas: “Entrevista com”, “Achados”, “Achados Gerais” e “Planilha de Indicadores”.

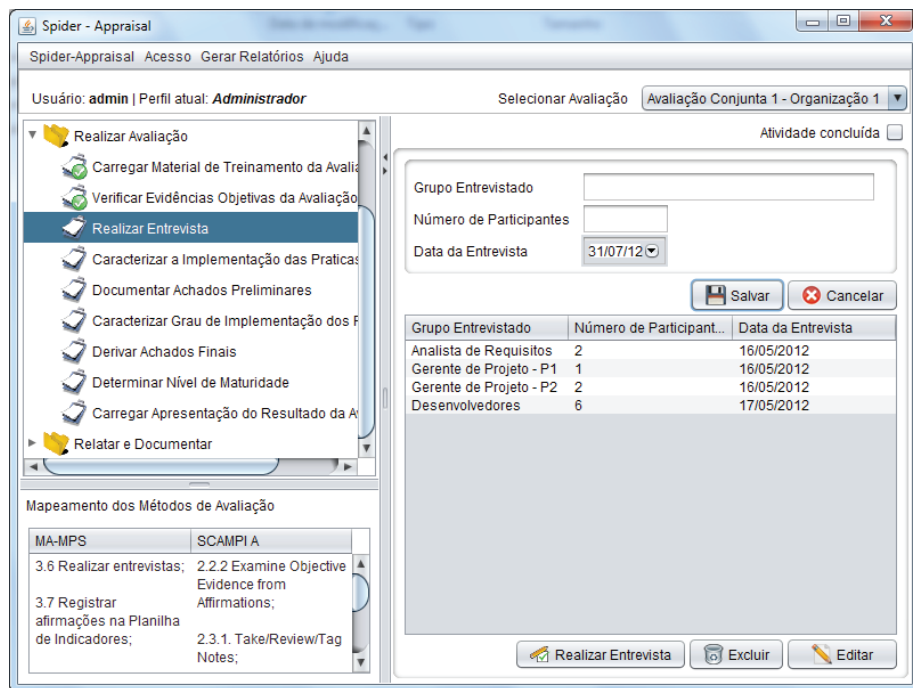


Figura 210. Tela Realizar Entrevista

Na aba Entrevista com “Grupo”, aparecerá a imagem de um homem de costas (representação do avaliador) e tantas imagens de bonecos (representação dos entrevistados) quanto o número de participantes que tiver sido cadastrado para o grupo entrevistado. Inicialmente aparecem nomes genéricos para os bonecos (“Pessoa 1”, “Pessoa 2”, e assim por diante) e estes devem ser renomeados para os nomes reais dos participantes entrevistados, bastando-se clicar com o botão direito sobre um boneco e escolher a opção “Renomear”. No exemplo ilustrado na Figura 210, o grupo entrevistado é o de “Analista de Requisitos” e o número de participantes é 2 (“Pessoa 1” e “Pessoa 2”), sendo que o “Pessoa 1” foi renomeado para “Joao”. Para registrar a entrevista realizada, basta clicar com o botão direito sobre o boneco entrevistado e escolher a opção “Abrir Conversa”, então será aberta a janela da Figura 212.

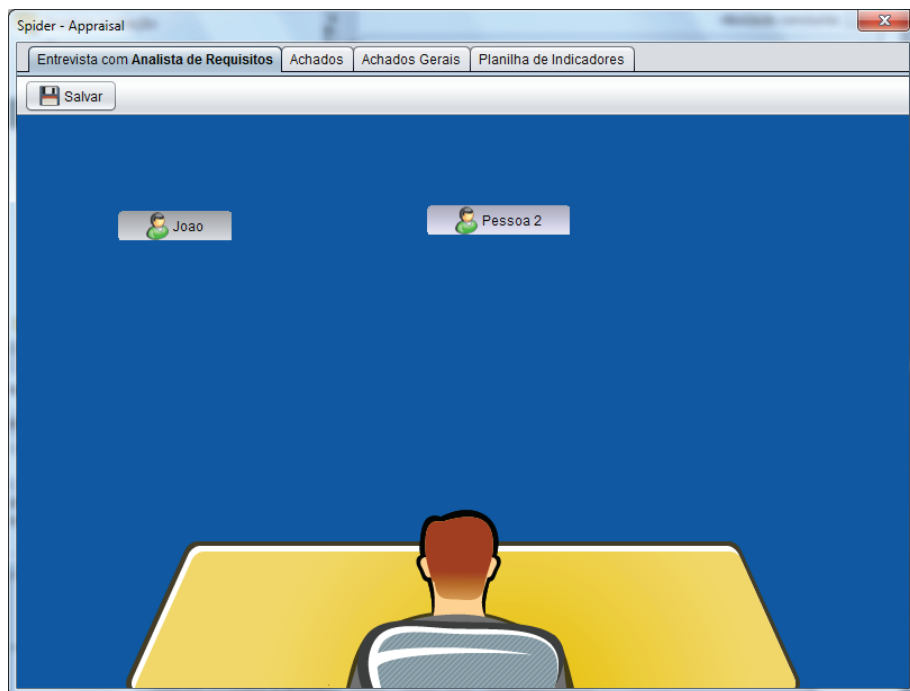


Figura 211. Tela do botão Realizar Entrevista – Aba Entrevista com “Grupo”

O exemplo da Figura 212 corresponde ao registro da entrevista com o participante “Joao”, em que devem ser preenchidas as perguntas e respostas no campo inferior da tela e, após clicar no botão “Enter” do teclado, será automaticamente armazenada a data/hora das anotações digitadas e exibidas no campo “Histórico”. Antes de fechar a janela, deve-se clicar no botão “Salvar”.

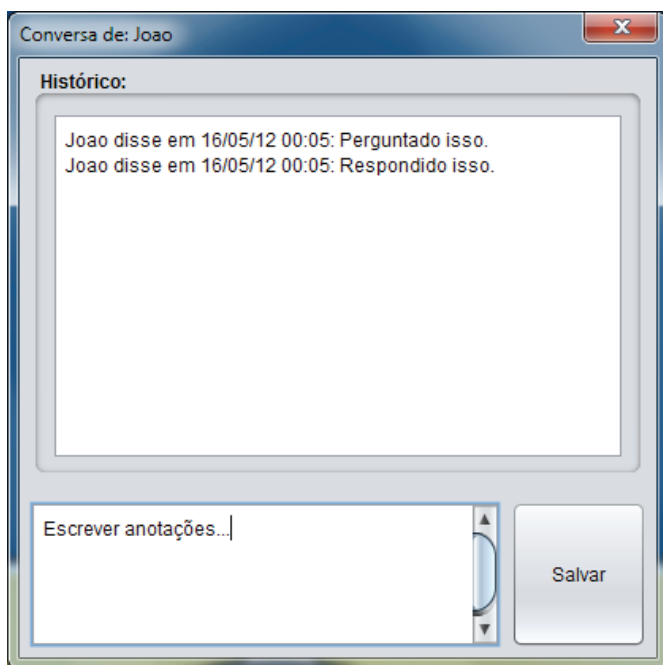
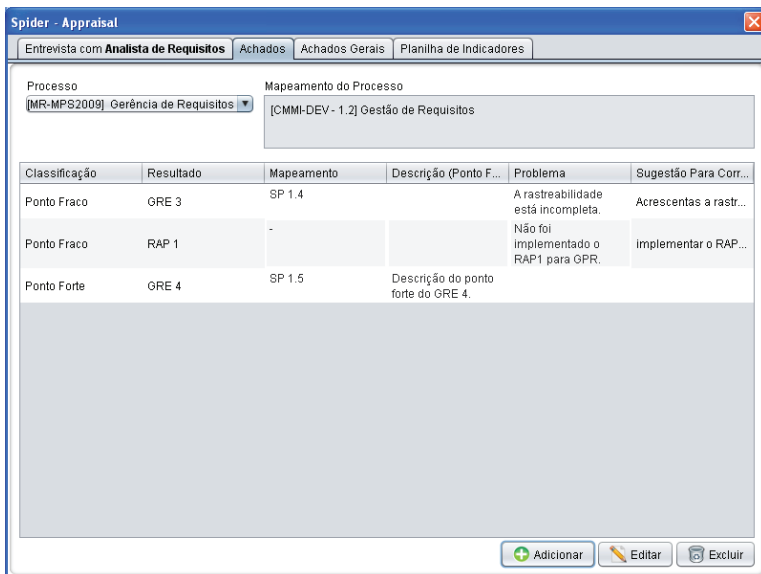


Figura 212. Tela Abrir Conversa

Na aba Achados podem ser consultados os achados cadastrados até o momento; bem como podem ser editados, excluídos e também adicionados novos achados, conforme visto na Figura 213. Além de adicionar achados dos tipos oportunidade de melhoria e ponto fraco, tal como feito na fase da pré-avaliação, podem também ser adicionados achados do tipo ponto forte.

Na aba Achados Gerais devem ser cadastrados os tipos achados gerais e achados gerais do modelo, preenchendo-se os pontos fortes, pontos fracos e oportunidades de melhoria, como ilustrado na Figura 214. Os achados gerais não estão atrelados a um determinado resultado/prática de um modelo, eles são observações gerais acerca da organização avaliada. Já os achados gerais do modelo são atribuídos ao próprio modelo de referência, de modo a poder criticá-lo e melhorá-lo continuamente.



Spider - Appraisal

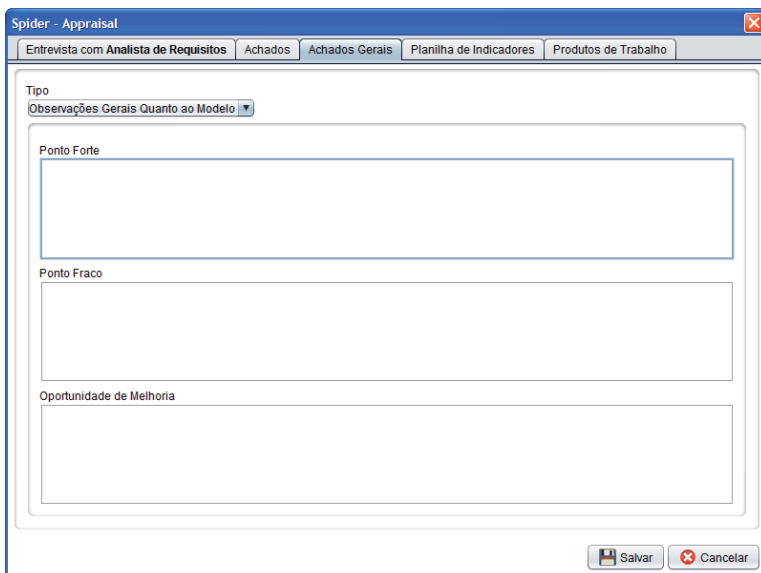
Entrevista com **Analista de Requisitos** | **Achados** | Achados Gerais | Planilha de Indicadores

Processo: **MR-MPS2009 Gerência de Requisitos** | Mapeamento do Processo: **[CMMI-DEV - 1.2] Gestão de Requisitos**

Classificação	Resultado	Mapeamento	Descrição (Ponto F...	Problema	Sugestão Para Corr...
Ponto Fraco	GRE 3	SP 1.4		A rastreabilidade está incompleta.	Acrescentar a rast...
Ponto Fraco	RAP 1	-		Não foi implementado o RAP1 para GPR.	Implementar o RAP...
Ponto Forte	GRE 4	SP 1.5	Descrição do ponto forte do GRE 4.		

+ Adicionar | ✎ Editar | 🗑 Excluir

Figura 213. Tela do botão Realizar Entrevista – Aba Achados



Spider - Appraisal

Entrevista com **Analista de Requisitos** | **Achados** | **Achados Gerais** | Planilha de Indicadores | Produtos de Trabalho

Tipo: **Observações Gerais Quanto ao Modelo**

Ponto Forte

Ponto Fraco

Oportunidade de Melhoria

📁 Salvar | ✖ Cancelar

Figura 214. Tela do botão Realizar Entrevista – Aba Achados Gerais

Na aba Planilha de Indicadores, podem ser consultadas as evidências objetivas cadastradas previamente para cada resultado esperado/prática e resultado de atributo de processo/prática genérica dos processos/áreas de processo dentro do escopo da avaliação. Nesse momento devem ser adicionadas as evidências do tipo “Afirmação”, que correspondem àquelas coletadas pelos avaliadores durante as entrevistas/apresentações/questionários feitas com os membros da organização. As informações que identificam uma evidência são: produto de trabalho, tipo da evidência, se é organizacional e os projetos associados, e uma descrição (ver Figura 215).

The image shows a software window titled "Spider - Appraisal". It contains two dropdown menus at the top: "Produto de Trabalho" and "Tipo da Evidência", with "Afirmação" selected in the second. Below these is a section labeled "Projetos" containing a table with a header row "Projeto" and a "Visualizar" button. The table body is empty. Below the table is a text area labeled "Descrição" with the placeholder text "Descrição da afirmação.". At the bottom right are two buttons: "Salvar" (with a floppy disk icon) and "Cancelar" (with a red X icon).

Figura 215. Adicionar Evidência do Tipo Afirmação

A atividade Caracterizar Práticas consiste na caracterização do grau de implementação dos resultados esperados/práticas específicas e de cada resultado esperado de atributo de processo/práticas genéricas dos processos/áreas de processo que pertencem ao escopo da avaliação. Conforme a Figura

216, abaixo da área onde estão listadas as evidências, deve-se atribuir para os resultados esperados e resultados de atributos de processos do MR-MPS-SW – para cada projeto – os valores T (totalmente), L (largamente), P (parcialmente), N (não implementado) ou NA (não avaliado) para caracterizar a implementação dos resultados esperados. O hífen simboliza que ainda não foi atribuída pontuação. Por último, deve-se elevar os valores da caracterização da implementação dos resultados esperados do nível de projeto ao nível de unidade organizacional, atribuindo-se para a coluna “Caracterização” os mesmos valores (T, L, P, N, ou NA). Já para as práticas específicas e práticas genéricas do CMMI-DEV, deve-se atribuir – para cada projeto – os valores F (*fully implemented*), L (*largely implemented*), P (*partially implemented*), N (*not implemented*) e NY (*not yet*). Do mesmo modo, deve-se elevar os valores da caracterização da implementação das práticas do nível de projeto ao nível de unidade organizacional, atribuindo-se para a coluna “Caracterização” os mesmos valores (F, L, P, N, ou NY).

Spider - Appraisal

Spider-Appraisal | Acesso | Gerar Relatórios | Ajuda

Usuário: admin | Perfil atual: Administrador

Selecionar Avaliação: Avaliação Conjunta 1 - Organização 1

Atividade concluída ☐

Caracterizar Implementação das Práticas

Processo: MR-MPS2009 Gerência de Requisitos

Mapeamento do Processo: CMMI-DEV - 1.2 Gestão de Requisitos

☒ Resultados Esperados ☐ Resultados de Atributos de Processos

GRE 1 GRE 2 GRE 3 GRE 4 GRE 5

Propósito: Os requisitos são entendidos, avaliados e aceitos junto aos fornecedores de requisitos, utilizando critérios objetivos.

Mapeamento: Sigla: SP 1.1 Prática Específica: Obter um Entendimento dos Requisitos

Evidência	Tipo	Fonte	ORG	P1	P2	Observações
Lista de Requisito...	AD	Analista de Req...				
Lista de Requisito...	AD	Analista de Req...				

Resultado	ORG	P1	P2	Caracterização
GRE 1	-	T	T	-
SP 1.1	-	F	F	-

Legenda

● Adequado ● Tem Fraqueza ● Inadequado ● Não Avaliado

Limpar Observações Salvar Cancelar

Mapeamento dos Métodos de Avaliação

MA-MPS	SCAMPI A
3.8 Caracterizar o grau de implementação de cada resultado esperado do processo e de cada resultado esperado de atributo do processo em cada projeto;	2.3.3 Document Practice Implementation;
3.9 Caracterizar, inicialmente, o grau de implementação de cada resultado esperado do processo e de cada resultado esperado de atributo do processo na unidade organizacional;	2.4.2 Characterize Implementation of Model Practices and Generate Preliminary Findings.
3.10 Caracterizar, inicialmente, o grau de implementação de cada	

Figura 216. Tela Caracterizar Práticas – Aba Caracterizar Implementação das Práticas

Quando se tratar de avaliações do modelo CMMI-DEV, deverá ser feita também a caracterização das metas, que consiste em classificar a satisfação das metas específicas e metas genéricas. Portanto, essa atividade não é realizada em avaliações do modelo MR-MPS-SW ou em avaliações integradas (conjuntas).

A tela da Figura 217 mostra a aba Caracterizar Grau de Satisfação das Metas para uma avaliação do modelo CMMI-DEV, em que na área “Metas” são exibidas as metas genéricas (GG) e as metas específicas (SG) correspondentes à área de processo “Monitoramento e Controle de Projeto”. Nesse exemplo está selecionada a meta específica “SG1”, para a qual na área “Meta” é exibida a sua descrição e no campo “Grau de Satisfação” deve ser feita a atribuição dos valores “Satisfeito”, “Não satisfeito” ou “Não avaliado”. O hífen simboliza que ainda não foi atribuída pontuação.

Na área “Práticas Relacionadas” são listadas as práticas genéricas/específicas correspondentes à meta genérica/específica selecionada e para cada prática é exibida na coluna “Caracterização” os valores que foram atribuídos durante a caracterização das práticas, na aba “Caracterizar Implementação das Práticas”. Após clicar no botão “Salvar”, o grau de satisfação é armazenado e exibido na coluna “Grau de Satisfação” da área “Metas”.

A atividade Documentar Achados Preliminares permite documentar os pontos fracos, pontos fortes e oportunidades de melhoria, que correspondem aos gaps (lacunas) dos processos implementados em relação às práticas requeridas do modelo de referência.

Na aba Achados podem ser consultados os achados cadastrados até o momento; bem como podem ser editados, excluídos e também adicionados novos achados; conforme visto na Figura 218. Além de adicionar achados dos tipos oportunidade de melhoria e ponto fraco, tal como feito na fase da pré-avaliação, podem também ser adicionados achados do tipo ponto forte.

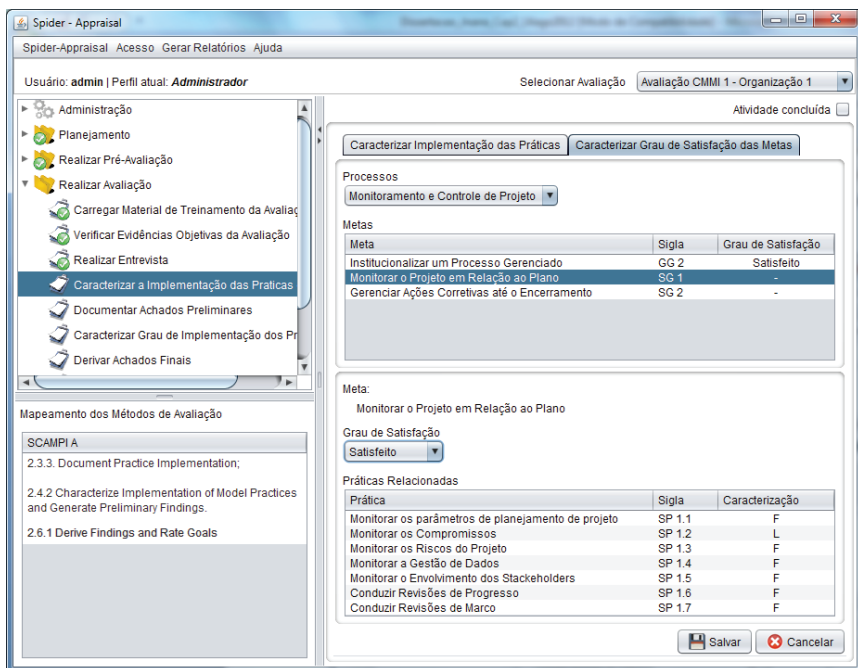


Figura 217. Tela Caracterizar Práticas – Aba Caracterizar Grau de Satisfação das Metas

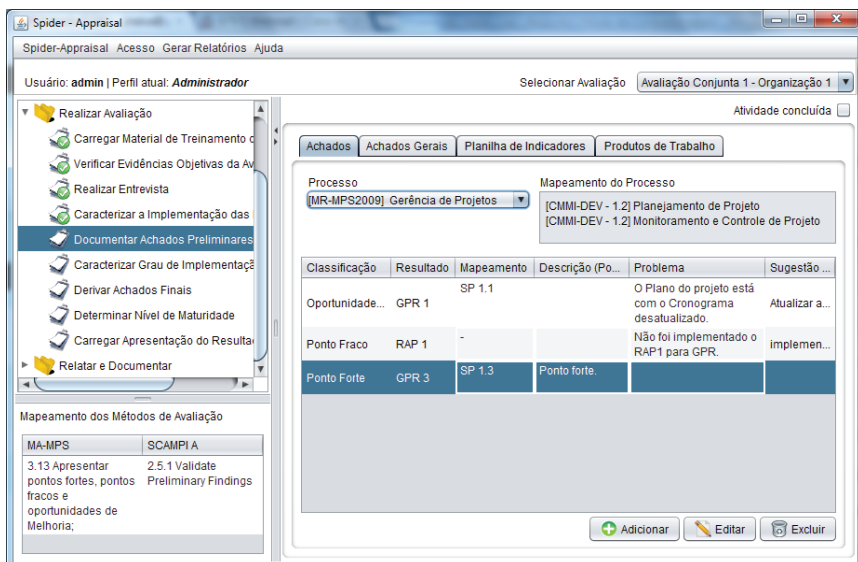


Figura 218. Tela Documentar Achados Preliminares – Aba Achados

Na aba Achados Gerais devem ser cadastrados os tipos achados gerais e achados gerais do modelo, preenchendo-se os pontos fortes, pontos fracos e oportunidades de melhoria; como ilustrado na Figura 219. Os achados gerais não estão atrelados a um determinado resultado/prática de um modelo, eles são observações gerais acerca da organização avaliada. Já os achados gerais do modelo são pontos fortes, pontos fracos ou oportunidades de melhoria atribuídos ao próprio modelo de referência, de modo a poder criticá-lo e melhorá-lo continuamente.

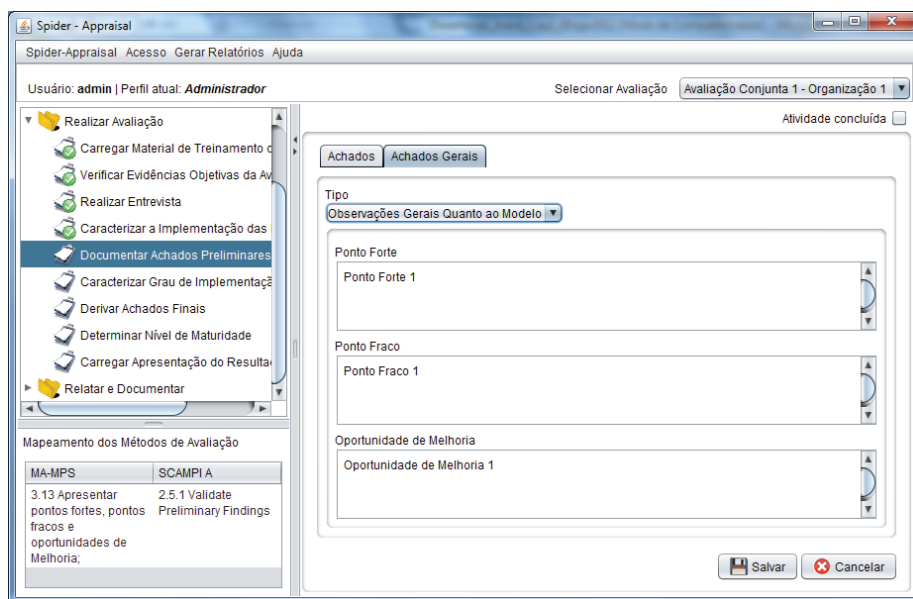


Figura 219. Tela Documentar Achados Preliminares – Aba Achados Gerais

A atividade Caracterizar Processos visa a caracterizar o grau de satisfação dos processos/áreas de processo através da atribuição de pontuação.

Na aba Caracterizar Grau de Satisfação dos Processos (ver Figura 220), na área “Processos” estão listados os processos/áreas de processo que pertencem ao escopo da avaliação. Em se tratando de avaliação estagiada, para cada processo/área de processo selecionado devem ser atribuídos ao campo “Grau de Satisfação” os valores “Satisfeito”, “Não Satisfeito”, “Não Avaliado” ou “Não

Aplicável”. Caso seja uma avaliação contínua do CMMI-DEV, as áreas de processo serão classificadas quanto ao seu nível de capacidade, atribuindo-se valores de 0 a 3. O hífen simboliza que ainda não foi atribuída pontuação.

Essa classificação deve ser feita com base no grau de satisfação alcançado pelos resultados/práticas de cada processo/área de processo, que estão listados na área “Indicadores”. Também pode ser consultada a aba Planilha de Indicadores para ver em detalhes as evidências objetivas registradas para as práticas/resultados esperados dos modelos.

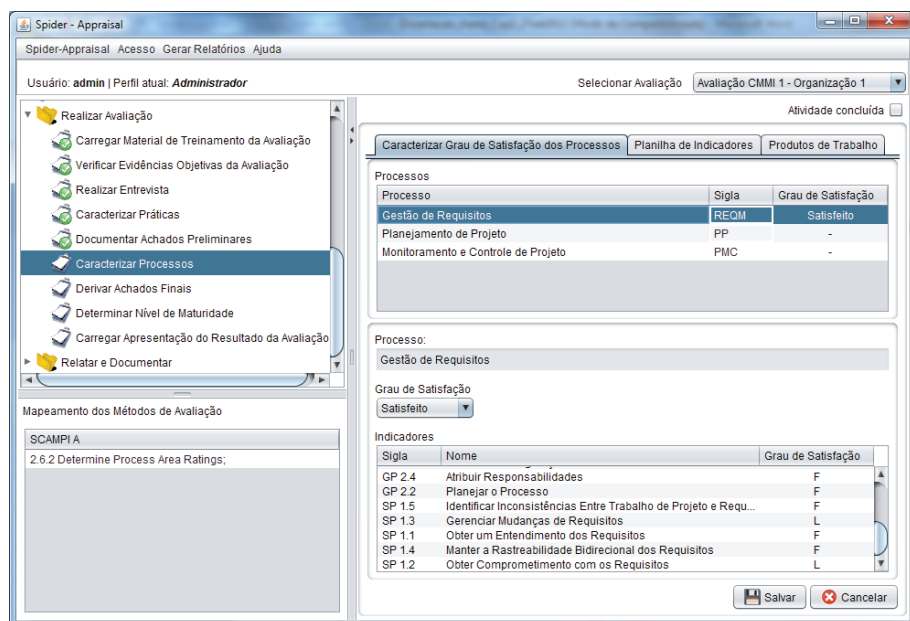


Figura 220. Tela Caracterizar Processos – Aba Caracterizar Grau de Satisfação dos Processos

Na atividade Derivar Achados Finais são derivados os achados finais da avaliação, a partir da validação dos achados preliminares feita junto aos membros da organização. Como visto na tela da Figura 221, estão disponíveis as abas Achados, Planilha de Indicadores, Produtos de Trabalho, Grau de Satisfação dos Processos, e Grau de Satisfação das Metas (apenas para o

CMMI-DEV). Basicamente o que podem ocorrer são modificações nos achados, mas também pode haver alterações na planilha de indicadores (alteração ou inclusão de novas evidências), nos produtos de trabalho, e até mesmo ser alterada a caracterização do grau de implementação dos processos, caso a equipe de avaliação julgue pertinente. Do mesmo modo pode ser modificado o grau de satisfação das metas, mas apenas quando se tratar de avaliação do CMMI-DEV.

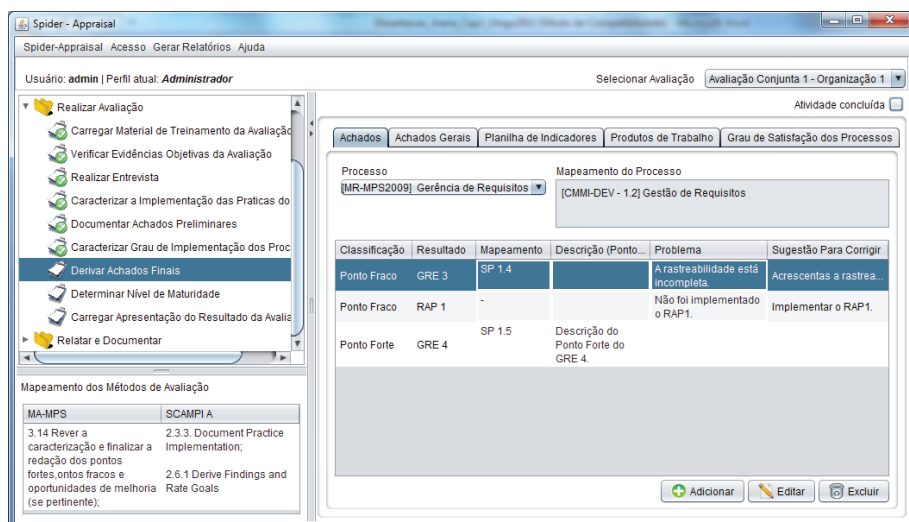


Figura 221. Tela Derivar Achados Finais – Aba Achados

Na atividade Determinar Nível de Maturidade deve ser feita a atribuição do nível de maturidade à unidade organizacional, de acordo com a caracterização da implementação dos processos/áreas de processo contemplados na avaliação. Ela acontece tanto na avaliação estagiada, como na avaliação contínua do CMMI-DEV, entretanto, na avaliação contínua também deve ser realizada a atividade “Determinar Perfil de Capacidade”, que é discutida no tópico adiante.

Conforme a Figura 222, a tela mostra os campos “Modelo de Referência Utilizado” (que contém o nome do modelo de referência utilizado e sua

versão, sendo que na avaliação conjunta aparecerá o nome dos dois modelos) e o “Nível Alvo de Maturidade” (que é o nível almejado, definido durante a fase de planejamento). Para o campo “Nível de Maturidade Avaliado” devem ser atribuídos os valores de “G” a “A” para o modelo MR-MPS-SW ou de 1 a 5 para o modelo CMMI-DEV. Quando se tratar de avaliação conjunta, o nível de maturidade será atribuído de acordo com os valores do CMMI-DEV e ao lado aparecerá a correspondência com os níveis de maturidade do MR-MPS-SW. Vale observar que a determinação do nível de maturidade deve considerar o grau de satisfação alcançado por cada processo/área de processo avaliado, o qual pode ser visualizado na coluna “Grau de Satisfação” da área “Processos Avaliados”.

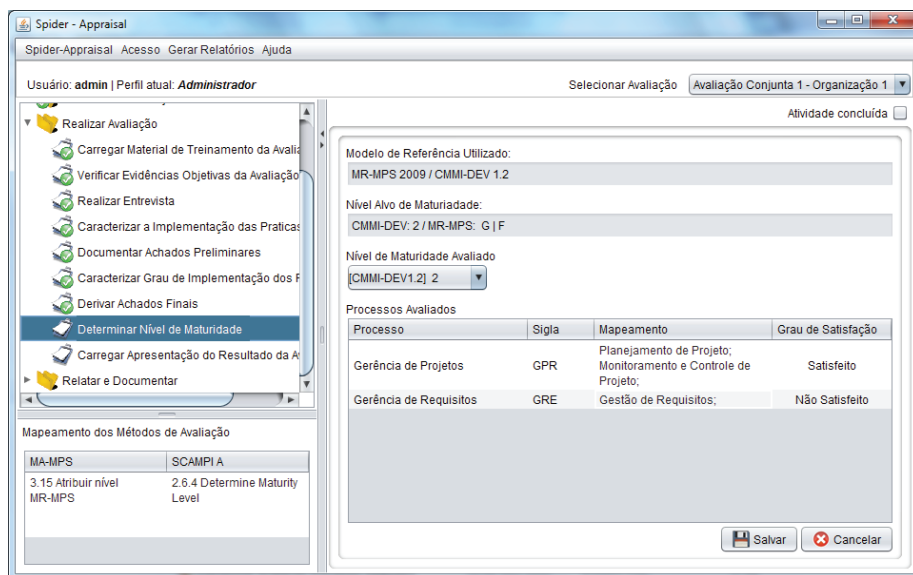


Figura 222. Tela Determinar Nível de Maturidade

A atividade Determinar Perfil de Capacidade determina o perfil de capacidade da organização avaliada, de acordo com a caracterização das metas dos processos/áreas de processo contemplados na avaliação, e ocorre apenas em avaliações contínuas do CMMI-DEV.

Conforme Figura 223, no topo da tela é exibido o nome do modelo de referência utilizado e sua versão. Na área “Processos Avaliados” são exibidos para cada processo: o nome do processo, sua sigla, o nível alvo de capacidade (que é o nível almejado, definido durante a fase de planejamento), o nível de capacidade avaliado (que é o nível de fato alcançado), e o grau de satisfação. Para determinar a capacidade, deve-se escolher um processo na área “Processos Avaliados” e, então, será exibido na área “Processo” o nome do processo e o campo “Nível de Capacidade” (para o qual deve ser atribuído os valores de 1 a 5 ou “Não Avaliado”). Vale observar que a determinação do perfil de capacidade deve considerar a caracterização das metas específicas e metas genéricas de cada processo/área de processo avaliado, as quais são listadas na área “Metas Relacionadas”.

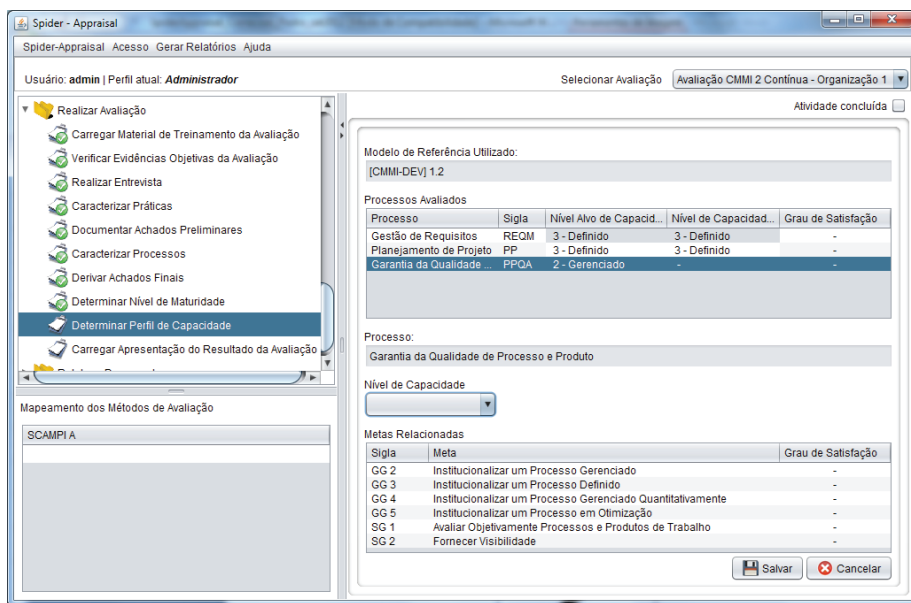


Figura 223. Tela Determinar Perfil de Capacidade

A atividade Carregar Apresentação do Resultado da Avaliação possibilita carregar o material de apresentação utilizado pela equipe de avaliação para comunicar o resultado da avaliação aos colaboradores da unidade

organizacional, sendo que normalmente o material é no formato de apresentação de slides. Ao clicar no botão “Carregar”, será aberta uma janela para pesquisar o arquivo no computador e, então, o arquivo selecionado será armazenado na ferramenta. A qualquer momento o material de apresentação poderá ser consultado pelos interessados, clicando-se no botão “Baixar” (ver Figura 224).

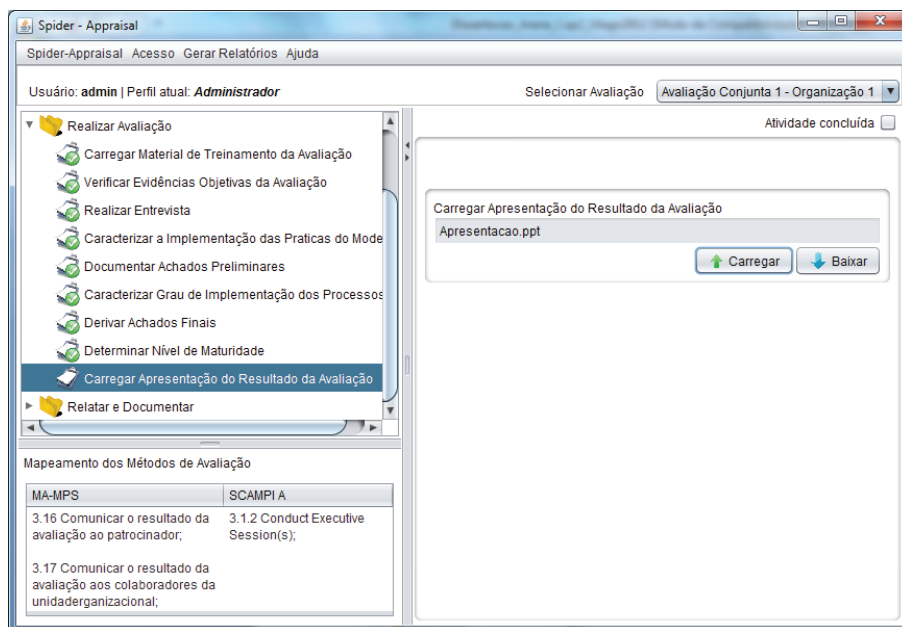


Figura 224. Tela Carregar Apresentação do Resultado da Avaliação

Basicamente a fase **Relatar e Documentar** tem por objetivo a documentação dos resultados da avaliação e a comunicação desses resultados aos interessados, além de incluir atividades de auditoria (quando houver) e coleta de lições aprendidas. A seguir são explicadas as atividades dessa fase que são sistematizadas pela ferramenta Spider-Appraisal.

Na atividade Documentar Resultados da Avaliação é gerado o relatório final e o resultado da avaliação, contemplando todos os dados relativos à avaliação e os resultados produzidos, para fins de relatório.

Conforme visto na Figura 225, na área “Documento” estão listados o Relatório da Avaliação e o Resultado da Avaliação, sendo que na avaliação conjunta são gerados relatórios separados para o MR-MPS-SW e para o CMMI-DEV. Para visualizar um relatório, basta clicar no botão “Baixar” e, então, será gerado automaticamente o relatório em formato PDF.

O Relatório de Avaliação é gerado a partir dos dados preenchidos no plano de avaliação e da pontuação contida na planilha de indicadores dos resultados/práticas por processo/área de processo; incluindo os achados (pontos fortes, fracos e oportunidades de melhoria), o grau de satisfação dos processos/áreas de processo, e o nível de maturidade/perfil de capacidade alcançado pela unidade organizacional. Trata-se de um relatório extenso e detalhado acerca das informações coletadas e produzidas ao longo da avaliação.

Já o Resultado da Avaliação é um relatório mais sintetizado, contendo as seguintes informações: os dados da organização avaliada, equipe de avaliação, o modelo e método de avaliação, os processos/áreas de processo do escopo da avaliação, e o grau de satisfação de cada processo/área de processo avaliado.

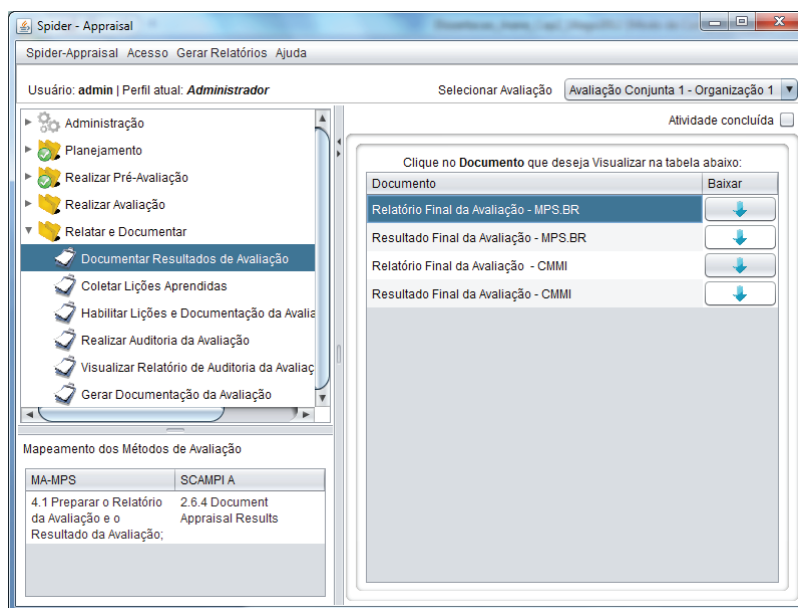


Figura 225. Tela Documentar Resultados da Avaliação

A atividade Coletar Lições Aprendidas permite que os envolvidos na avaliação façam o julgamento da execução da avaliação e o registro de lições aprendidas. Conforme a Figura 226, cada participante (patrocinador, avaliador líder, equipe de avaliação, coordenador da IA, implementador, etc.) deve preencher um formulário com questões acerca da realização da avaliação, a fim de registrar as lições aprendidas.

The screenshot shows the 'Spider - Appraisal' application window. The title bar includes 'Spider-Appraisal', 'Acesso', 'Gerar Relatórios', and 'Ajuda'. The user is logged in as 'admin' with the role of 'Administrador'. The 'Selecionar Avaliação' dropdown is set to 'Avaliação Conjunta 1 - Organização 1'. The 'Atividade concluída' checkbox is unchecked. The 'Patrocinador' tab is selected among other roles like 'Líder', 'Coordenador da IA', 'Implementador', and 'Equipe de Avaliação'.

The main content area contains a message from the 'Senhor Patrocinador:' and two numbered questions for the sponsor to answer:

Para que possamos melhorar continuamente o programa de melhoria, contamos com suas opiniões e sugestões, respondendo às questões seguintes e entregando este formulário para o avaliador líder, dentro de um envelope lacrado e rubricado. Garantimos total confidencialidade sobre seus comentários.

1) A empresa ficou satisfeita com a Equipe de Avaliação?

2) Comente o relacionamento com a Instituição Avaliadora.

The bottom section, 'Mapeamento dos Métodos de Avaliação', contains a table with two columns: 'MA-MPS' and 'SCAMPI A'.

MA-MPS	SCAMPI A
3.19 Avaliar a execução da avaliação pelo patrocinador;	3.2.1 Collect Lessons Learned;
3.20 Avaliar a execução da avaliação pela equipe de avaliação;	

Figura 226. Tela Coletar Lições Aprendidas – Aba Patrocinador

A atividade Habilitar Lições e Documentação da Avaliação para o Auditor tem por finalidade habilitar as lições aprendidas e a documentação da avaliação para o auditor designado. Como mostrado na Figura 227, devem ser selecionados os documentos que deverão estar disponíveis para o auditor e, então, clicar no botão “Salvar” e marcar a atividade como “Atividade concluída”. Após isso, o Auditor poderá executar a atividade “Realizar Auditoria da Avaliação”, em que poderá acessar e visualizar apenas os documentos que foram habilitados. Importante ressaltar que essa e outras atividades de auditoria

só aparecerão no menu da ferramenta se no cadastro da avaliação tenha sido designado um auditor para a avaliação, caso contrário essa atividade será omitida do menu e não será executada.

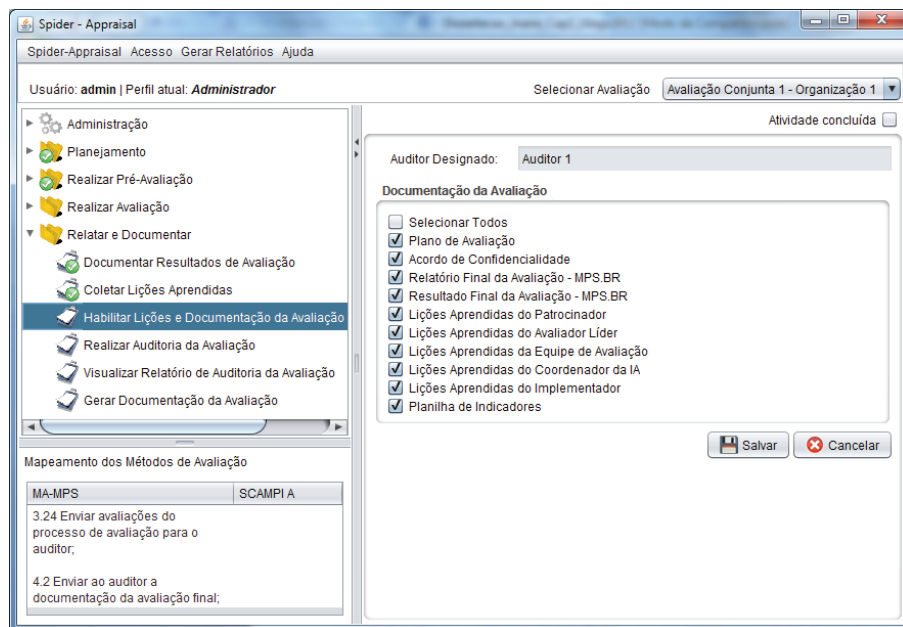


Figura 227. Tela Habilitar Lições e Documentação da Avaliação para o Auditor

Na atividade Realizar Auditoria da Avaliação o auditor realizará a auditoria do trabalho realizado pela equipe de avaliação durante toda a avaliação. Como mostrado na Figura 228, o auditor poderá acessar os documentos que foram habilitados para a auditoria, listados na área “Documentos da Pré-Avaliação” e em “Documentos da Avaliação”, clicando-se em cima do nome do documento (Plano de Avaliação, Acordo de Confidencialidade, etc.) para que este seja aberto para consulta. Após a análise, o auditor deverá indicar se o documento está completo (sim ou não) e correto (sim ou não). O relatório da auditoria também possui os seguintes campos a serem preenchidos: descrição detalhada de todos os problemas encontrados; itens que necessitam de correção, apreciação sobre o resultado da avaliação (não há dúvidas ou há dúvidas);

e, por fim, o parecer (aprovado, aprovado com modificações, ou não aprovado). Após preencher e clicar no botão “Salvar”, é possível gerar o relatório em formato PDF através do menu “Gerar Relatórios”.

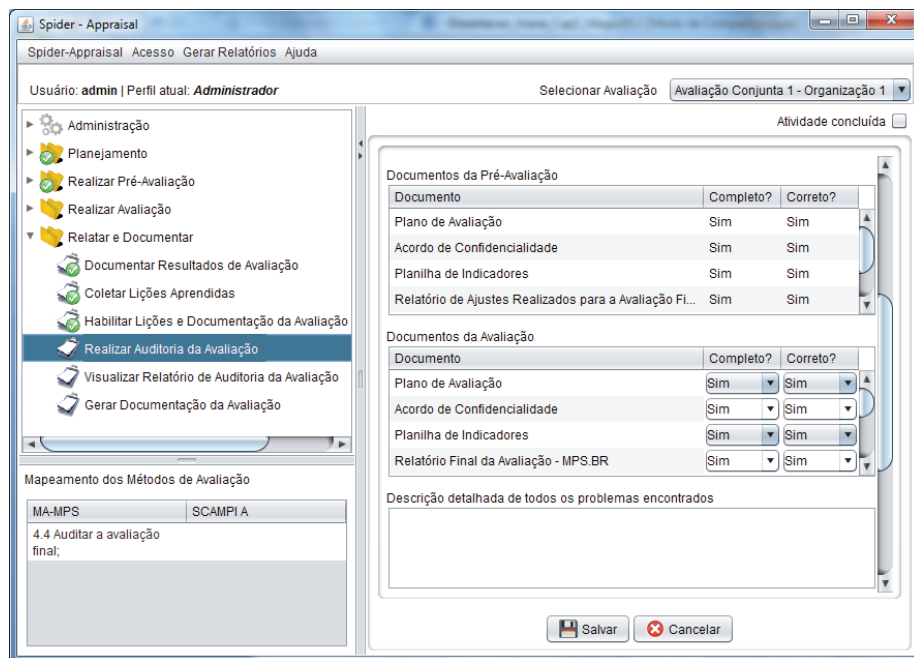


Figura 228. Tela Realizar Auditoria da Avaliação

Na atividade Visualizar Relatório de Auditoria da Avaliação os avaliadores tem acesso ao relatório de auditoria da avaliação que foi preenchido pelo auditor, no qual podem consultar todos os problemas encontrados na documentação da avaliação, os itens que necessitam de correção, bem como o parecer final. Se o parecer tiver sido “aprovado”, nenhuma correção significativa deverá ser efetuada. Caso seja “aprovado com modificações”, os avaliadores terão que realizar as correções apontadas pelo auditor. E se o parecer for “não aprovado”, foram detectadas inconsistências e irregularidades que não podem ser reparadas e, portanto, a avaliação deverá ser cancelada. A Figura 229 apresenta o relatório de auditoria, o qual pode ser acessado apenas para

visualização e, sendo assim, a funcionalidade de edição está desabilitada para os avaliadores. É possível gerar o relatório em formato PDF através do menu “Gerar Relatórios”.

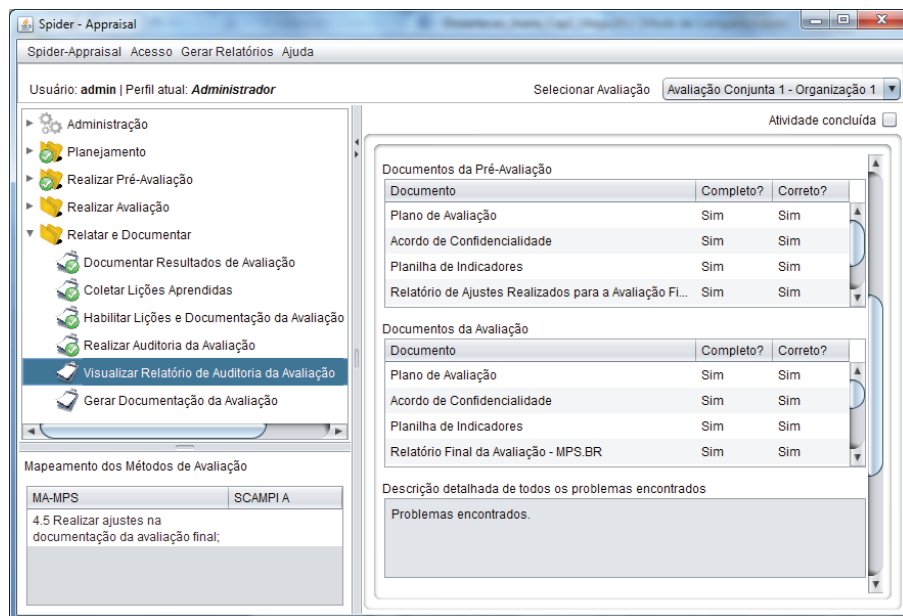


Figura 229. Tela Visualizar Relatório de Auditoria da Avaliação

A atividade Gerar Documentação da Avaliação tem por finalidade gerar a documentação final e os resultados da avaliação para que possam ser consultados e entregues aos interessados.

Conforme visto na Figura 230, na área “Documento” estão listados todos os documentos e resultados produzidos durante a avaliação, sendo que na avaliação conjunta são gerados relatórios separados para o MR-MPS-SW e para o CMMI-DEV. Para visualizar um relatório, basta clicar no botão “Baixar” e, então, será gerado automaticamente o relatório em formato PDF.

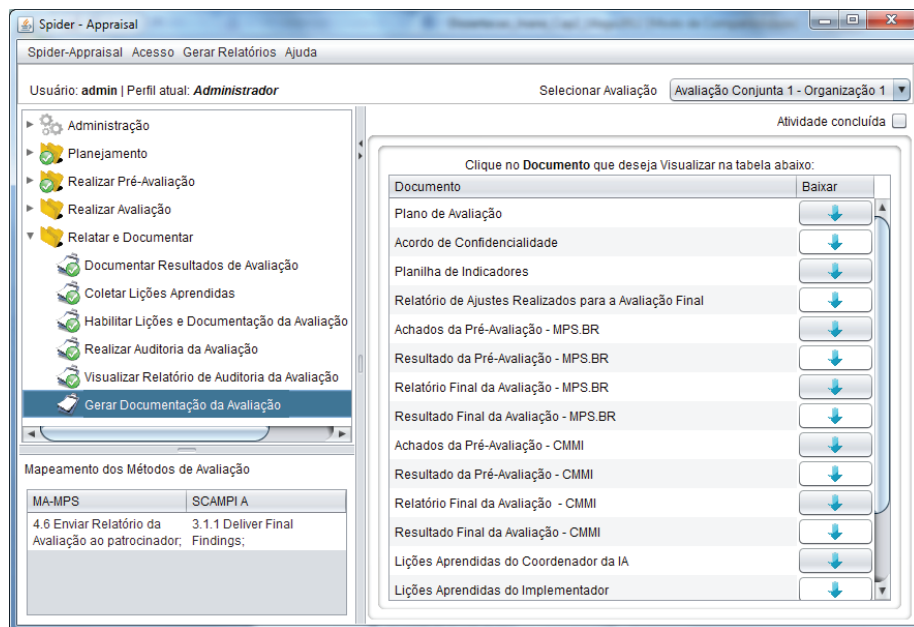


Figura 230. Tela Gerar Documentação da Avaliação



Capítulo 22

Abordagens para Rastreabilidade de Requisitos

em [Malcher e Oliveira, 2014], [Malcher et al., 2015]

Paulo Robson Campelo Malcher
Sandro Ronaldo Bezerra Oliveira

Com o objetivo de tornar as organizações mais competitivas nos negócios, as respostas das Tecnologias de Informação têm a necessidade de serem cada vez mais rápidas, porém a qualidade do que é entregue, ou influencia essa entrega, é também um fator muito importante para manter essa competitividade [Ferreira, 2010]. Nesse contexto, a Engenharia de Requisitos (ER), segundo Pressman (2011), surgiu a fim de ajudar os engenheiros de software a compreender melhor o problema por meio de um conjunto de tarefas que levam a um entendimento de qual será o impacto do sistema sobre o negócio, do que os *stakeholders* necessitam e como será a interação dos usuários finais com o sistema. Dentre as atividades que compõem a engenharia de requisitos, pode-se dividi-las em seis áreas genéricas: Levantamento, Análise, Documentação, Verificação, Validação e o Gerenciamento dos Requisitos [Kotonya e Sommerville, 1998].

Este trabalho contempla a área de gerenciamento de requisitos, que abrange toda a compreensão e o controle de mudanças nos requisitos de sistema que é tida como essencial para o sucesso de projetos de software e está presente em padrões e modelos de qualidade, como a Norma ISO/IEC 12207 e os modelos CMMI-DEV e MR-MPS-SW. A Rastreabilidade de Requisitos é

uma das práticas que compõem a Gerência de Requisitos (GRE) [SEI, 2010] e é tida como uma tarefa indispensável nesse processo e um fator de qualidade no que diz respeito ao desenvolvimento de software. Para Gotel e Finkelstein (1997), ela é definida como a capacidade de descrever e seguir o ciclo de vida de um requisito em ambas as direções, da origem ou da implementação, passando entre todas as especificações relacionadas.

Outro ponto a ser destacado neste trabalho é a utilização do método de Revisão Sistemática da Literatura (RSL), que é um dos principais métodos da Engenharia de Software Baseada em Evidências (*Evidence-based Software Engineering* – EBSE) e é classificado como estudo secundário, já que depende dos estudos primários utilizados para revelar evidências e construir conhecimento [Dybå *et al.*, 2007; Travassos, 2007 *apud* Costa, 2010]. Costa (2010) ainda afirma que a essência do paradigma baseado em evidência é coletar e analisar sistematicamente todos os dados disponíveis sobre determinado fenômeno para obter uma perspectiva mais completa e mais ampla do que se pode captar por meio de um estudo individual.

Nesse sentido, realizar uma RSL que tenha como foco descobrir quais as abordagens descritas na literatura para auxiliar a rastreabilidade de requisitos, pode ser de grande valia para um maior conhecimento sobre a área. Nesse sentido, o objetivo desse trabalho é apresentar abordagens (técnicas, modelos e ferramentas) encontradas na literatura, por meio da revisão sistemática, para o apoio à rastreabilidade de requisitos e realizar uma análise quanto a sua implementação no contexto do MR-MPS-SW. Vale mencionar que nem todas as abordagens são apresentadas nesta seção, outras podem ser encontradas em [Malcher, 2015].

22.1. A Revisão Sistemática da Literatura

A revisão sistemática segue uma metodologia específica e peculiar, apresentada por Kitchenham (2007), onde define que revisões sistemáticas começam

pela definição de um protocolo de revisão e resume as etapas de uma revisão sistemática em três fases principais: planejamento, condução e apresentação. Nesta seção serão apresentadas definições referentes à fase de planejamento. O protocolo na íntegra pode ser encontrado em http://spider.ufpa.br/projetos/spider_rastreabilidade/SPIDER_ProtocoloRevisao.pdf.

22.1.1. Objetivo da Revisão

A revisão sistemática realizada teve o objetivo de identificar abordagens para apoiar a atividade de Rastreabilidade de Requisitos no contexto de projetos de software, no período de 2003 a 2013, no que tange a processos, modelos, metodologias, técnicas, ferramentas e afins. Desta forma, tem-se a seguinte estrutura para o objetivo, conforme proposto em Santos (2010):

- **Analisar:** relatos de experiência e publicações científicas através de um estudo baseado em revisão sistemática;
- **Com o propósito de:** identificar abordagens para apoiar atividades de Rastreabilidade de Requisitos;
- **Com relação à:** definição e uso de processos, frameworks, metodologias, ferramentas e demais instrumentos empregados para a implantação e execução das atividades de Rastreabilidade de Requisitos em organizações de desenvolvimento de software;
- **Do ponto de vista:** de pesquisadores e organizações desenvolvedoras/mantenedoras de software;
- **No contexto:** acadêmico e industrial.

22.1.2. Questões de Pesquisa

Com base no objetivo de investigação desta revisão, foi definida a seguinte questão de pesquisa, a qual guiou esta Revisão Sistemática da Literatura:

(Q1) Quais as abordagens existentes para apoiar as atividades de Rastreabilidade de Requisitos?

22.1.3. Seleção de Fontes

Com base em critérios de seleção pré-estabelecidos e nas restrições da pesquisa, foram selecionadas as seguintes fontes de pesquisa onde foram realizadas as buscas dos estudos primários, de acordo com a disponibilidade de pesquisa e a relevância para área de Qualidade de Software: IEEEExplore Digital Library; El Compendex; ISI Web of Knowledge; ACM; Scopus; Anais do WAMPS – Workshop Anual do MPS; Anais do WER – Workshop de Engenharia de Requisitos; Anais do SBQS – Simpósio Brasileiro de Qualidade de Software.

Vale ressaltar que outras definições realizadas durante a fase do planejamento foram omitidas desse trabalho por não estarem diretamente ligadas aos objetivos do mesmo, tais como: a definição de *strings* de busca; os critérios de inclusão e exclusão dos estudos; os critérios de avaliação de qualidade; e a definição dos processos de seleção, avaliação e extração dos resultados da revisão.

22.2. Extração de Dados

Em acordo com o que foi definido no planejamento da revisão, no que diz respeito à busca realizada nas fontes selecionadas, esta foi executada em cada fonte com uma *string* de busca específica. A partir disso, obteve-se um total de 1509 estudos, no qual: 433 trabalhos foram identificados na IEEE; 437 na El Compedex; 124 na Scopus; 316 na ACM; 175 na ISI Web Knowledge; 6 no SBQS; 9 no WAMPS; e, por fim, 9 foram identificados no WER. Como pode

ser visualizado no gráfico da Figura 231(a). E na Figura 231(b) a distribuição das abordagens.

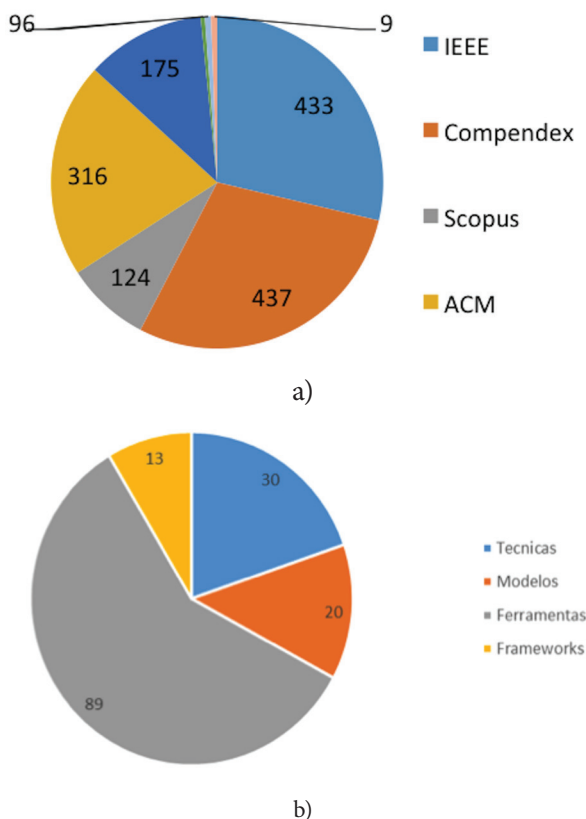


Figura 231. (a) Distribuição dos Estudos Retornados por Fonte de Pesquisa; (b) Distribuição de Estudos Retornados por Tipo de Abordagens

A partir do processo de seleção definido no protocolo de revisão, foram selecionados os estudos primários por meio da leitura do título, palavras-chaves, resumo e conclusão de cada estudo retornado, e em seguida foram aplicados os critérios de inclusão e exclusão. Após o processo de seleção, o número de estudos foi bastante reduzido, sendo selecionados 411 estudos. A evolução em números do processo de seleção de estudos primários é apresentada na Tabela 21.

Tabela 21. Evolução dos Estudos Primários após o Processo de Seleção.

Seleção de Estudos Primários				
Fontes	Estudos Retornados	Estudos Excluídos		Estudos Incluídos
		Duplicados	Outros	
IEEE	433	0	269	164
El Compedex	437	228	99	110
Scopus	124	95	13	16
ACM	316	61	182	73
ISI Web Knowledge	175	110	41	24
SBQS	6	0	0	6
WAMPS	9	0	0	9
WER	9	0	0	9
Total	1509	0	0	411

Após a seleção dos estudos primários, estes foram lidos em sua totalidade a fim de identificar abordagens que possam auxiliar a rastreabilidade de requisitos. Nesse intuito, foram extraídas as principais abordagens a serem definidas nas subseções a seguir.

22.2.1. Técnicas

Falbo (1998) diferencia método e técnica definindo que um método é um procedimento sistemático para a realização de uma ou mais atividades, definindo passos e heurísticas, já uma técnica é também um procedimento para a realização de uma atividade, contudo, menos rígido e detalhado.

Dentre os estudos selecionados foram identificadas cerca de 30 técnicas de apoio à rastreabilidade de requisitos disponíveis na literatura, onde é possível dividi-las em técnicas de visualização de rastreabilidade e técnicas de geração de *links* de rastreabilidade. Dentre as técnicas de geração de *links* de rastreabilidade, pode-se dividi-las em técnicas manuais, semiautomáticas e automáticas. O que diferenciam essas técnicas é a forma com que os *links* de rastreabilidade são gerados, por exemplo: nas técnicas ditas manuais, os *links* são gerados diretamente pelos interessados; já nas semiautomáticas, os

links de rastreabilidade são geradas de forma automática, porém necessitam da aprovação dos interessados; por fim, as técnicas automáticas são as que não necessitam de fatores externos para a geração dos *links* de rastreabilidade. Por questão de espaço, foram escolhidas as técnicas com mais ocorrência na literatura, segundo a revisão realizada, para serem apresentadas nesse trabalho.

a) Traceability Matrix

A técnica de matriz de rastreabilidade é a que pode ser definida como de visualização e que pode usar tanto as técnicas de geração de *link* de rastreabilidade da forma manual, semiautomática e automática, porém pode-se verificar que na maioria dos estudos é definida de forma manual, mesmo com o auxílio de algum ferramental em sua implementação.

No resultado da Revisão Sistemática da Literatura realizada foi a técnica tida como manual mais citada nos estudos. Esta técnica utiliza uma matriz para gerar a rastreabilidade bidirecional de forma vertical ou horizontal. Uma limitação desta técnica está no fato de relacionar os artefatos sempre em pares, o que dificulta uma visão mais ampla, quando se analisa o impacto de mudanças, além de ser uma técnica onerosa em sua implementação e manutenção. Hayes *et al.* (2005) afirma que as Matrizes além de auxiliar nas tarefas referentes à rastreabilidade de requisitos, elas também podem auxiliar na análise de completude do software e em seu teste. Li e Hayes (2013) corroboram com a afirmação de Hayes quando afirma que é uma técnica fundamental para muitas técnicas de segurança, tais como a análise de criticidade, a análise de impacto de mudanças e testes de regressão.

b) Graphs

Grafos são mais uma forma de visualização de rastreabilidade, que também podem ser definidas na literatura como árvores, que geralmente fazem o uso da geração manual de links. Entretanto, existem trabalhos como os de Chen

(2010) e de Schwarz *et al.* (2009) que apresentam a geração automática de grafos. A ideia dessa técnica é utilizar os artefatos a serem relacionados como os vértices do grafo e as arestas como *links*, nesse caso podendo ou não acrescentar a dependência dos mesmos por meio da direção das arestas.

c) Vector Space Model (VSM)

Este tipo de técnica faz parte da área denominada Recuperação de Informação (*Information Retrieval* – IR), que é um método que segundo Narayan *et al.* (2011) possui técnicas que têm se mostrado muito eficazes para auxiliar a geração automática de *links* de rastreabilidade, reduzindo o tempo necessário para gerar o mapeamento de rastreabilidade. Em particular, vários pesquisadores têm aplicado recentemente Técnicas de Recuperação de Informação [Deerwester *et al.*, 1990; Harman, 1992 *apud* De Lucia, 2007] para o problema da recuperação de *links* de rastreabilidade entre artefatos de diferentes tipos.

Essa técnica apresenta sua utilidade para a rastreabilidade, dividindo os itens de rastreabilidade em um modelo de espaço vetorial. VSM é uma abordagem padrão que calcula a similaridade do cosseno entre uma consulta e um documento, cada um dos quais é representado como um vetor de termos ponderados [Gibiec *et al.*, 2010]. Hayes *et al.* (2003) utilizam o modelo de espaço vetorial para recuperar *links* de rastreabilidade entre os requisitos e comparar os resultados obtidos através da aplicação de diferentes variantes do modelo básico: em particular, o uso de palavras-chave do documento é mostrado como um meio para melhorar os resultados do modelo de espaço vetorial.

d) Latent Semantic Information (LSI)

LSI é uma técnica de recuperação de informação, proposta pela primeira vez no trabalho de Deerwester *et al.* (1990), que é baseada no modelo de espaço vetorial (VSM), onde assume-se que existe uma estrutura subjacente ou latente no uso da palavra para cada conjunto de documentos. LSI utiliza técnicas

estatísticas para estimar essa estrutura latente. A descrição dos termos e documentos com base na estrutura semântica latente subjacente é usada para representar e recuperar informações. Esta representação pode ser usada para comparar e indexar documentos usando uma variedade de medidas de similaridade. Ao aplicar LSI ao código-fonte e sua documentação interna associada (ou seja, comentários), componentes candidatos podem ser comparados com relação a essas medidas de similaridade. Por ser uma técnica de IR baseada em VSM, a técnica de LSI é uma técnica muito citada na literatura no que diz respeito a técnicas automáticas como nos trabalhos de De Lucia (2008), Jiang *et al.* (2007), e Maletic e Marcus (2001).

e) Probabilistic Approaches

Nesse tipo de técnica são utilizados modelos de redes probabilísticas para realizar a recuperação de informação. Esses modelos, segundo Gibiec *et al.* (2010), têm sido demonstrados por meio de experimentação prévia para executar de forma equivalente a outros modelos de recuperação de informação. Seu uso envolve três atividades de busca principais que são: o pré-processamento do texto; a probabilidade de *links*; e a análise dos resultados. Antoniol *et al.* (2002) afirma que um modelo probabilístico IR calcula a probabilidade de que dois documentos estão relacionados e usa a probabilidade calculada para classificar links gerados. *Jensen-Shannon Model* (JSM) foi proposto por Abadi *et al.* (2008), e é impulsionado por uma abordagem e hipótese técnica de teste probabilístico. JSM representa cada um documento por meio de uma distribuição de probabilidades, isto é, uma matriz de termo-a-documento normalizado.

22.2.2. Modelos

Como resultados da extração de dados da revisão sistemática da literatura, foram identificados cerca de 20 modelos que envolvem a rastreabilidade de

requisitos. Alguns deles focados na rastreabilidade de requisitos e outros de uma forma mais ampla envolvendo a gerência de requisitos.

a) Traceability Information Model (TIM)

É um modelo básico sobre rastreabilidade, muito referenciado na literatura por ser uma visão conceitual de um projeto de implantação de uma estratégia de rastreabilidade. É definido como um modelo genérico que pode fazer uso de diferentes abordagens em sua implementação, utilizando-se da UML para sua implantação. Basicamente consiste na definição de tipos de entidades, de artefatos e das relações de rastreabilidade entre esses artefatos. Ele também define quais tipos de artefatos que podem ser rastreados, qual o tipo de relação entre esses artefatos e o tipo de relação de rastreabilidade. O Centro de Excelência para Rastreabilidade de Software [CoEST, 2015] afirma que o uso deste modelo é considerado como uma das melhores práticas para a gestão e planejamento de rastreabilidade em todo o ciclo de vida de um projeto, que pode ser enfatizado nos trabalhos de Mäder *et al.* (2009), Cleand-Huang *et al.* (2012) e Mäder *et al.* (2013).

b) Ramesh's Traceability Model

Este modelo foi proposto por Ramesh e Jarke (2001), sendo um modelo de informação de rastreabilidade (TIM), que descreve toda a estratégia da prática de rastreabilidade em uma organização. Este modelo identifica a informação de rastreabilidade, os requisitos, os raciocínios, a alocação de requisitos e de recursos. Como características relevantes deste modelo destacam-se: o relacionamento entre requisitos e suas derivações; entre estes com os documentos de origem; entre requisitos e documentos de testes; entre requisitos e componentes do sistema; entre requisitos e *stakeholders*; dentre outros. O modelo permite representar as solicitações de mudanças, porém não comporta

a representação dos impactos gerados e não possibilita representar versionamento e configuração de granularidade.

22.2.3. Ferramentas

Foram encontradas na literatura cerca de 90 ferramentas que possuem alguma relação com rastreabilidade de requisitos. As ferramentas encontradas implementam diferentes técnicas com relação à geração de links e da visualização da rastreabilidade em diversos contextos, alcançando a pré e pós rastreabilidade, a rastreabilidade horizontal e vertical, as formas para frente e para trás, e por fim a bidirecionalidade da rastreabilidade.

a) REquirements TRacing On target (RETRO)

É uma ferramenta desenvolvida no trabalho de Hayes *et al.* (2006), que utiliza a técnica de geração de *link* de rastreabilidade automática por meio de algumas técnicas de Recuperação de Informação. A mais citada na literatura é a técnica de Modelo de Espaço Vetorial (VSM), o que resulta na geração automática de matrizes de rastreabilidade para a visualização da prática. Hayes realizou um estudo de caso na ferramenta o que comprovou melhor aproveitamento em seu uso quando comparada a forma manual de realizar a rastreabilidade.

b) DOORS

Telelogic DOORS [DOORS, 2005] é uma ferramenta comercial para o gerenciamento de requisitos que suporta vários tipos de objetos de requisitos, tipos de atributos e a geração de *links* de rastreabilidade. DOORS utiliza uma matriz de rastreabilidade e árvores com *hiperlinks* para a visualização e a manutenção dos *links* de rastreabilidade. Pode-se definir a ferramenta,

no contexto da rastreabilidade, como uma ferramenta que utiliza a técnica de geração de *links* de rastreabilidade de forma semiautomática, pois as relações são identificadas manualmente, mas as ligações entre os elementos que são relacionados são geradas automaticamente pela ferramenta com base nessas relações.

c) Caliber RM

CaliberRM (2015) é uma ferramenta para o gerenciamento de requisitos, desenvolvida pela Borland. A ferramenta mantém as informações de rastreabilidade entre os vários requisitos e também entre artefatos em outras ferramentas. Ela também oferece uma série de maneiras de manter uma visão geral de todos os requisitos e combina esses recursos para analisar o impacto das propostas de mudança. Ela também permite que os requisitos de software conectem-se a uma variedade de artefatos de todo o ciclo de vida. A ferramenta utiliza a técnica manual para geração de *link* de rastreabilidade e para a visualização dos *links* da rastreabilidade utiliza matriz de rastreabilidade e grafos.

d) Poirot

Poirot [Lin *et al.*, 2006] é uma ferramenta baseada na web de apoio à rastreabilidade. Utiliza um modelo de rede probabilística como técnica de geração automática de links de rastreabilidade, ou seja, utiliza uma das técnicas de recuperação de informação. Provê a rastreabilidade entre os requisitos, elementos de *design*, código e outros artefatos armazenados na ferramenta. Utiliza como forma de visualização da rastreabilidade tanto um formato textual quanto um visual, mas especificamente por *hyperlinks*. Em muitos trabalhos, a exemplo dos trabalhos de Cleland-Huang *et al.* (2007) e Nair *et al.* (2013), a ferramenta sempre é citada como referência quando se trata da utilização de abordagens probabilísticas para o apoio à rastreabilidade.

e) Rational RequisitePro

É uma ferramenta para o gerenciamento de requisitos desenvolvida pela IBM, que fornece a possibilidade de realizar o relacionamento entre requisitos, casos de uso e casos de teste. Além do que, quando ocorre uma mudança em um requisito ela identifica os artefatos de software que são afetados. Um exemplo de aplicação desenvolvida para a utilização dessa matriz pode ser encontrado no RequisitePro [IBM, 2010]. No contexto da rastreabilidade de requisitos, ela é reconhecida por diversos autores da área como uma ferramenta de apoio, por exemplo nos trabalhos de Cleland-Huang *et al.* (2002; 2007) e Hayes *et al.* (2003; 2005).

22.3. Indicação das Abordagens para Implementar o MR-MPS-SW

Como já apresentado, a prática da rastreabilidade de requisitos no MR-MPS-SW está inserida no processo de GRE do nível de maturidade G. Dentre os resultados esperados desse processo, os que tem ligação direta com a rastreabilidade são os GRE3, GRE4, GRE5. Esses resultados são apresentados nas subseções seguintes, assim como, a indicação das abordagens encontradas na literatura que melhor podem satisfazer sua implementação.

22.3.1. GRE 3 - A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida

Este resultado segundo a Softex (2012a), indica a necessidade de se estabelecer um mecanismo que permita rastrear a dependência entre os requisitos e os produtos de trabalho, e que a rastreabilidade bidirecional deve acontecer tanto de forma horizontal quanto vertical. Nesse resultado, dois conceitos

principais em rastreabilidade são citados para se alcançar este resultado: o fato de se criar um mecanismo de rastreio; e que a bidirecionalidade deve acontecer tanto de forma horizontal quanto vertical.

No contexto das abordagens encontradas e apresentadas neste trabalho, quanto a definição da forma horizontal e vertical da rastreabilidade, pode-se melhor utilizar qualquer técnica de geração de *links* que fazem da recuperação de informação, e quanto à visualização da rastreabilidade, a matriz de rastreabilidade ainda se mostra mais satisfatória para uma visão mais ampla sobre a rastreabilidade. Quanto aos modelos apresentados, estes auxiliam no estabelecimento da estratégia de rastreabilidade, que está fortemente relacionada à satisfação deste resultado. Com relação ao apoio ferramental disponível para o atendimento deste resultado, pode-se destacar a RETRO.

22.3.2. GRE 4 - Revisões em planos e produtos de trabalho do projeto são realizadas visando a identificar e corrigir inconsistências em relação aos requisitos

Este resultado segundo a Softex (2012a), define que a consistência entre os requisitos e os produtos de trabalho do projeto deve ser avaliada e os problemas identificados devem ser corrigidos. Nesse sentido, o auxílio que a rastreabilidade pode proporcionar à implementação deste resultado está na definição do relacionamento entre requisitos e outros produtos de trabalho, bem como ela pode auxiliar na identificação da consistência dessas relações.

Com base nesse auxílio que a rastreabilidade pode prover e nas abordagens encontradas e apresentadas nesse trabalho, pode-se verificar que com base nas revisões realizadas na rastreabilidade e em seus *links* pode-se por meio da visualização desses *links* identificar as inconsistências e prover suas correções. Uma técnica que possibilita uma melhor visualização desse tipo é a que envolve a implementação de grafos. Quanto às ferramentas, podem-se citar a DOORS e CaliberRM.

22.3.3. GRE 5 - Mudanças nos requisitos são gerenciadas ao longo do projeto

Este resultado segundo a Softex (2012a), afirma que requisitos adicionais podem ser incorporados no projeto, requisitos podem ser retirados do projeto e/ou mudanças podem ser feitas nos requisitos já existentes. Desta forma, é necessário tratar essas mudanças e analisar o seu impacto. Com relação a mudanças que podem ocorrer, atualizações devem ser feitas na estrutura de rastreabilidade estabelecida, o que irá diferir entre a forma de geração de *links* de rastreabilidade está na maneira como a mesma acontece: se for de forma manual, deve-se ser feita essa mudança também de forma manual; no caso da utilização de forma semiautomáticas ou automáticas, o impacto geralmente é identificada automaticamente. Um bom exemplo para o auxílio nesse resultado esperado é a ferramenta RequisitePro, que alerta quando uma mudança ocorre e qual o seu impacto.



Capítulo 23

Apoio à Execução do Processo de Software

em [Silva et al., 2012a], [Silva et al., 2012b], [Portela et al., 2012], [Portela et al., 2014]

Carlos dos Santos Portela

Antônio André Cunha da Silva

Elder de Jesus Ferreira da Silva

Mariano Gomes

Maurício Ronny de Almeida Souza

Wallace Michel Pinto Lira

Sandro Ronaldo Bezerra Oliveira

Alexandre Marcos Lins de Vasconcelos

O constante uso de produtos computacionais na sociedade contemporânea fez com que sistemas de computadores se tornassem indispensáveis no dia a dia. No entanto, conforme as tecnologias evoluem, o nível de exigência dos usuários também aumenta, refletindo no crescimento da complexidade dos requisitos do software [SBC, 2006]. Sendo assim, desenvolver software acaba sendo uma atividade complexa, pois para que o produto final desenvolvido esteja dentro do esperado pelo usuário é necessário que o desenvolvimento siga um escopo bem definido e atenda aos prazos, custos e recursos estimados no início do projeto [Oliveira, 2007].

Com o objetivo de aumentar a qualidade de software para melhor atender às necessidades do cliente, a Engenharia de Software tem produzido ferramentas que auxiliam o desenvolvimento de software e controlam o processo de desenvolvimento [Reis, 2003]. Neste contexto, destaca-se a área de

Tecnologia de Processos de Software, que envolve a construção de ambientes para modelagem, execução, simulação e evolução de processos de desenvolvimento de software. Contudo, observou-se que apenas automatizar o desenvolvimento não é suficiente para atingir todos os critérios de qualidade necessários no desenvolvimento de software [Bertollo, 2006].

Segundo Oliveira (2005), apesar da qualidade do produto ser algo bom, a qualidade do processo de produção é também importante, pois pode-se dizer mais sobre a qualidade de um software observando como o software foi desenvolvido (processo) ao invés de levar em consideração apenas o produto final (software). Diante desta constatação, surgiram importantes mecanismos de avaliação e certificação da qualidade de software, tais como o modelo internacional de qualidade de software CMMI-DEV e o modelo nacional MR-MPS-SW. Essas abordagens tiveram grande aceitação no mercado e passaram a ser consideradas ferramentas para avaliar o nível da qualidade das organizações desenvolvedoras de software [Oliveira, 2005].

Sendo assim, este trabalho apresenta uma ferramenta de apoio à execução flexível de processos de software que seja aderente aos perfis de capacidade dos modelos de qualidade CMMI-DEV e MR-MPS-SW. Esta ferramenta foi construída a partir do *framework* desenvolvido em [Portela, 2012] para auxiliar à execução de processos de software. De acordo com Oliveira (2011), um *framework* é uma customização de um processo que apresenta um fluxo de atividades genéricas necessárias para a execução de qualquer processo de software. Assim, a ferramenta apresentada neste trabalho, que possui o mesmo nome do *framework*, Spider-PE, reflete em suas funcionalidades as boas práticas definidas nos modelos de qualidade.

23.1. A Concepção Técnica da Ferramenta Spider-PE

A partir do *framework* Spider-PE, desenvolveu-se uma solução de software que implementa de forma semi-automatizada e flexível o fluxo de atividades

que tem como objetivo a melhoria da qualidade do processo . O software em questão é a ferramenta Spider-PE (*Process Enactment*), desenvolvida no âmbito do Projeto SPIDER sob licença GPL.

A ferramenta Spider-PE foi programada na linguagem de programação JAVA como uma aplicação para *Desktop*. Para sua correta configuração é necessário a instalação do JAVA JDK 6.0 ou superior, do SGBD (Sistema Gerenciador de Banco de Dados) MySQL 5.5 ou superior, além de ser necessário o espaço disponível em disco de 150 MB e no mínimo 512 MB de memória na máquina que executará a ferramenta.

Com o intuito de reutilizar e evitar o desenvolvimento de funcionalidades específicas, foram integradas à ferramenta Spider-PE outras ferramentas de software livre. Para apoio à gestão de mudança é usado o SVN (Subversion), sistema que adota a Licença Apache de software livre, disponível em <http://subversion.apache.org>. Na ferramenta Spider-PE, utiliza-se o SVN para gerenciar a evolução do projeto através do controle e acesso de versões dos produtos de trabalho. Também se utiliza o Redmine, ferramenta de licença GPL que, no contexto das funcionalidades desta ferramenta, cadastra, monitora e acompanha a solução de possíveis problemas que possam surgir durante a execução do projeto. Esta ferramenta encontra-se disponível em <http://www.redmine.org>.

No apoio às estimativas de processo, adotam-se diversas ferramentas de licença GPL desenvolvidas pelo Projeto SPIDER: Spider-APF, que permite realizar estimativas utilizando o método de análise por pontos de função; Spider-UCP, utilizada para medição de tamanho de software a partir do método Pontos por Caso de Uso (*Use Case Points* – UCP); e a Spider-CoCoMo, que provê apoio à estimativa de prazo, esforço e tamanho de equipe utilizando o método CoCoMo - *Constructive Cost Model*.

A avaliação de processo é feita através da ferramenta Spider-CL, que também adota a licença GPL. Esta ferramenta objetiva o desenvolvimento de *checklists*, o uso destes em diferentes contextos e o registro destes usos. No contexto da Spider-PE, utiliza-se a Spider-CL como forma de avaliar os

produtos de trabalho gerados e as áreas do processo através de critérios objetivos (descritos em *checklists* de avaliação).

No apoio ao processo de medição, utiliza-se a Spider-MsControl, licença GPL, que possibilita a definição, coleta, análise e acompanhamento de medidas. Na Spider-PE, esta ferramenta é utilizada na definição de métodos para monitorar e controlar as atividades que estão sendo desenvolvidas durante o projeto, a fim de garantir que a realização destas ocorra conforme o planejado.

As ferramentas desenvolvidas pelo Projeto SPIDER encontram-se disponíveis em <http://spider.ufpa.br>. O código-fonte da ferramenta Spider-PE encontra-se disponível em <https://github.com/projetospider/Spider-PE>. O executável da ferramenta Spider-PE encontra-se disponível em <http://www.mediafire.com/download/22weqbaxcwf2g6/Spider-PE.jar>.

Neste capítulo serão descritos quais são os objetivos da ferramenta Spider-PE. Na próxima seção, o fluxo de atividades implementado pela ferramenta, oriundo do *framework* Spider-PE, é descrito com detalhes. Posteriormente, são elucidados os conceitos da Máquina de Estados aplicada à ferramenta, esta Máquina de Estados é um dos componentes mais importantes da ferramenta, pois define regras de comportamento ao processo na fase de execução do processo. Por fim, a interface da ferramenta e suas funcionalidades são apresentadas através de capturas de tela.

23.1.1. Objetivos

Os objetivos da ferramenta Spider-PE estão ligados à adoção de modelos de melhoria, desenvolvimento da ferramenta sob licença de software livre e a aplicação de um formalismo segundo o padrão SPEM [OMG, 2008].

Para atingir esses objetivos, primeiramente a ferramenta busca executar processos de maneira flexível e semi-automatizada conforme as boas práticas descritas nos modelos de qualidade CMMI-DEV e MR-MPS-SW. É possível evidenciar o atendimento do Nível F do MR-MPS-SW e Nível 2 do

CMMI-DEV a partir das funcionalidades da ferramenta Spider-PE associadas a uma metodologia de uso [Silva *et al.*, 2012a]. Para atender estas boas práticas dos modelos de qualidade, a ferramenta implementa um fluxo de atividades genéricas descritas no framework de execução Spider-PE [Portela, 2012], além disso, a ferramenta Spider-PE usa uma máquina de execução responsável por implementar o processo conforme o formalismo descrito na linguagem xSPIDER_ML [Portela *et al.*, 2012], definida segundo as notações do modelo SPEM.

A utilização de tecnologias de software livre para realizar tarefas de melhoria da qualidade do processo dentro da Spider-PE é outro objetivo da ferramenta. Este objetivo, também é um dos objetivos gerais do Projeto SPIDER, que se justifica pelo aumento crescente de esforços para a criação de softwares não-proprietários que visam o atendimento das necessidades comuns de organizações e indivíduos. Conforme descrito anteriormente, a ferramenta foi desenvolvida utilizando a linguagem de programação livre Java e demais tecnologias de software livre. Além disso, durante seu desenvolvimento, foram utilizadas as ferramentas de apoio também de caráter livre, IDE Eclipse Juno e MySQL Workbench 5.0 para a visualização da base de dados. Desta forma, através da Spider-PE, empresas e organizações possuem uma alternativa consistente às tecnologias de execução de processo proprietárias.

A ferramenta também objetiva ser compatível com outras ferramentas desenvolvidas no contexto do Projeto SPIDER. Uma dessas ferramentas é a Spider-PM. A partir dessa ferramenta é possível modelar e definir um modelo de processo de software segundo as notações da linguagem SPIDER_ML. Este modelo de processo pode ser salvo e importado para a Spider-PE. Desta maneira, o processo modelado poderá ser executado e monitorado de maneira semi-automatizada pela ferramenta. Adicionalmente, a Spider-PE também objetiva a integração com diversas outras ferramentas construídas no âmbito do Projeto SPIDER, como a Spider-APF; Spider-UCP; Spider-CoCoMo; Spider-CL e Spider-MsControl.

23.1.2. Fluxo de Atividades

Conforme mencionado anteriormente, a ferramenta Spider-PE é baseada no fluxo de atividades definido no *framework* homônimo. As atividades definidas neste *framework* devem ser customizadas de acordo com o perfil e peculiaridades das organizações que irão utilizá-lo [Portela, 2012]. Através da ferramenta apresentada neste trabalho, o fluxo descrito no *framework* Spider-PE pode ser implementado de maneira mais adequada dentro das organizações.

O conjunto de fases e atividades deste *framework* foram baseadas no mapeamento dos níveis de capacidade dos modelos MR-MPS-SW e CMMI-DEV [Silva *et al.*, 2012a]. Além disso, este fluxo de atividades obedece o formalismo da linguagem xSPIDER_ML [Portela *et al.*, 2012], aderente ao modelo SPDM 2.0. O fluxo de atividades definidos no *framework* foi sistematizado e dividido em três fases: Gerenciamento do Processo, Execução das Atividades do Processo e Aplicação do Formalismo de Execução. Cada uma destas fases é composta por atividades que descrevem os objetivos, artefatos de entrada e saída, procedimentos, ferramentas de apoio, dentre outras características que atendem aos requisitos propostos nos modelos e normas de qualidade.

a) Fase de Gerenciamento do Processo

A primeira fase é o Gerenciamento do Processo. Esta fase é subdividida em 3 (três) etapas: Planejamento, Execução e Monitoramento, como pode ser observado na Figura 232.

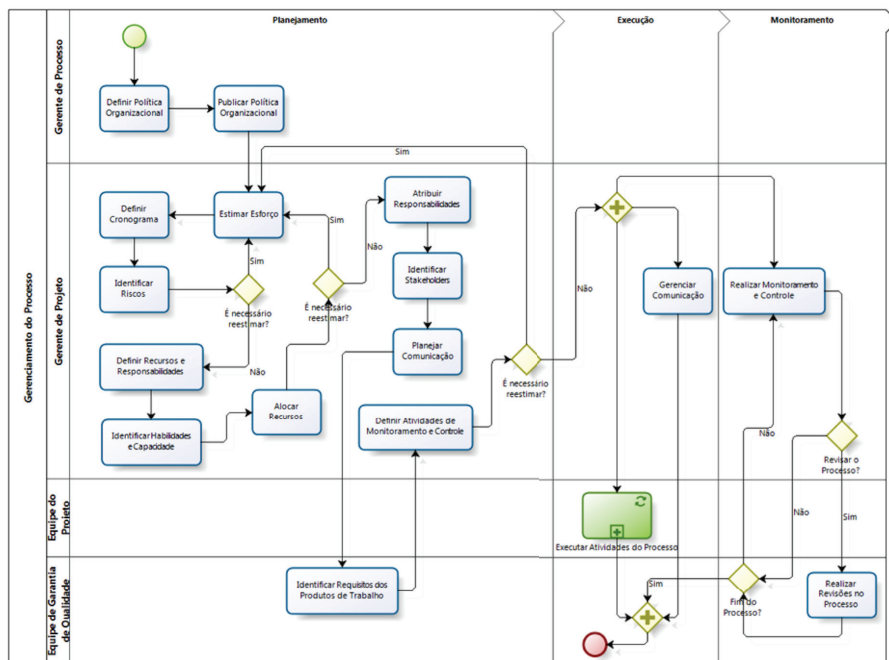


Figura 232. Fase de Gerenciamento de Processo.

O objetivo principal dessa fase é o planejamento e acompanhamento da execução do processo segundo as recomendações dos padrões de qualidade CMMI-DEV e MR-MPS-SW. Nesta fase, é realizada a definição da política do processo, a estimativa de recursos e cronograma, alocação de recursos humanos, dentre outras atividades. Além disso, esta fase é composta pela sub-fase Execução das Atividades do Processo, destacada em cor verde na Figura 232.

Primeiramente, o Gerente do Projeto, o responsável por acompanhar a execução de processo no módulo de Gerência do Processo da ferramenta Spider-PE, irá definir e publicar uma política organizacional para o processo. Além disso, o gerente de projeto fica responsável pela estimativa de esforço necessária para executar as atividades do processo. Através do cálculo desse esforço é possível definir o cronograma do processo e posteriormente identificar os riscos. Dependendo do impacto do risco, o Gerente de Projeto poderá

adotar medidas de mudança e re-estimar o esforço do processo. É importante destacar que apesar do Gerente de Projeto ser o principal responsável pela execução das atividades nesta fase, estas deverão ser realizadas juntamente com a Equipe do projeto, pois os seus membros é quem de fato irão executar de as atividades definidas do projeto.

Posteriormente, o Gerente de Projeto irá definir os recursos (hardware e software) e as responsabilidades do projeto. O Gerente irá analisar as habilidades necessárias para realizar as atividades do processo e a capacidade dos membros da Equipe. Desta maneira, é possível definir as responsabilidades durante todo o processo, ou seja, quais são os perfis de recursos humanos qualificados para executar as atividades de maneira mais adequada. Nesta etapa, também, verifica-se a necessidade de realizar treinamentos para a capacitação da Equipe envolvida. Com base na identificação dos recursos e com a definição da responsabilidade do projeto é possível realizar e alocação destes recursos, ou seja, definir quais recursos serão utilizados em cada atividade. Como a alocação está associada a capacidade da Equipe e a disponibilidade de recursos, poderá existir a necessidade de re-estimativa do esforço para a realização do Projeto. Em seguida, é realizada a delegação das responsabilidades aos membros da Equipe.

O próximo passo do Gerente de Projeto é identificar os interessados (*stakeholders*) do projeto, ou seja, os indivíduos que de alguma forma serão afetados pelo projeto. Esta identificação deve ser feita para a elaboração do Plano de Comunicação. Este plano de comunicação será um guia da maneira de como e onde as informações relevantes serão disponibilizadas para os interessados identificados.

Após e realização do plano de comunicação, a Equipe de Garantia da Qualidade deverá identificar os requisitos dos produtos de trabalho, com base na definição de padrões, *templates* e demais atributos. Por fim, o Gerente de Projeto deverá definir quais e como serão as atividades de monitoramento e controle do processo, com o objetivo de que o processo seja executado conforme o planejamento.

Depois do término da Etapa de Planejamento, a Etapa de Execução é iniciada. Nesta segunda etapa, o gerente deverá gerenciar a comunicação de acordo com o plano pré-estabelecido. Este gerenciamento deverá ocorrer paralelamente durante todo o andamento do processo. A Equipe do Projeto ficará responsável pela execução das atividades definidas no processo. Esta atividade é caracterizada como uma subfase, essencial para o processo e é descrita com detalhes a seguir.

Finalmente, a última etapa realizada é a de Monitoramento. Nesta etapa, o Gerente de Projeto é responsável pela identificação, registro e monitoramento dos problemas que poderão surgir durante o andamento do processo. O Gerente de Projeto também fica responsável pelo acompanhamento para a solução destes problemas. Nesta mesma etapa, a Equipe de Garantia de Qualidade é responsável por revisões no processo que visam a aderência do processo conforme especificado inicialmente, caso necessário, a Equipe deverá tratar as possíveis não conformidades encontradas.

b) Fase de Execução das Atividades do Processo

A fase de execução das atividades do processo é apresentada na Figura 233. Nesta fase, é onde a execução do processo propriamente dita é realizada. As atividades descritas no modelo de processo deverão ser executadas pelos seus respectivos responsáveis e os produtos de trabalho necessários deverão ser gerados conforme especificado na fase de Gerenciamento de Processo. Esta fase também é derivada das boas práticas descritas nos modelos CMMI-DEV e MR-MPS-SW.

A primeira etapa dessa fase é uma subfase de Aplicação do Formalismo de Execução, que será descrito a seguir. Esta subfase está destacada em verde na Figura 232. A ferramenta Spider-PE é de suma importância para a realização desta etapa, uma vez que somente através do uso desta ferramenta é que o formalismo poderá ser usado corretamente. É importante ressaltar que esta subfase deve ser feita paralelamente as demais atividades da fase de Execução

das Atividades do Processo. Desta maneira, o formalismo proposto pode funcionar sobre as atividades do processo.

Uma dessas atividades adicionais que devem ser realizadas paralelamente a Fase de Aplicação do Formalismo é geração de produtos de trabalho necessários para a realização das atividades. A acessibilidade e versionamento de cada um desses produtos de trabalho também deve ser realizados. Além disso, a Equipe do Projeto deverá verificar a aderência do processo aos padrões e requisitos.

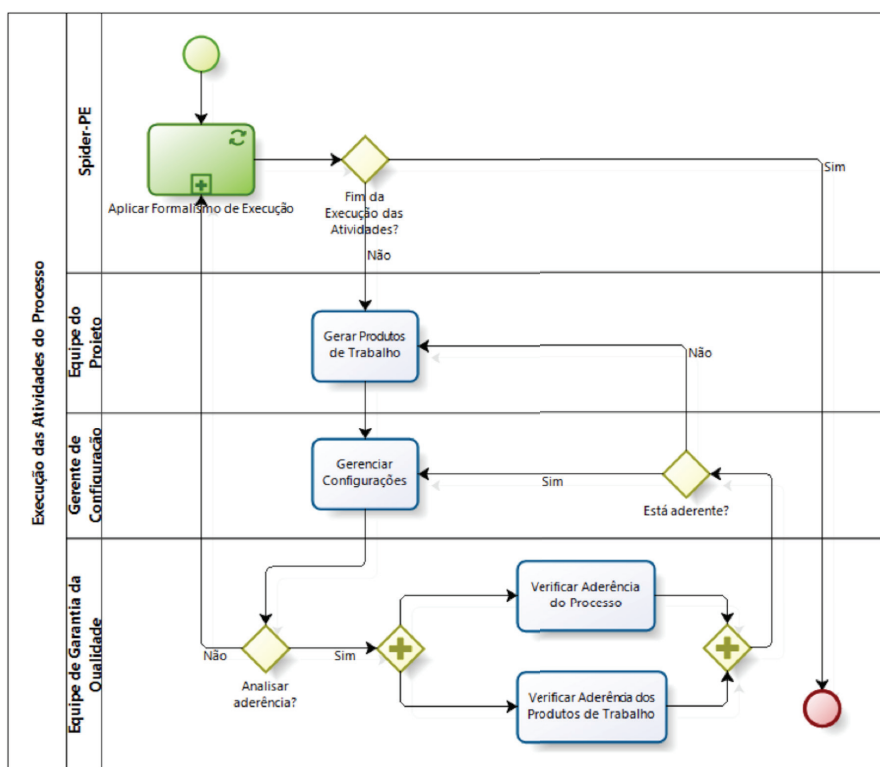


Figura 233. Fase de Execução das Atividades do Processo.

A Equipe de Garantia da Qualidade deverá verificar se os produtos de trabalhos gerados estão condizentes aos padrões estabelecidos na Fase de

Gerenciamento de Processo. No caso de alguma inconsistência, o produto de trabalho deverá ser gerado novamente. Para isso, a Equipe de Garantia da Qualidade deverá trabalhar junto à Equipe do Projeto, pois são estes que irão criar os produtos de trabalhos a serem verificados.

c) Fase de Aplicação do Formalismo de Execução

O objetivo dessa fase é a implementação das regras do formalismo descritos na linguagem xSPIDER_ML [Portela *et al.*, 2012] de forma que as atividades referidas nessa linguagem possam atender aos conceitos de flexibilidade e execução semi-automatizada. Esta implementação é feita através da instanciação dessas atividades em uma máquina de execução e de apoio da ferramenta apresentada neste trabalho. É importante frisar que as atividades descritas na Fase de Aplicação do Formalismo devem ocorrer em paralelo com a Fase de Gerenciamento do Processo e a Fase de Execução das Atividades do Processo, pois o formalismo deve atuar sob as atividades presentes nestas fases. A Figura 234 mostra uma visão geral da Fase de Aplicação do Formalismo de Execução.

As atividades descritas nessa fase são organizadas conforme a ordem de importância da máquina de execução da ferramenta Spider-PE. No entanto, as atividades também podem ser realizadas por outras ferramentas que executem modelos de processo baseados no padrão SPEM 2.0

Desta maneira, a primeira atividade desta fase é a aplicação da hierarquia do processo que objetiva classificar os componentes do processo conforme o seu nível hierárquico. Este nível hierárquico será descrito com maiores detalhes na seção seguinte. O próximo passo é a aplicação das restrições do processo, estas restrições determinam o tipo de conexão entre as atividades e o fluxo de execução do processo. Em seguida, deve-se ser feito a atividade de controle das variáveis do projeto. Este controle é realizado através das informações obtidas na Fase de Gerenciamento do Processo.

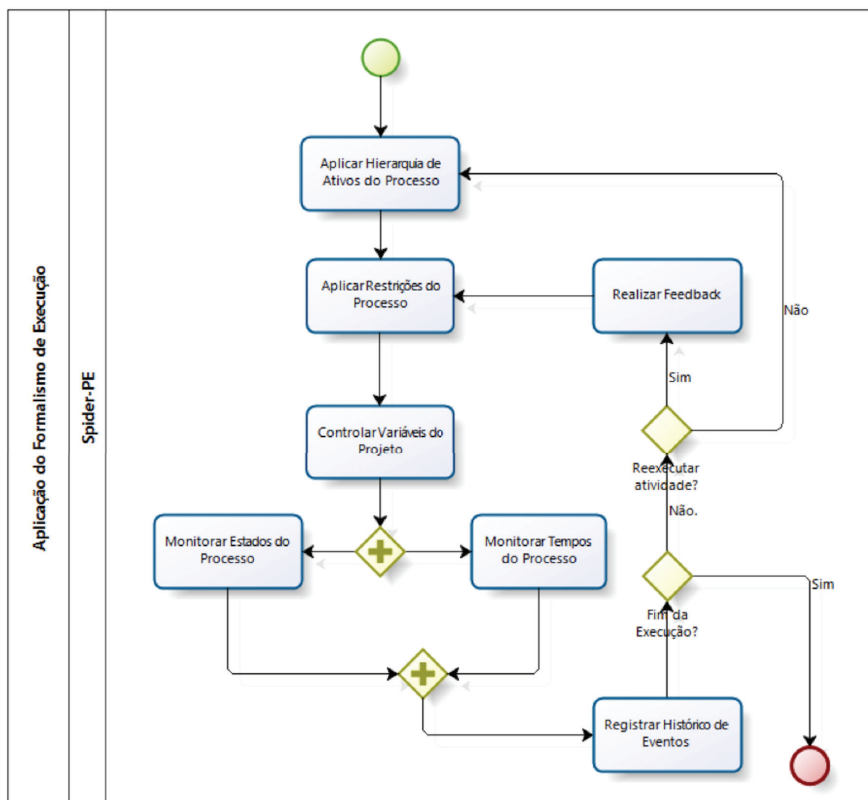


Figura 234. Fase de Aplicação do Formalismo de Execução.

As atividades de Monitoramento de Estados do Processo e Monitoramento de Tempos do Processo permitem uma representação e acompanhamento das atividades do processo em tempo real, ou seja, permitem visualizar se as atividades do processo estão sendo executadas em dado momento e se estão sendo executadas dentro do tempo proposto na Fase de Gerenciamento do Processo. Um registro do histórico de eventos relacionados as mudanças que venham a ocorrer nas atividades processo deve ser feito em um log de registro através da Máquina de Execução. Por fim, a atividade de Realizar *Feedback* é responsável pela retomada de atividades à execução, caso necessário.

Devido as características dessa fase, as organizações que utilizarem a Spider-PE estarão realizando a execução de processo de forma

semi-automatizada, a partir do uso da própria ferramenta, das demais ferramentas de apoio e da intervenção humana. Assim, durante a execução dos seus processos, essas organizações poderão optar por fluxos alternativos, retornar a execução de uma atividade já realizada, obter informações sobre o andamento do processo e demais ações, caracterizando assim a flexibilidade.















23.1.3. Máquina de Estados




A Máquina de Estados do processo diz respeito ao mecanismo de Transição de Estados e Tempo dos componentes do processo, estes componentes do processo são oriundos da linguagem de modelagem da ferramenta SPIDER_ML e para um melhor entendimento do funcionamento da Máquina de Estado, eles são descritos na subseção a seguir. Na seção seguinte, são descritos como estes componentes do processo estão relacionados entre si formando uma estrutura de hierarquia. Finalmente, na última subseção, as regras relacionados as transições de estado e tempo (*status*) são descritas e exemplificadas, esta última subseção explica o funcionamento Máquina de Estados em si. Os conceitos explicados nesta seção são oriundos da especificação técnica da linguagem xSPIDER_ML [Portela *et al.*, 2012] e sua importância nesse trabalho é primordial uma vez que define as regras de comportamento dos componentes processo e ser executado pela ferramenta Spider-PE.

a) Componentes do Processo



Os componentes do processo utilizados pela ferramenta SPIDER-PE estão descritos na especificação da xSPIDER_ML, estes componentes são oriundos da linguagem SPIDER_ML, mais especificamente são os componentes presentes na estrutura dos Processos Instanciados. A Tabela 22 descreve cada um desses componentes utilizados pela ferramenta.





Tabela 22. Descrição dos Componentes do Processo.

Ícone	Nome	Descrição
	Processo Instanciado	Uma instância do Processo Padrão para um projeto específico.
	Fase	Um período significativo de um Processo Instanciado.
	Iteração	Um conjunto de atividades e tarefas que serão repetidas dentro de uma fase.
	Marco	Um evento significativo em um Processo Instanciado.
	Atividade	Um trabalho que deve ser realizado no decorrer de um Processo Instanciado e que pode ser decomposto em outras atividades ou tarefas.
	Tarefa Instanciada	Um trabalho que deve ser realizado no decorrer de um Processo Instanciado e que não pode ser decomposto.
	Papel Instanciado	Um papel desempenhado por um recurso no Processo Instanciado.
	Produto de Trabalho Instanciado	Um produto de trabalho consumido, gerado ou modificado em um Processo Instanciado.
	Ferramenta Instanciada	Uma ferramenta que deve ser utilizada durante a realização de uma tarefa em um Processo Instanciado.
	Procedimento	Representa uma conduta que deve ser adotada durante a realização de uma tarefa do Processo Padrão ou de um Processo Instanciado.
	Estado Inicial	Representa o ponto de partida de um fluxo representado no Processo Instanciado.
	Estado Final	Representa um ponto para a finalização de um fluxo descrito no Processo Instanciado.
	Decisão e União	Indica um ponto onde existem dois ou mais caminhos que podem ser seguidos, mas apenas um desses caminhos será seguido. Também indica o ponto onde dois ou mais fluxos alternativos se unem novamente e um único fluxo.
	Barra de Separação	Representa um ponto que inicia a execução em paralelo de dois ou mais fluxos.

Ícone	Nome	Descrição
	Barra de Junção	Representa um ponto onde dois ou mais fluxos que ocorrem em paralelo são finalizados.
	Transição	Representa uma transição entre duas fases, iterações, atividades, tarefas e demais elementos de um processo instanciado.
	Nota	Representa uma descrição ou informação adicional que pode ser provida a respeito de um componente.

b) Hierarquia do Processo





Os componentes descritos anteriormente obedecem uma estrutura de hierarquia. Esta estrutura permite que o processo esteja organizado e assim seja possível mensurar o trabalho e o momento de execução de cada uma das partes do processo mais facilmente, e consequentemente do próprio processo. A execução do processo ocorre sob os processos instanciados, representados pela notação . Na linguagem SPIDER_ML, um processo instanciado é composto por um conjunto sequencial de componentes Tipo Fase . Uma fase representa um período de tempo do processo padrão e duas fases não podem ocorrer simultaneamente.

Os componentes do Tipo Fase podem ser descritos através de duas abordagens: ciclo de vida sequencial, onde cada fase é composta por componentes do tipo Atividade  e Tarefa Instanciada ; e ciclo de vida iterativo, onde cada fase é composta por componentes do tipo Iteração  e Marco .

No ciclo de vida iterativo, o componente do tipo Interação é composto por componentes do tipo Atividade e Tarefa Instanciada. Os marcos podem ocorrer no final de cada iteração, eles representam as revisões do projeto.

Os componentes do tipo Atividade são basicamente componentes do tipo iteração ou componentes do tipo fase com o ciclo de vida sequencial. Isto quer dizer, que cada Atividade é composta por componentes do tipo Atividade

e Tarefa Instanciada. Sendo assim, o componente do tipo Atividade pode conter uma ou mais atividades uma dentro da outra.

Os componentes do tipo Tarefa Instanciada são os principais componentes da execução do processo. São unidades indivisíveis e representam o que de fato vai ser realizado durante o decorrer do projeto de desenvolvimento. Em cada tarefa é necessário definir alguns elementos para que ela possa ser executada com êxito. Estes elementos são: os Papeis Instanciados , que representam os recursos humanos responsáveis pela execução da tarefa; as Ferramentas Instanciadas , que são os instrumentos necessários para a realização da tarefa; os Procedimentos , que definem as condutas que devem ser adotadas durante a realização da tarefa; e os Produtos de Trabalho Instanciados , que são artefatos que podem ser tanto o insumo necessário para a execução de uma tarefa (artefato de entrada) quanto o resultado da execução desta tarefa (artefato de saída).

A Figura 235 é um exemplo da estrutura de hierarquia entre os componentes do processo descritos nesta subseção.

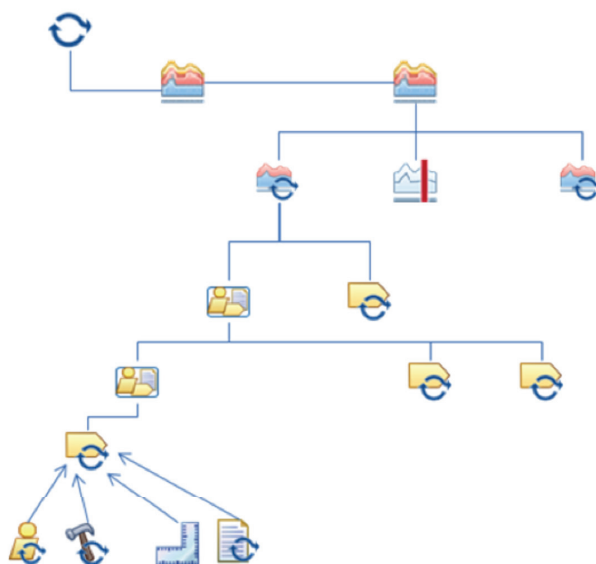


Figura 235. Hierarquia dos Componentes do Processo.

c) Regras do Formalismo

As regras do formalismo apresentadas nessa subseção descrevem o comportamento da Máquina de Estados em si, ou seja, descrevem de que forma as transições de estado e tempo (*status*) ocorrem dentro do processo. Uma transição de estado é caracterizada pela situação atual do projeto (Não Iniciado, Executando, Pausado e Finalizado); e uma transição de tempo é caracterizada pela relação entre o tempo planejado e o tempo real do projeto (Atrasado, Em tempo e Adiantado). Também existe a definição de estado para a utilização de recursos (Fora de uso, Em Uso e Finalizado). Por fim, é necessária a definição de Regras de Dependência entre dois componentes do processo, ou seja, quando é permitida o início da próxima tarefa dentro da sequência. As regras de transição e dependência são provenientes da especificação técnica da linguagem xSPIDER_ML [Portela *et al.*, 2012].

c.1) Transições de Estado

Os estados definidos na linguagem xSPIDER_ML são oriundos da abordagem WebAPSEE [Lima *et al.*, 2006], devido as referências utilizadas por ela e sua abrangência. Os possíveis estados e as suas transições, desta abordagem, são apresentados na Figura 236.



Figura 236. Diagrama de Transição de Estados [Lima *et al.*, 2006].

A linguagem xSPIDER_ML utiliza uma versão simplificada do modelo apresentado a cima. Estes estados são *notStarted*, *started*, *paused* e *finished* baseados, respectivamente, nos estados Pronta, Em execução, Em pausa e Concluída da abordagem WebAPSEE.

A Figura 237 apresenta um diagrama para os componentes do Tipo Tarefa com as seguintes possibilidades de mudança de estado que é disparado de acordo com evento realizado sobre estas.

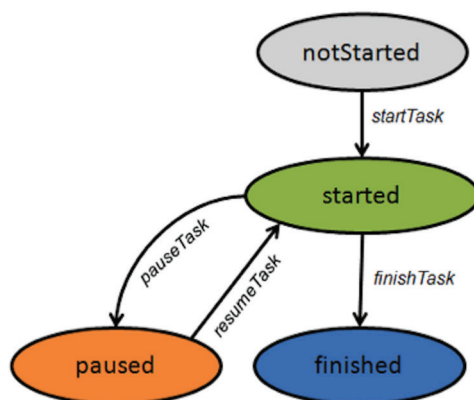


Figura 237. Transição de Estados do Componente Tarefa.

As possíveis mudanças de estado relacionadas na Figura 237 são melhor visualizadas no exemplo abaixo, onde, um componente tipo tarefa em estado inicial *notStarted* é representada pelo ícone 🔄; tarefa no estado *started* pelo ícone 🟢; no estado *paused* pelo ícone 🟠 e no estado *finished* pelo ícone 🔵.

Em relação aos outros tipos de componentes, estes herdam os estados dos componentes tarefas que os compõe de acordo com algumas regras pré-estabelecidas.

- Quando todas as tarefas do processo possuírem o mesmo estado, os demais componentes do processo possuirão, também, este mesmo estado, como visto na Figura 238;

- Quando pelo menos uma Tarefa estiver sido iniciada (estado = *started*), os demais componentes do processo hierarquicamente relacionados a esta Tarefa possuirão, também, estado igual a *started*, como pode ser visto na Figura 239;
- Quando pelo menos uma Tarefa estiver sido pausada (estado = *paused*), e as demais tarefas do mesmo nível hierárquico desta estiverem pausadas ou finalizadas, os demais componentes do processo hierarquicamente relacionados a esta Tarefa possuirão estado igual a *paused*, como pode ser visto na Figura 240;
- Caso uma Tarefa do processo tenha estado igual a finalizado e todas as demais estiverem não iniciadas, os demais componentes do processo hierarquicamente relacionados a esta Tarefa possuirão estado igual a *started* (para o elemento Processo, o estado correspondente é *enacting*), como pode ser visto na Figura 241.

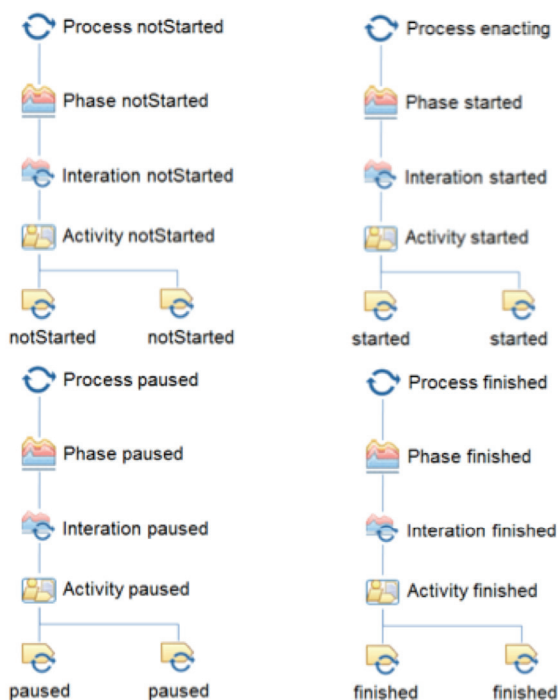


Figura 238. Hierarquia de Estado 1.

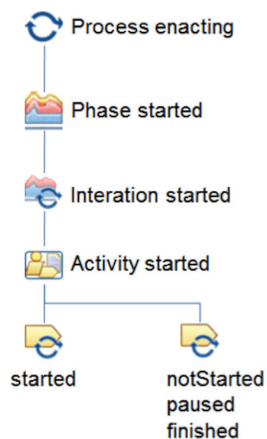


Figura 239. Hierarquia de Estado 2.

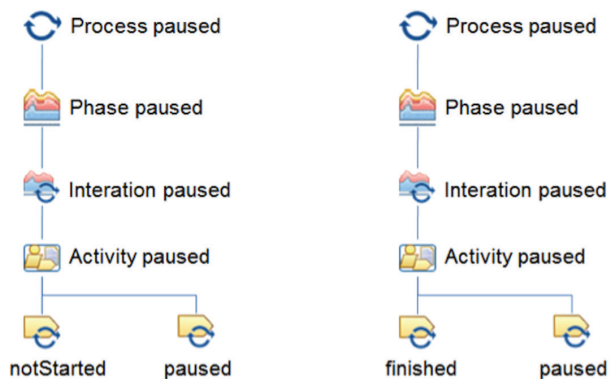


Figura 240. Hierarquia de Estado 3.

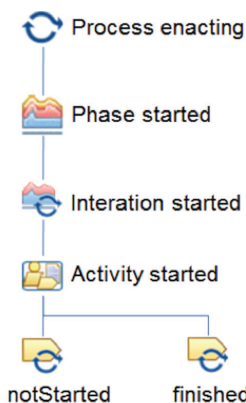


Figura 241. Hierarquia de Estado 4.

c.2) Transições de Tempo

As transições de tempo dizem a respeito ao *status* aplicados a um componente do Tipo Tarefa Instanciada. O *status* de uma tarefa é obtido comparando o tempo estimado para a realização da tarefa e tempo que de fato foi necessário para sua real execução. Assim, o diagrama da Figura 242 apresenta as possíveis mudanças relacionados ao status.

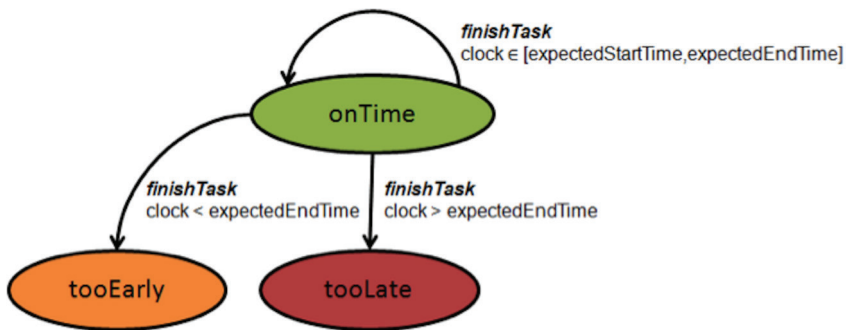


Figura 242. Diagrama de Transição de Tempo.

Para melhor entendimento, um exemplo para as transições de tempo realizadas em um componente Tipo Tarefa Instanciada é mostrado na Figura 243, onde se supõe que o tempo para executar uma tarefa é 8 horas. Assim, existem três possibilidades:

- Caso a Tarefa tenha sido concluída de acordo com o tempo estimado (8 horas), o *status* da Tarefa será *onTime*;
- Caso a Tarefa tenha sido concluída em um tempo menor que o estimado (6 horas, por exemplo), o *status* da Tarefa será *tooEarly*;
- Por fim, caso a Tarefa tenha sido concluída em um tempo maior que o estimado (12 horas, por exemplo), o *status* da Tarefa será *tooLate*;

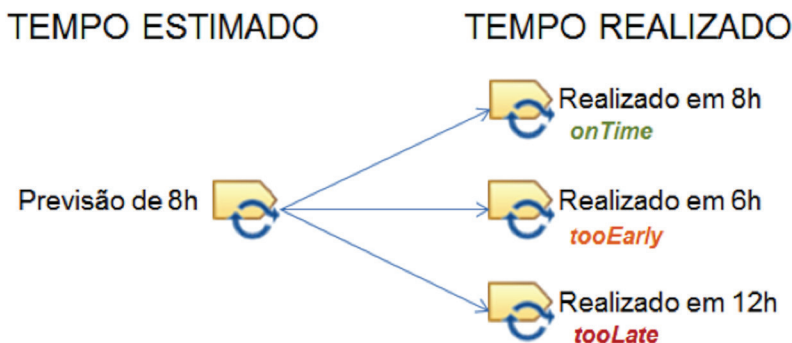


Figura 243. Exemplo de Transição de Estado em Tarefa Instanciada.

Para os outros componentes do processo, a máquina de estados da ferramenta dispara um *clock* interno que compara a soma de todos os sub-elementos de determinado componente definindo assim seu tempo real de execução. Este tempo real de execução é, dessa forma, comparado com o a soma do tempo estimado dos sub-elementos do componente em questão. Um exemplo, oriundo de [Portela *et al.*, 2012] define a seguinte situação:

- Suponha que um Projeto P foi estimado para ser realizado em 15 horas e que cada Fase do projeto (F1 e F2) foi estimada para ser realizada em 12 e 3 horas, respectivamente. Dentro da Fase F1, foi planejado que haveria duas Iterações (I1 e I2) que seriam realizadas em 9 e 3 horas, respectivamente. A Iteração I1 é constituída de duas Atividades (A1 e A2), estimadas para serem realizadas em 7 e 2 horas cada. A Atividade A1 é dividida em duas Tarefas (T1 e T2), para serem executadas em um tempo estimado de 3 e 4 horas, respectivamente. Já a Atividade A2, estimada em 2 horas, possui apenas uma Tarefa (T3), estimada no mesmo período de tempo (2 horas). A Iteração I2 é constituída de uma Atividade (A3) e uma Tarefa (T4) que, de acordo com o planejamento, seriam realizadas no mesmo

tempo estimado da Iteração (3 horas). A Figura 244 representa a situação descrita.

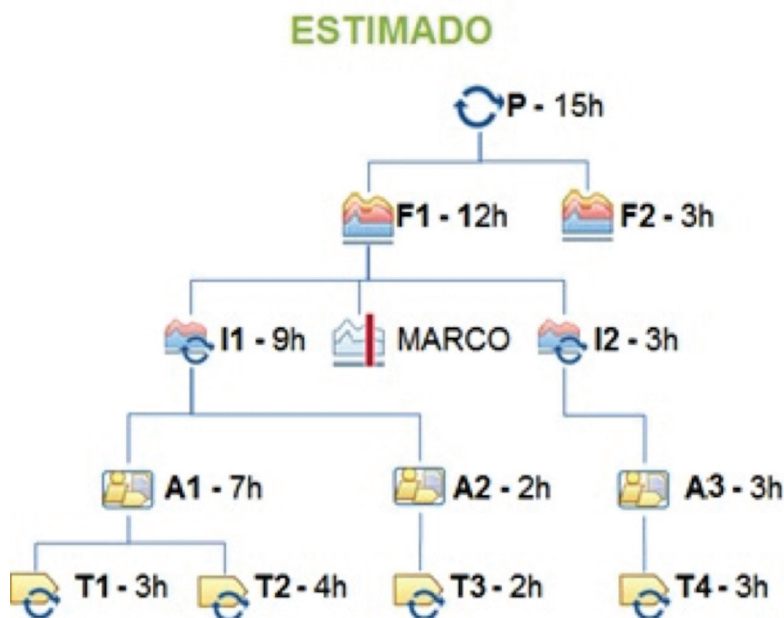


Figura 244. Exemplo de Estimativa de Tempo entre Componentes do Processo.

- Uma vez que o esforço necessário para realizar as atividades do processo tenha sido estimado e todo o planejamento realizado, o processo pode então ser executado. Suponha que, neste exemplo, durante a execução não foram finalizadas as Fases F1 e F2. Suponha, também, que a Iteração I2, a Atividade A3 e a Tarefa T4 não foram finalizadas. Consequentemente, o Processo ainda não foi concluído (encontra-se Em Execução). Entretanto, suponha que já foram realizadas as Tarefas T1, T2 e T3 no tempo de 4, 2 e 3 horas, respectivamente, e que as Atividades A1 e A2 foram realizadas em 6 e 3 horas, respectivamente. Também, suponha que a Iteração I1 foi realizada em 9 horas. A Figura 245 representa a situação descrita anteriormente.

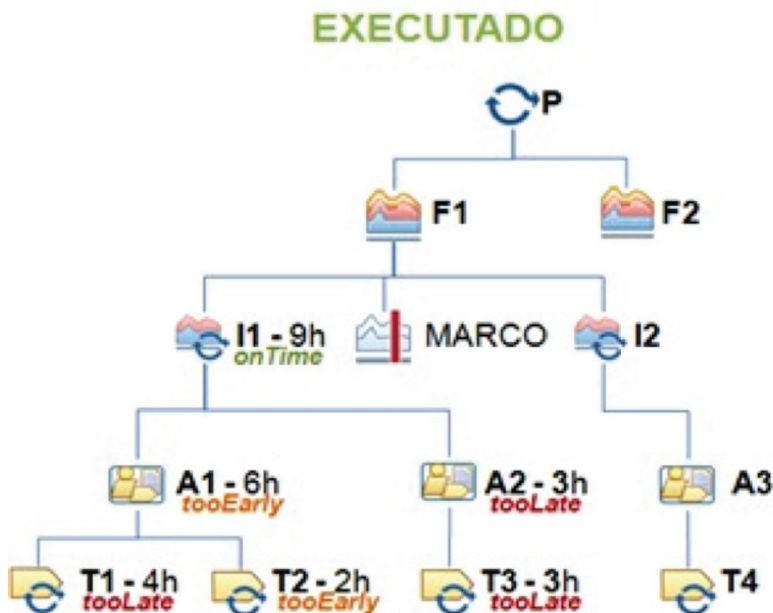


Figura 245. Exemplo de Transição de Tempo entre Componentes do Processo.

Ao final de cada execução, é dado aos elementos do processo um *status*, referente à relação do tempo em que a sua execução foi realizada com o tempo estimado para sua execução. Alguns elementos que são executados com atraso (A2, T1 e T3) recebem o *status* de *tooLate*; outros que são terminados adiantados em relação ao tempo estimado (A1 e T2) recebem o *status* de *tooEarly*; e um elemento finalizado no tempo estimado (I1) recebe o *status* de *onTime*.

É importante ressaltar que o atraso na execução de um elemento hierarquicamente inferior não infere necessariamente que o elemento hierarquicamente superior a este terá a sua execução atrasada. Um exemplo desta situação é a Tarefa T1, que apesar de ter sido concluída com atraso, não atrasou a realização da Atividade A1. Isso ocorreu devido a Tarefa T2 ter sido finalizada antes do prazo estimado. Além disto, a Iteração I1 foi concluída no prazo estimado, apesar de seus elementos hierarquicamente inferiores não terem sido terminados no prazo estimado.

Para um entendimento mais formalizado das transições de estado e tempo, definiu-se uma linguagem formal em [Portela e Gomes, 2011]. Essa linguagem define um passo a passo detalhado do funcionamento dessas transições e todas as suas possibilidades.

c.3) Transição de Estado de Recursos

Um recurso é associado a uma tarefa e ele pode adotar três classificações distintas: quando o recurso é de uso exclusivo de uma tarefa por vez (*Exclusive*), quando o recurso é de uso compartilhado por várias tarefas simultaneamente (*Shareable*) e quando o recurso pode ser utilizado apenas uma vez (*Consumable*).

Desta forma, as regras relacionadas a transição de estado de recursos, também oriundas da abordagem WebAPSEE [Lima *et al.*, 2006], apresentam o diagrama para recursos do tipo *Exclusive* e *Shareable* definido na Figura 246.

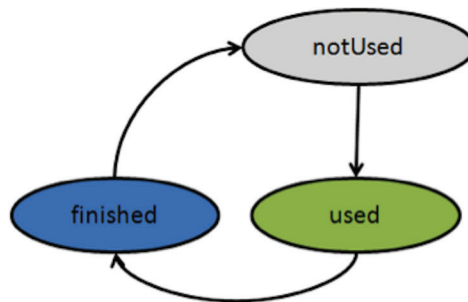


Figura 246. Diagrama de Transição de Estado para Recursos *Exclusive* e *Shareable*.

Para os recursos do tipo *Consumable*, tem-se o diagrama apresentado na Figura 247.

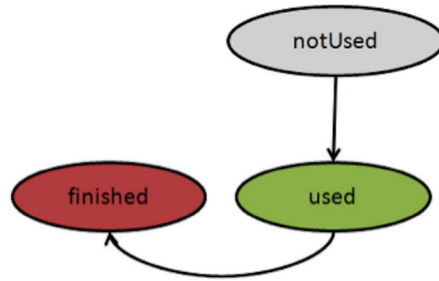


Figura 247. Diagrama de Transição de Estado para Recursos *Consumable*.

c.4) Regras de Dependência Entre Componentes

As regras de dependência definem os diferentes tipos de relações entre componentes do processo, de acordo com a SPEM 2.0. Essas regras são necessárias, pois os componentes do processo são organizados de forma sequencial dentro do modelo de processo.

Para exemplificar o uso destas regras, suponha-se que uma sequência (relação) entre dois Componentes do Processo em que um destes componentes (referido como o Componente B no exemplo abaixo) dependa do início ou término de outro componente (denominado como o Componente A no exemplo abaixo), a fim de ser iniciado ou finalizado. Poderá haver quatro tipos de dependências entre estes elementos:

- ***finishToStart***: o Componente B não pode iniciar até o Componente A finalizar. Por exemplo, se existirem dois componentes, “Criar DAO” e “Testar DAO”, “Testar DAO” não poderá ser iniciada até que “Criar DAO” seja finalizado. Este é o tipo mais comum de dependência, estabelecido como o padrão para uma nova instância de conexão entre componentes;
- ***finishToFinish***: o Componente B não pode finalizar até o Componente A finalizar. Por exemplo, se existirem dois Componentes, “Cadastrar Cliente” e “Editar Dados do Cliente”,

“Editar Dados do Cliente” não pode finalizar até “Cadastrar Cliente” ser finalizado;

- ***startToStart***: o Componente B não pode inicializar até o Componente A for iniciado. Por exemplo, se existirem dois componentes, “Codificar Requisitos” e “Coletar Métricas”, “Coletar Métricas” não pode iniciar até que “Codificar Requisitos” inicie;
- ***startToFinish***: o Componente B não pode finalizar até o Componente A ser iniciado. Este tipo de dependência pode ser utilizado para agendamento *just-in-time*, para um Marco ou para a data de término do projeto a fim de minimizar o risco de um componente finalizar antes de seus componentes dependentes. Se um componente relacionado precisa finalizar antes da data de conclusão de um Marco do projeto, mas não importa exatamente quando, e não existe a necessidade de finalizar, pois isso afetaria o *just-in-time* do componente, poderia criar-se uma *startToFinish dependency* (ver SPEM 2.0) entre o componente que necessita ter um *just-in-time* (o predecessor) e o componente relacionado (o sucessor). Então, se o progresso no componente sucessor for atualizado, isso não afetará as datas previstas do componente predecessor.

23.2. A Ferramenta Spider-PE

Conforme mencionado anteriormente, a ferramenta Spider-PE foi desenvolvida utilizando a linguagem de programação JAVA. Sua interface gráfica foi construída através dos componentes gráficos da biblioteca de programação SWING nativa do JAVA, com suporte do *plugin WindowBuilder 3.8.1* utilizado na ferramenta de desenvolvimento integrado (IDE) Eclipse Juno. As funcionalidades desta ferramenta tem como fundamentação os fluxos do *framework* Spider-PE e o formalismo da máquina de estados, cujo conceitos são oriundos da linguagem xSPIDER_ML, descritos nas seções anteriores.

Com o objetivo de dar suporte às atividades descritas nos fluxos de atividades, a ferramenta Spider-PE utiliza o conceito de módulos para especificar um determinado conjunto de funcionalidades agrupadas e os atores responsáveis por sua execução. Sendo assim, a ferramenta possui três módulos: **Administração**, **Gerência do Processo** e **Execução do Processo**. Respectivamente, os atores responsáveis pela operação destes módulos são: **Administrador**, **Gerente de Processo** e **Usuário do Processo**.

A Figura 248 apresenta a tela inicial da ferramenta onde é possível autenticar-se na através dos três módulos descritos anteriormente. O módulo pode ser escolhido através da seleção do Tipo de *Login*, que apresenta uma lista dos atores responsáveis por cada um desses módulos. Nas seções abaixo, estes módulos são descritos com maiores detalhes.



Figura 248. Tela de *Login* da Ferramenta.

23.2.1. Módulo de Administração da Ferramenta

A principal função deste módulo é definir as opções de configuração da ferramenta, tais como, configuração de ferramentas integradas, servidor de

envio de e-mail e definição do processo que será gerenciado e executado nas etapas posteriores.

- **Configuração de Ferramentas de Estimativa, Gestão de Mudança e Medição:** As telas de configuração de ferramenta são de quatro tipos, divididas em Ferramentas de Estimativa: Spider-APF, Spider-UCP e Spider-CoCoMo, Ferramentas de Gestão de Configuração (SUBVersion e Redmine), Ferramenta de Medição (Spider-MsControl) e Ferramenta de Avaliação (Spider-CL). Na Figura 249 é possível especificar o caminho do arquivo executável da ferramenta. Conforme descrito no início deste capítulo, estas são ferramentas desenvolvidas e mantidas pelo Projeto SPIDER (com exceção das Ferramentas de Gestão de Configuração) e podem ser encontradas em <http://www.spider.ufpa.br>.



Figura 249. Tela de Configuração de Ferramentas de Estimativas.

- **Configuração de Comunicação:** A configuração de comunicação da ferramenta diz a respeito do cadastro dos dados de e-mail, como visto na Figura 250. Esta é uma configuração importante para o uso da ferramenta, pois através desse e-mail que será enviado mensagens importantes para os usuários do processo. Como por exemplo, envio de mensagens de tarefas atrasadas ou envio de informações importantes para os interessados no projeto (*stakeholders*).

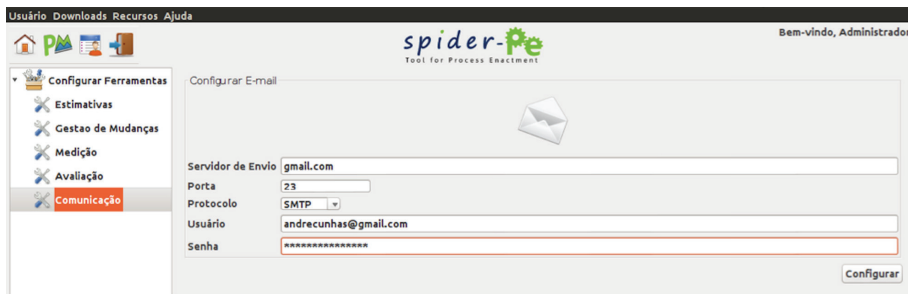


Figura 250. Tela de Configuração de E-mail.

- **Configuração do Processo:** Nesta etapa é possível realizar o carregamento do processo que será gerenciado e executado nos outros módulos, como visto na Figura 251. Este processo deve estar no formato XML e ter sido gerado pela ferramenta Spider -PM. Internamente, a Spider -PM gera um arquivo XML contendo um processo padrão e vários processos instanciados relacionados.



Figura 251. Tela de Configuração de Processo.

A ferramenta Spider-PE, através do uso do plugin JAVA xStream para leitura de arquivos XML, coloca cada um dos processos instanciados em um arquivo XML diferente. Uma vez que a execução de processo ocorre sob o processo instanciado, é possível realizar o carregamento dos dados no banco de dados da Spider-PE sem haver conflito de dados, como pode ser acompanhado na Figura 252.

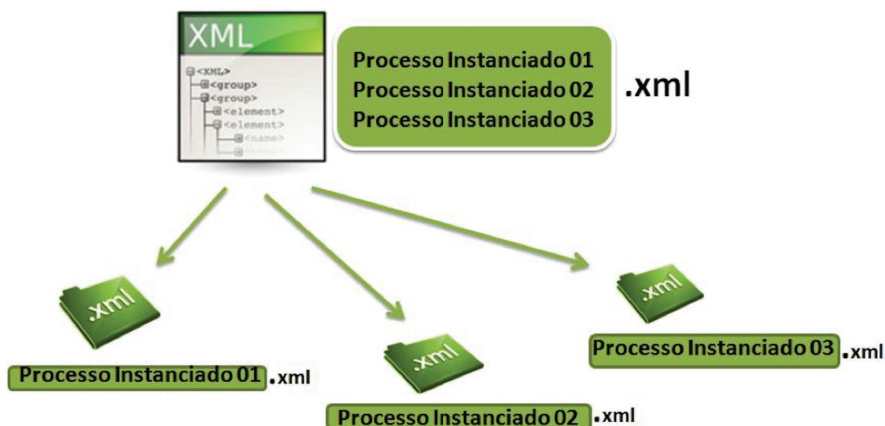


Figura 252. Leitura do Arquivo XML oriundo da Spider-PM.

Na etapa de Cadastro de Gerente é possível definir quem será o gerente do processo carregado na etapa de Configuração do Processo, como visto na Figura 253. O gerente deve ser um recurso humano previamente cadastrado na tela de Recursos Humanos no módulo de Administração. Importante ressaltar que a tela de tipos pode ser acessada tanto pelo Módulo de Administração quanto pelo Módulo de Gerência do Processo.

A captura de tela mostra a interface do usuário do Spider-PM. No topo, há uma barra de navegação com links: "Usuário", "Downloads", "Recursos", "Ajuda". O logotipo "spider-PM" e o slogan "Tool for Process Enactment" estão no topo central. À direita, o texto "Bem-vindo, Administrador" é exibido. No lado esquerdo, há um menu lateral com opções: "Configurar Ferramentas", "Estimativas", "Gestão de Mudanças", "Medição", "Avaliação" e "Comunicação" (destacado em vermelho). A área principal da tela é intitulada "Cadastrar Gerentes de Processo". Ela contém campos para "Processo" (Selecione...), "Gerente" (Selecione...), "Login (e-mail)", "Senha" e "Repetir Senha". Um botão "Cadastrar" está à direita. Abaixo, há uma tabela com as seguintes colunas: "Processo", "Gerente", "Login" e "Excl...". A tabela contém uma única linha de dados: "Processo Spider", "elder", "le" e um ícone de exclusão (X vermelho).

Figura 253. Tela de Configuração de Cadastro de Gerente.

As configurações realizadas nesse módulo devem ser realizadas antes do início da abertura dos demais módulos, através dessa correta configuração é que as principais funcionalidades da ferramenta poderão ser executadas satisfatoriamente.

23.3.2. Módulo de Gerência de Processo

Este módulo baseia-se na aderência aos modelos de melhoria MR-MPS-SW e CMMI-DEV, por isto, está relacionado às atividades de planejamento e monitoramento do processo. Este módulo é baseado no fluxo de atividades descrito anteriormente. Assim como a Fase de Gerenciamento de Processo, o módulo de Gerência de Processo é composto de três etapas: **Planejamento, Execução e Monitoramento**.

O módulo de Gerência do Processo inicia-se a partir da etapa de planejamento do processo, onde o Gerente de Processo realiza atividades diversas, como:

- **Política Organizacional:** esta etapa consiste em definir e publicar uma política organizacional (Figura 254). Esta política contém diretrizes e responsabilidades para áreas específicas do processo, como Gerência de Projeto, Gerência de Requisitos, dentre outras. Para implementar esta funcionalidade foi necessária a utilização das bibliotecas SVNKit e iText do JAVA. Quando o Gerente do Processo publica uma política organizacional, um arquivo no formato PDF é gerado por meio da biblioteca iText. Este arquivo, que contém as informações da política, é armazenado no servidor por meio do SVNKit que acessará a ferramenta SUBVersion, configurada no Módulo de Administração;

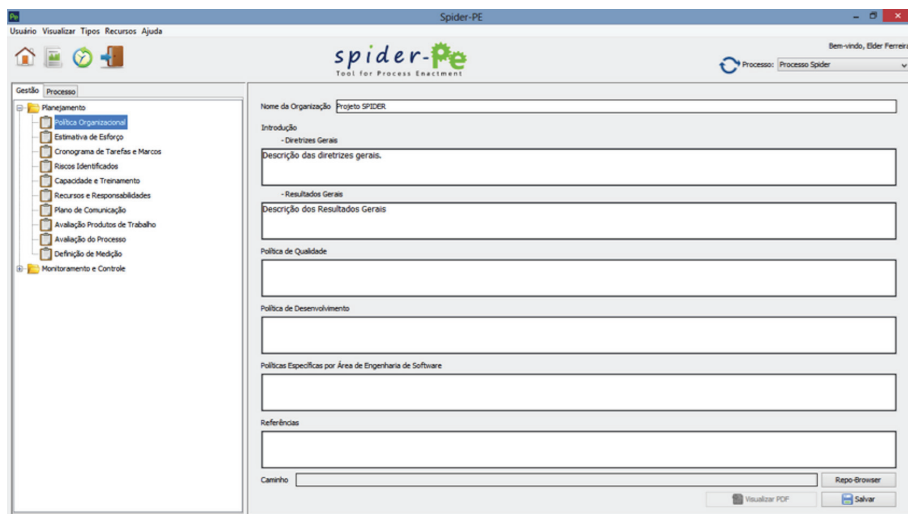


Figura 254. Tela da Política Organizacional.

- **Estimativa de Esforço:** esta etapa consiste em estimar o esforço necessário para realizar as atividades e tarefas do processo (Figura 255);

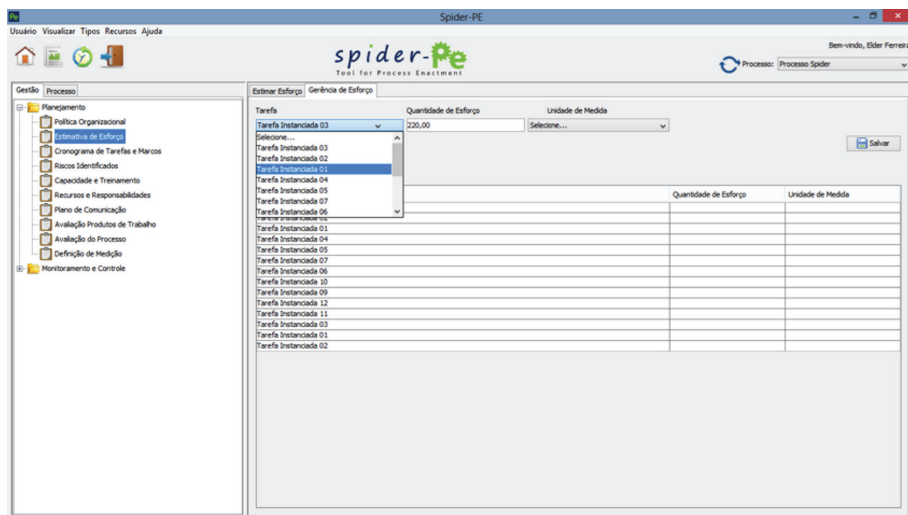


Figura 255. Tela de Estimativa de Esforço

- **Cronograma de Tarefas:** esta etapa é baseada na estimativa de esforço da etapa anterior, essa etapa consiste em definir o cronograma do processo. Esta funcionalidade utiliza a biblioteca JCalendar do JAVA para a geração de calendários (como pode ser visto na Figura 256);

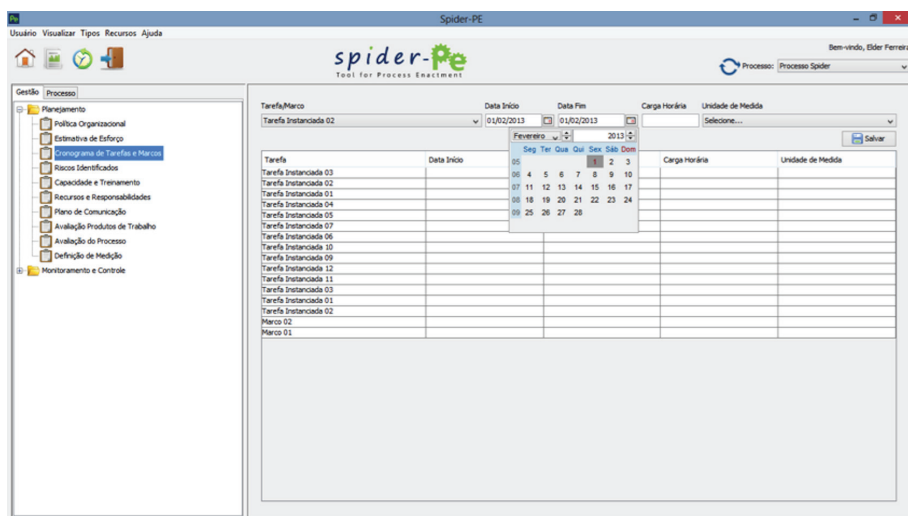


Figura 256. Tela de Cronograma de Tarefas

- **Riscos Identificados:** esta etapa consiste em identificar os riscos do processo e definir um plano de mitigação e contingência para estes riscos (Figura 257). Dependendo do impacto do risco, poderá haver a necessidade de re-estimativa do esforço necessário na Tela de Estimativa de Esforço;

The screenshot shows the Spider-PE software interface. The title bar reads 'Spider-PE'. The top menu bar includes 'Usuário', 'Visualizar', 'Tipos', 'Recursos', and 'Ajuda'. The top right corner says 'Bem-vindo, Elder Ferreira' and 'Processo: Processo Spider'. The sidebar on the left is titled 'Gestão | Processo' and contains a tree view with the following items: 'Planejamento' (expanded), 'Política Organizacional', 'Estimativa de Esforço', 'Cronograma de Tarefas e Marcos', 'Capacidade e Treinamento', 'Recursos e Responsabilidades', 'Plano de Comunicação', 'Avaliação Produtos de Trabalho', 'Avaliação do Processo', 'Definição de Medição', and 'Monitoramento e Controle'. The main area is titled 'Cadastro de Risco' and 'Configuração de Riscos'. It contains a form with the following fields: 'Nome do Risco' (text box), 'Tipo de Risco' (dropdown menu), 'Descrição' (text area), 'Probabilidade' (text box), 'Impacto' (text box), 'Severidade' (text box), 'Mitigação' (text box), 'Contingência' (text box), and 'Responsável' (dropdown menu). At the bottom right of the form are buttons for 'Limpar', 'Excluir', and 'Cadastrar'. Below the form is a table with the following columns: 'Risco', 'Tipo', 'Probabilidade', 'Impacto', and 'Responsável'. The table contains one row: 'risco 1', 'tipo de risco 1', '20%', and 'Elder Ferreira'.

Figura 257. Tela de Riscos Identificados

- **Capacidade e Treinamento:** esta etapa consiste em identificar as capacidades dos membros da Equipe do Projeto (Figura 258). Esta capacidade pode ser definida a partir das habilidades necessárias para concluir as atividades do processo. Ainda nesta etapa, verifica-se a possibilidade de realização de treinamentos para suprir as necessidades de capacitação da equipe;
- **Recursos e Responsabilidades:** esta etapa consiste em definir os recursos (hardware e software) e as responsabilidades (recursos humanos) necessários para a realização das atividades do processo (Figura 259). Após a identificação das responsabilidades se realiza a alocação destes com os papéis definidos dentro do projeto;
- **Plano de Comunicação:** sub etapa consiste em identificar os interessados (*stakeholders*) afetados pela realização do projeto, com o objetivo de planejar a comunicação entre estes (Figura 260). Planejar a comunicação significa fazer com que os *stakeholders* tenham acesso às informações e documentos relevantes. Este acesso às informações é feito através do cadastro dos e-mails dos interessados;

The screenshot shows the 'Spider-PE' application window. The title bar indicates the user is 'Ben-vindo, Elder Ferreira'. The interface is divided into several sections:

- Sidebar (Gestão | Processo):** A tree view showing the process structure. The 'Capacidades e Treinamento' (Capabilities and Training) item is selected under the 'Processo' (Process) category.
- Main Form (Capacidades | Treinamento):**
 - Recurso Humano:** A dropdown menu showing 'André Cunha' and an email field with 'andrecunha@gmail.com'.
 - Conhecimento:** A text field containing 'Hibernate, J2EE, JPE'.
 - Papel Exercido:** A section with a 'desenvolvedor' (developer) role and a 'Formação Acadêmica' (Academic Formation) field with 'Análisis da informação'.
 - Formação Profissional:** A section with a 'Formação Acadêmica' field containing 'Introdução ao RPS-BM'.
 - Formação Profissional:** A section with a 'Formação Profissional' field containing 'Introdução ao RPS-BM'.
 - Formação Profissional:** A section with a 'Formação Profissional' field containing 'Introdução ao RPS-BM'.
- Table:** A table with columns: Nome, Email, Conhecimento, Papel, Formação Acadêmica, Formação Profissional, and Cursos e Certificações. It is currently empty.

Figura 258. Tela de Capacidade e Treinamento.

The screenshot shows the 'Spider-PE' application window. The title bar indicates the user is 'Ben-vindo, Elder Ferreira'. The interface is divided into several sections:

- Sidebar (Gestão | Processo):** A tree view showing the process structure. The 'Recursos e Responsabilidades' (Resources and Responsibilities) item is selected under the 'Processo' (Process) category.
- Main Form (Recursos | Responsabilidades):**
 - Tarefa/Marco:** A dropdown menu showing 'Tarefa Instanciada 04'.
 - Papel Associado:** A dropdown menu showing 'Papel Instanciado 04'.
 - Habilidades Requeridas:** A text field.
 - Recurso Humano:** A dropdown menu showing 'André Cunha'.
 - Papel Exercido:** A text field.
 - Conhecimento:** A text field.
- Table:** A table with columns: Tarefa/Marco, Papel, and Responsáveis. It contains one row:

Tarefa/Marco	Papel	Responsáveis
Tarefa Instanciada 03	Papel Instanciado 03	André Cunha

Figura 259. Tela dos Recursos e Responsabilidades.

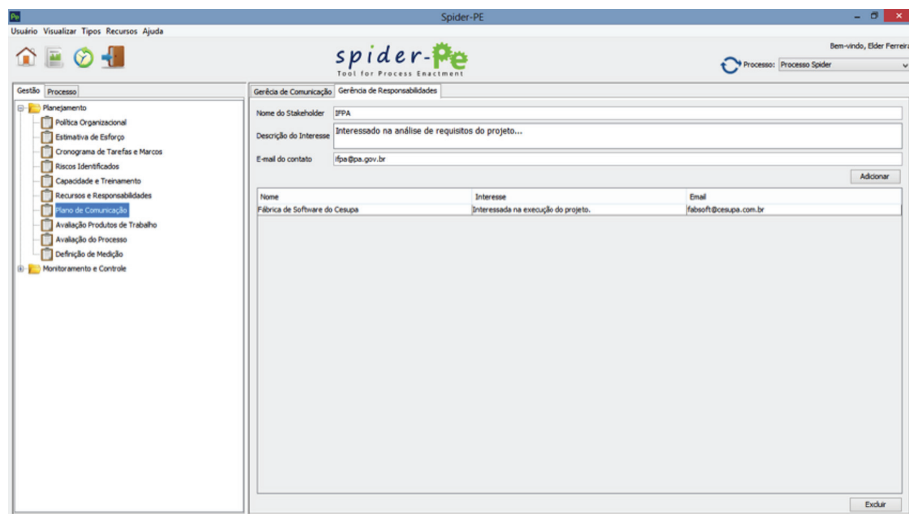


Figura 260. Tela de Plano de Comunicação.

- **Avaliação dos Produtos de Trabalho:** esta etapa consiste em identificar e avaliar os requisitos dos produtos de trabalho, a partir da definição de padrões, *templates* e demais atributos necessários para a geração de cada um destes produtos de trabalho durante todo o desenvolvimento do processo (Figura 261). A avaliação destes produtos de trabalho é feita através do apoio da ferramenta Spider-CL, configurada no módulo de administração. O resultado dessa avaliação é armazenado no servidor por meio do SVNKit;
- **Avaliação do Processo:** esta etapa consiste em gerar um checklist de avaliação das diversas áreas do Processo (Figura 262). Essa avaliação é feita através do apoio da ferramenta Spider-CL, configurada no módulo de administração. O resultado dessa avaliação é armazenado no servidor por meio do SVNKit;
- **Definição de Medição:** esta etapa consiste em definir as atividades de monitoramento e controle que objetiva garantir a realização do processo ocorra conforme o planejado (Figura 263). Este monitoramento é feito através do apoio da ferramenta Spider-MsControl,

configurada no módulo de administração. O resultado dessa avaliação é armazenado no servidor por meio do SVNKit;

A interface do Spider-PE, versão 1.0.0, apresenta o menu 'Gestão' com opções como Planejamento, Políticas Organizacionais, Estimativa de Esforço, Cronograma de Tarefas e Marcos, Riscos Identificados, Capacidade e Treinamento, Recursos e Responsabilidades, Plano de Comunicação, Avaliação Produtos de Trabalho (destacado), Avaliação do Processo, Definição de Medição e Monitoramento e Controle. A seção principal, 'Avaliação de Produto de Trabalho', contém o logotipo 'spider-PE Tool for Checklist' e o texto 'Criar Checklist para Avaliação do Produto de Trabalho'. Abaixo, há campos para 'Artefato' (menu suspenso), 'Enviar Checklist', 'Nome do Checklist criado', 'Salvar Em' e botões 'Procurar', 'Rapo-Browser' e 'Cadastrar'. Uma tabela com cabeçalhos 'Artefato', 'Nome do Checklist', 'Checklist', 'Incluir' e 'Excluir' está presente na base da página.

Figura 261. Tela de Avaliação dos Produtos de Trabalho

A interface do Spider-PE, versão 1.0.0, apresenta o menu 'Gestão' com opções como Planejamento, Políticas Organizacionais, Estimativa de Esforço, Cronograma de Tarefas e Marcos, Riscos Identificados, Capacidade e Treinamento, Recursos e Responsabilidades, Plano de Comunicação, Avaliação Produtos de Trabalho, Avaliação do Processo (destacado), Definição de Medição e Monitoramento e Controle. A seção principal, 'Avaliação de Área de Processo', contém o logotipo 'spider-PE Tool for Checklist' e o texto 'Criar Checklist para Avaliação da Área de Processo'. Abaixo, há campos para 'Área de Processo' (menu suspenso), 'Enviar Checklist', 'Nome do Checklist criado', 'Salvar Em' e botões 'Procurar', 'Rapo-Browser' e 'Cadastrar'. Uma tabela com cabeçalhos 'Área de Processo', 'Nome do Checklist', 'Checklist', 'Incluir' e 'Excluir' está presente na base da página.

Figura 262. Tela de Avaliação do Processo.

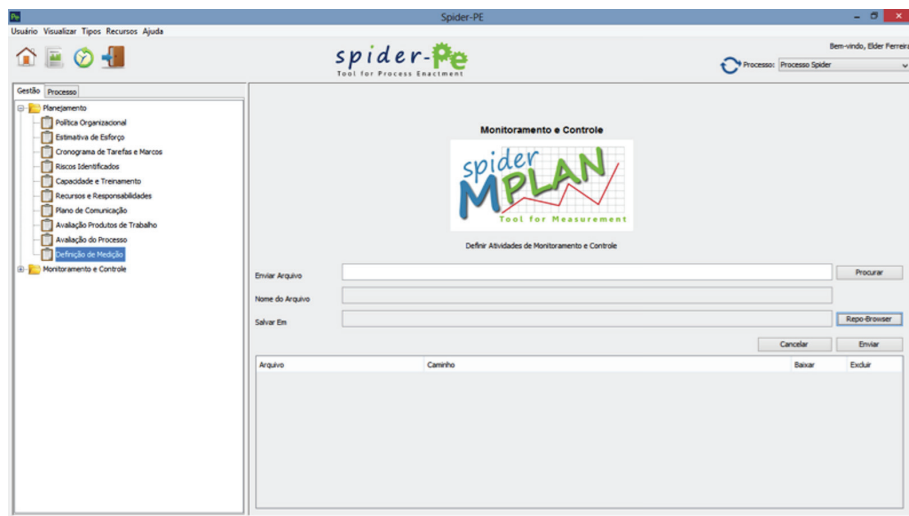


Figura 263. Tela de Definição de Medição.

- **Configuração de Tipos:** esta etapa consiste no cadastro em diferentes tipos de campos presentes na ferramenta (Figura 264), estes tipos são informações comuns durante as telas cadastros presentes na ferramenta, como por exemplo, as unidades de medida de carga horária, que podem ser configuradas em dias, horas, minutos, entre outros.

Além da etapa de planejamento, o módulo de Gerente de Processo possui a etapa de Monitoramento, onde o Gerente de Processo monitora os problemas que podem ocorrer durante a execução do projeto. Na etapa de monitoramento é realizada a etapa de Revisões no Processo (Figura 265). Esta etapa consiste em realizar revisões sobre as avaliações feitas em cima dos produtos de trabalho, da medição do processo e do próprio processo. Além disso, esta etapa também deve identificar, registrar e monitorar os possíveis problemas que surgirem durante a execução, e também manter o acompanhamento da resolução destes. Estas revisões são realizadas durante os marcos do processo. Para a realização dessas revisões, utilizam-se as

ferramentas de apoio Spider-CL, Spider-MsControl e Redmine, configuradas no módulo de administração.

Figura 264. Tela de Configuração de Tipos.

Figura 265. Tela de Revisões no Processo.

Outra funcionalidade importante do módulo de Gerência do Processo é a definição das dependências entre as tarefas a serem executadas, como mostra a Figura 266. Ou seja, definir se uma a dependência entre duas Tarefas, Tarefa 01 e Tarefa 02, por exemplo, deve ser *finishToStart* (a Tarefa 02 apenas pode ser inicializada quando a Tarefa 01 for concluída), *startToStart* (a Tarefa 02 apenas pode ser inicializada quando a Tarefa 01 também for inicializada), *startToFinish* (a Tarefa 02 apenas pode ser finalizada quando a Tarefa 01 for inicializada) e *finishToFinish* (a Tarefa 02 apenas pode ser finalizada quando a Tarefa 01 também for finalizada).

Esta funcionalidade da ferramenta utiliza a biblioteca JGraph do JAVA para a criação dos componentes gráficos da modelagem do processo de software. Essa biblioteca permite uma representação dos componentes do modelo a partir da notação presente na teoria dos grafos, onde cada componente é um vértice e cada transição uma aresta.

Estas definições de dependência são aplicadas sobre o modelo de processo carregado previamente no módulo de administração desta ferramenta, este modelo de processo é oriundo da Spider-PM.

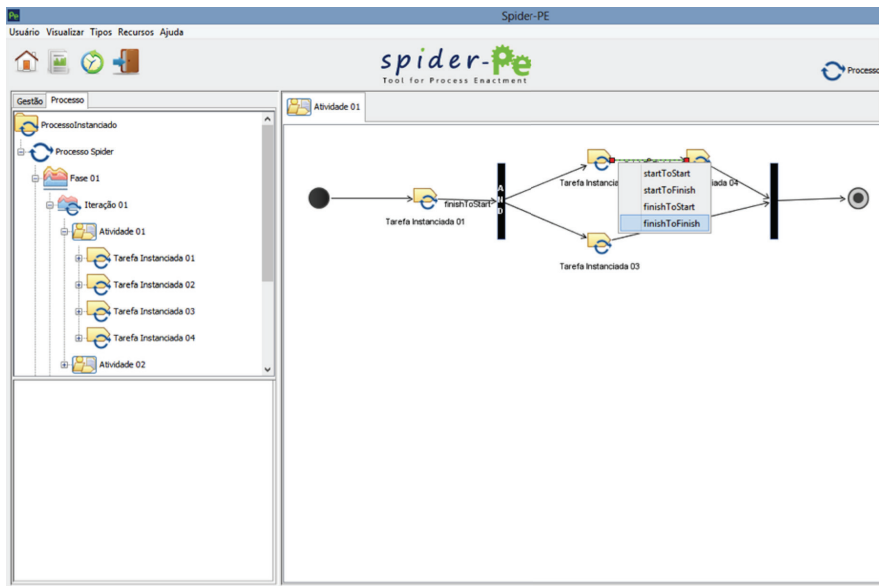


Figura 266. Tela de Definição de Dependências das Tarefas do Processo.

23.3.3. Módulo de Execução do Processo

Este módulo é responsável pela execução das tarefas do processo, ele realiza opção de comparação de prazos estimados e reais, registro do trabalho já feito e acompanhamento da execução. As funcionalidades presentes nestes módulos estão relacionadas aos conceitos do formalismo descritos anteriormente.

- Estado das Tarefas:** Nesta etapa é possível definir três estados para uma tarefa (Figura 267): **Não iniciado** (quando não há ícone no canto inferior do ícone da tarefa), significa dizer que a tarefa em questão ainda não foi inicializada; **Executada**, significa dizer que a tarefa em questão começou a ser feita; **Pausada**, significa que a tarefa em questão está parada; e **Finalizada**, significa que a tarefa em questão teve todas suas pendências concluídas. Estes estados foram exemplificados anteriormente na seção sobre a Máquina de Estados;

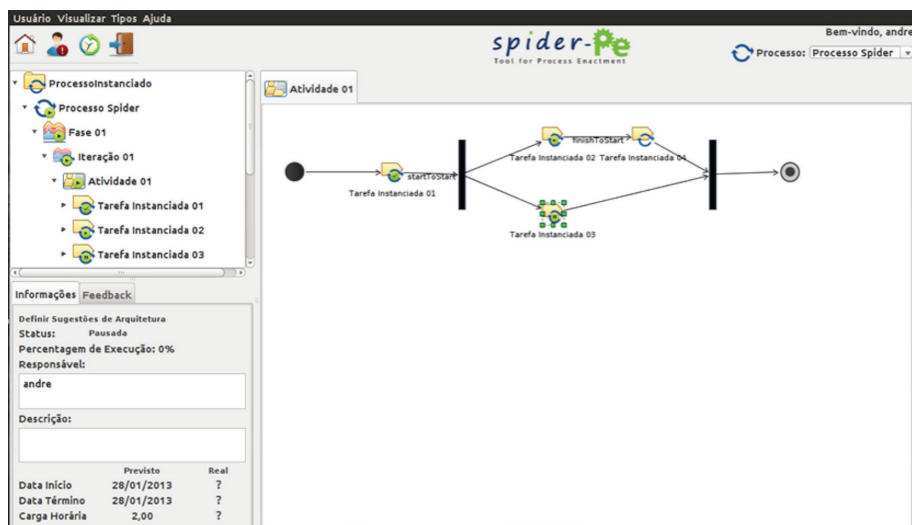


Figura 267. Exemplo de Definição dos Estados da Tarefa.

- **Informação e *Feedback* da Tarefa:** Conforme pode ser visto na Figura 268, localizado na parte inferior à esquerda existem duas abas. O conteúdo presente nestas abas muda de acordo com o componente selecionado. Na aba *Informações* é possível visualizar o *status*, percentagem de execução, o responsável e uma breve descrição de uma tarefa. Além disso, há informações relacionadas ao prazo real e prazo previsto, e carga horária real e carga horária prevista. Na aba de *Feedback* é possível, além de mudar o estado da tarefa, informar a sua percentagem de execução através de uma barra;

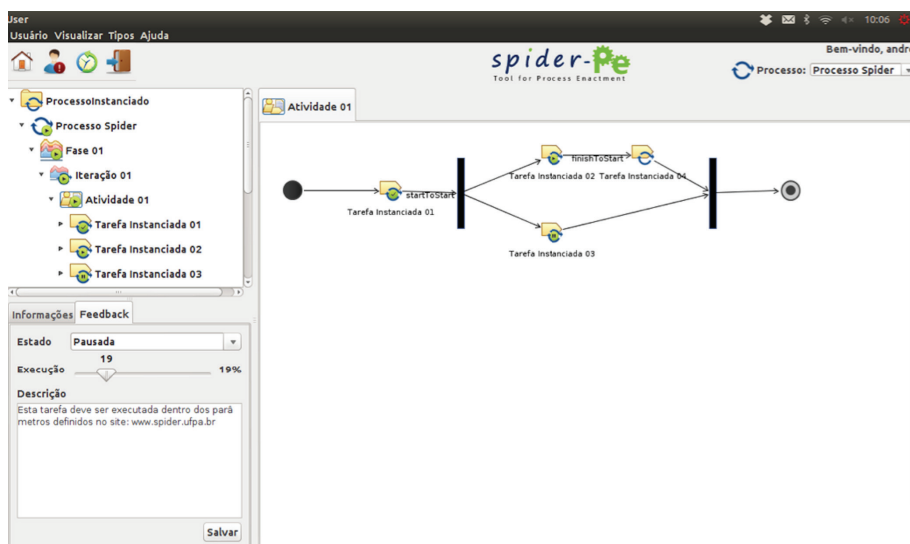


Figura 268. Tela de Informações e *FeedBack*.

- **Notificação de Tarefas atrasadas:** Quando o prazo real de execução de uma tarefa excede o tempo estimado, é disparada uma notificação para o usuário do processo responsável por aquela determinada tarefa. As notificações podem ser verificadas no segundo ícone de acesso rápido do módulo de Execução do Processo, localizado no topo à esquerda da Figura 269. Essa notificação é enviada através do

e-mail pré-definido na etapa de definição de Capacitação, conforme definido no módulo de Gerência de Processo;

- **Log de atividades:** Nesta etapa é possível realizar o acompanhamento de todas as tarefas do processo através do terceiro ícone do painel de acesso rápido do módulo de Execução, como visto na Figura 270. Neste acompanhamento é possível visualizar a data e hora, o estado e o status das tarefas do processo.

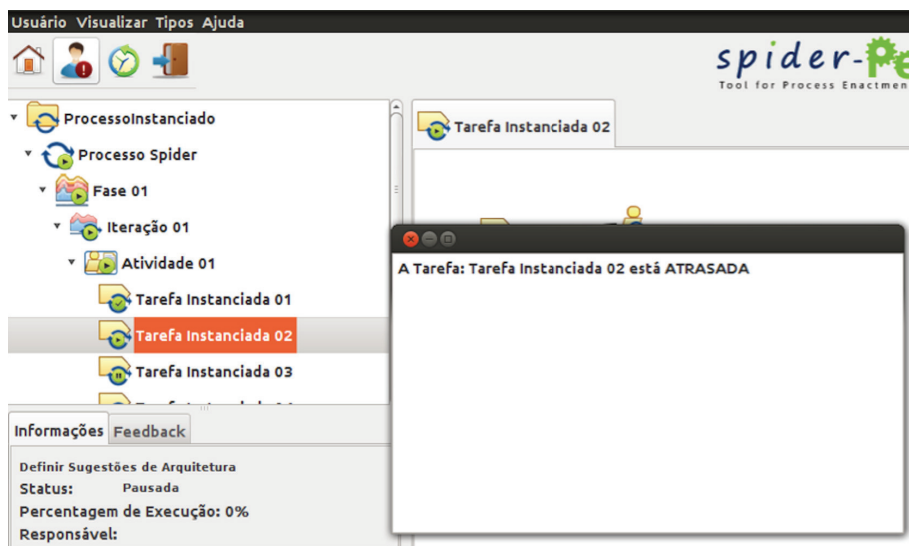


Figura 269. Notificação de Tarefas Atrasadas.

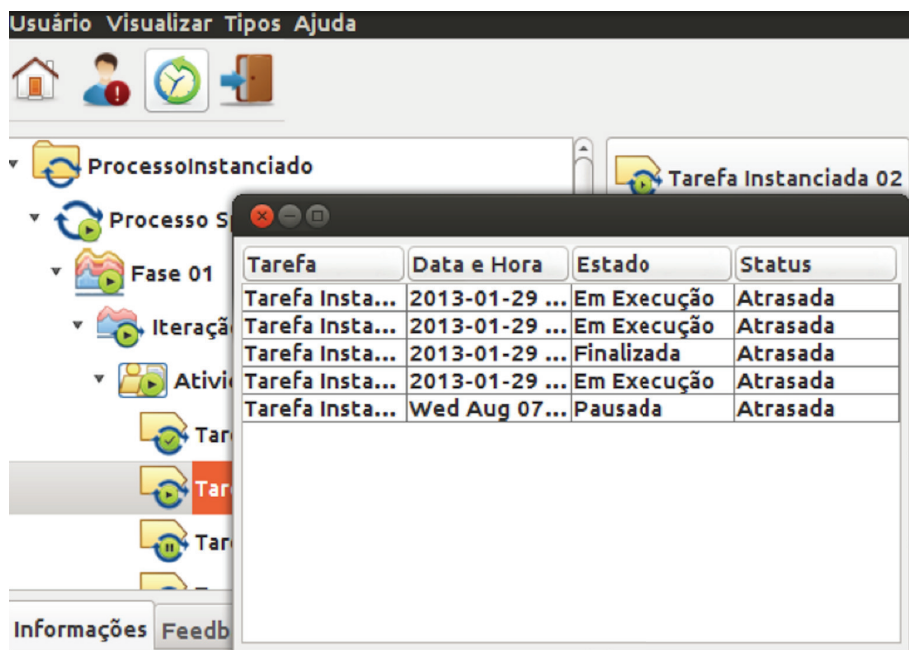


Figura 270. Log de Atividades.

23.3. Aderência da Spider-PE ao CMMI-DEV e MR-MPS-SW

Esta seção objetiva evidenciar o apoio da ferramenta Spider-PE ao atendimento das recomendações equivalentes dos modelos MR-MPS-SW e CMMI-DEV, relacionadas ao conceito de Capacidade. Entende-se por Capacidade do processo o grau de refinamento e institucionalização com que o processo é executado na organização/unidade organizacional [SOFTEX, 2012a].

Esta ferramenta apoia o Nível F do MR-MPS-SW e o Nível 2 do CMMI-DEV, onde o Processo é considerado Gerenciado. A escolha destes níveis específicos deve-se ao fato de que eles são níveis iniciais dos modelos em questão e por isso tendem a ser de maior complexidade para implementação da maturidade organizacional [Oliveira, 2010]. Além disso, nesses níveis os modelos recomendam que o Processo deve ser institucionalizado. Institucionalizar um processo significa colocá-lo em prática na organização,

ou seja, executá-lo. Sendo assim, para cada funcionalidade da Spider-PE, identificaram-se os Resultados de Atributos de Processo (RAP) do MR-MPS-SW e Práticas Genéricas (GP) do CMMI-DEV apoiadas por estas funcionalidades.

Para realizar esta análise de aderência, primeiramente definiram-se níveis de apoio das funcionalidades da Spider-PE em relação às recomendações dos modelos de qualidade, conforme Tabela 7. Para cada nível de apoio é associada uma determinada cor a fim de, posteriormente, representar visualmente estes níveis na análise realizada.

Tabela 23. Níveis de Apoio da Spider-PE ao MR-MPS-SW e CMMI-DEV.

Nível de Apoio	Descrição
Total	As funcionalidades da Spider-PE apoiam às recomendações de um determinado RAP do MR-MPS e da GP correspondente no CMMI-DEV.
Parcial	As funcionalidades da Spider-PE apoiam parcialmente as recomendações de um determinado RAP do MR-MPS e da GP correspondente no CMMI-DEV. Ou seja, estas atividades não atendem a todas as recomendações deste RAP ou desta GP.

A seguir, a Tabela 24 apresenta a relação das funcionalidades da Spider-PE, os RAP do MR-MPS-SW e as GP do CMMI-DEV. Esta relação foi feita a partir da análise dos requisitos necessários para atender as recomendações destes modelos. Desta maneira, compararam-se o quanto as funcionalidades da Spider-PE apoiam às exigências necessárias para atendimento dos RAP do MR-MPS-SW e das GP do CMMI-DEV.

Tabela 24. Análise do Apoio da Spider-PE aos Modelos MR-MPS-SW e CMMI-DEV.

Componente do MR-MPS-SW e CMMI-DEV	Funcionalidades	Módulo
RAP 2 e GP 2.1	Definir Política Organizacional Publicar Política Organizacional	Gerência do Processo
RAP 3 e GP 2.2	Estimar Esforço Definir Cronograma Identificar Riscos Definir Atividades de Monitoramento e Controle	
RAP 3, RAP 5, RAP 6 e GP 2.2, GP 2.3, GP 2.4	Definir Recursos e Responsabilidades	
RAP 7 e GP 2.5	Identificar Habilidades e Capacidade	
RAP 8 e GP 2.7	Planejar Comunicação	
RAP 11	Identificar Requisitos dos Produtos de Trabalho	
RAP 4, RAP 9, RAP10, RAP 14 e GP 2.8, GP 2.9, GP 2.10	Realizar Revisões no Processo	
RAP 1 e GP 1.1	Gerar Produtos de Trabalho	
RAP 12, RAP 13 e GP 2.6	Gerenciar Configurações	
RAP 1 e GP 1.1	Fornecer <i>Feedback</i> Alterar Processo	Execução do Processo



Capítulo 24

RisAgi: Uma Metodologia Ágil para Gestão de Riscos em Projetos de Desenvolvimento de Software

em [Santos et al., 2013]

José Brício M. C. Neto

Kleeson Bruno C. Santos

Paulo Vitor R. Cardoso

Sandro Ronaldo Bezerra Oliveira

O sucesso das organizações desenvolvedoras de software cada vez mais depende da qualidade final dos seus produtos em equilíbrio com o cumprimento dos prazos, custos e escopo estabelecidos, assim como, a minimização das incertezas e maximização das oportunidades que cercam um projeto, tudo isso em um menor tempo possível de desenvolvimento [Gusmão e Moura, 2004]. Nesse contexto, o uso de um bom processo ágil de gestão de projetos alinhado com um bom gerenciamento dos riscos de projeto, faz-se necessário. Contudo, as organizações que procuram aplicar as metodologias ágeis para gerir seus projetos, irão se confrontar com a carência de um gerenciamento mais efetivo e explícito na área de riscos usando estas metodologias. A área de gerenciamento de riscos é uma das mais importantes dentro da gestão de projetos, sendo tratada como uma das nove áreas de conhecimento do guia específico de gestão de projetos, o PMBOK (*Project Management Body of Knowledge*) [PMI, 2013]. Porém, nos *frameworks* ágeis, ela é vagamente citada e definida.

Segundo Cohn (2010), um grande esforço em gestão de riscos de forma explícita torna-se desnecessário quando um projeto utiliza uma abordagem ágil. As iterações curtas, foco em software funcionando, resolução de impedimentos, forte ênfase em testes automatizados, entregas e *feedbacks* frequentes ao cliente ajudam as equipes a evitar o risco que a maioria dos projetos enfrentam, o de não entregar nada. Assim, não é de se estranhar que muitos projetos ágeis renunciem a qualquer forma de Gestão de riscos explícita [Cohn, 2010].

No entanto, essa ideia não é mais aceita como um consenso geral entre muitos “agilistas” (praticantes das metodologias ágeis), principalmente quando projetos de software são complexos, de longa duração e com o escopo grande. Certamente, as incertezas serão maiores. Dessa forma, o uso de uma metodologia ágil para gestão de riscos surge como uma boa opção para as organizações que desejam gerenciar as incertezas dos projetos de forma mais ágil e em conjunto com outra metodologia ágil de gerência de projetos.

Neste contexto, o objetivo deste trabalho é realizar um estudo na área de gestão de riscos focando nos métodos tradicionais e ágeis com intuito de propor uma solução para a seguinte problemática: ausência de gestão explícita de riscos nas metodologias ágeis. Assim, propõem-se uma metodologia ágil para gestão de riscos baseado nos processos de gerenciamento de riscos constantes no PMBOK e nos principais métodos ágeis praticados na indústria de software, possibilitando um gerenciamento dos riscos do projeto de forma mais contundente e explícita, sem ferir os princípios ágeis.

24.1. Gerência de Riscos: Uma Visão Geral

Eventos imprevistos podem causar, em muita das vezes, efeitos catastróficos em projetos de software. Esses imprevistos surgem no momento em que se inicia um projeto, pois todo projeto de software enfrenta problemas referentes, por exemplo, ao escopo, cronograma e qualidade ao decorrer do tempo. As

organizações desenvolvedoras de software possuem muitas dificuldades para, no mínimo, controlar os efeitos negativos desses imprevistos. É nesse contexto que surge o gerenciamento de riscos [Gusmão e Moura, 2004].

Segundo o PMI (2013), a Gerência de riscos é uma das disciplinas dentro da Gerência de projetos a qual consiste num processo sistemático de identificar, analisar e responder aos riscos do projeto, procurando obter vantagem das oportunidades de melhoria sempre que possível. Essa área vem sendo alvo de estudos de vários pesquisadores, devido englobar muitos fatores imprevisíveis e de controle difícil [Ribeiro e Gusmão, 2008].

Pode-se considerar que risco de projeto é um evento ou condição incerta que, se ocorrer, terá um efeito positivo ou negativo em pelo menos um objetivo do projeto [PMI, 2013]. Dessa forma, a Gestão de Riscos visa controlar essas incertezas inerentes a todos os projetos, desde o início dos mesmos.

Segundo Ribeiro e Gusmão (2008), apesar das várias abordagens que tratam da gerência de riscos, existe um consenso sobre as principais atividades que compõem o processo de Gerência de riscos, a saber:

- o Planejamento da Gerência de Riscos, onde são definidas as estratégias do gerenciamento, os recursos necessários e a metodologia adotada;
- a Identificação dos Riscos, onde é realizado um levantamento preliminar de todos os possíveis riscos existentes no projeto e se formaliza a documentação contendo os dados coletados;
- a Análise dos Riscos, onde se definem a probabilidade do risco ocorrer, o possível impacto causado ao projeto e as suas prioridades;
- o Planejamento das Respostas aos Riscos, nesta etapa são determinados os riscos que irão ser gerenciados e as estratégias de ação para resolvê-los, onde são elaborados os planos de ação para controle da gerência;
- o Monitoramento e Controle dos Riscos, há uma observação da efetividade dos planos de ação durante o desenvolvimento do projeto,

tendo uma melhor compreensão do andamento do projeto, avaliando-se a situação atual para localizar possíveis desvios do planejado; e a Comunicação, onde se dá uma visibilidade dos riscos entre a equipe e os membros do projeto.

Essas abordagens fazem parte das metodologias tradicionais de desenvolvimento de software – metodologias orientadas a processos e muita documentação [Ribeiro e Gusmão, 2008], onde a Gerência de riscos é tratada de forma explícita e visível por todos. Na visão do Guia PMBOK a Gerência de Riscos de Projetos inclui os processos de planejamento, identificação, análise, planejamento de respostas e monitoramento e controle dos riscos do projeto, ressaltando que esses processos interagem entre si e com os processos de outras áreas de conhecimento [PMI, 2013]. Ainda neste guia, os objetivos do gerenciamento dos riscos são aumentar a probabilidade e o impacto dos eventos positivos e reduzir a probabilidade e o impacto dos eventos negativos do projeto, a partir do uso de entradas, técnicas e ferramentas, e saídas propostas aos processos.

Por outro lado, recentemente surgiram as metodologias ágeis. Um conjunto de metodologias regidas por princípios comuns onde o foco são as pessoas, a interação entre elas e o feedback rápido [Manifesto Ágil, 2001]. Dessa forma, a área de gerência de riscos foi tratada de forma não explícita, pois muitos adeptos das metodologias ágeis afirmavam que “os *frameworks* ágeis com seus feedbacks constantes, priorização de escopo, monitoramento diário, entregas rápidas, iterações curtas [...] minimizariam os riscos do projeto, assim, resultando em um gerenciamento de riscos intrínseco destes *frameworks*” [Bassi, 2013]. Contudo, essa ideia não é um consenso universal entre os “agilistas”.

Com base no manifesto ágil e nos vários *frameworks* ágeis existentes, como exemplo o Scrum [Schwaber, 2009] e o XP [Beck e Andres, 2005], é possível afirmar que cada um contém ações que visam minimizar a incidência e/ou impacto dos riscos dentro de cada processo. As ideias de *feedback*

constante, e de iterações curtas, permitem que erros sejam identificados mais rapidamente e que sejam resolvidos em pouco tempo [Cohn, 2010].

Não existe um consenso definitivo sobre a necessidade de gestão de risco dentro do método ágil. Isso leva muitos a acreditar que a gestão de risco é irrelevante em um modelo iterativo. Alguns seguem a abordagem de ignorar os riscos até que eles manifestem-se em problemas, pois, a partir desse momento, é possível tratá-los e controlá-los como impedimentos do projeto. Dessa forma, fica evidente que, nos métodos ágeis a preocupação recai na contingência dos impedimentos, deixando de lado a mitigação e o olhar proativo dos mesmos [Veethil, 2013]. Apenas essas práticas ágeis não mitigam a maioria das incertezas do projeto e acabam que, em muitas vezes, lidam apenas com problemas de uma forma mais rápida, o que ocasiona uma interpretação errônea de que qualquer risco é tratável ou tem baixo impacto e que apenas deve ser tratado quando se torna um problema.

24.2. RisAgi: Um Metodologia Ágil para a Gestão de Riscos

O desenvolvimento deste trabalho teve início com uma pesquisa na literatura específica sobre gerência de riscos em projetos de desenvolvimento de software, analisando a visão do PMBOK e das várias metodologias ágeis existentes. Em uma segunda etapa foi analisada a problemática referente à gestão de riscos dentro das metodologias ágeis, com o intuito de propor uma solução para a mesma.

Após essas pesquisas e análises foi definida uma solução de gestão ágil de riscos para projetos de desenvolvimento de software, contemplando a definição de um fluxo de atividades, papéis, cerimônias e um conjunto de produtos de trabalho. Nesse momento, foram realizadas inúmeras reuniões com um implementador e avaliador de processos de software no contexto do MR-MPS-SW para analisar a adequação/avaliação das propostas definidas no cenário do desenvolvimento de software. Vale ressaltar que este

especialista possui experiência em gerenciamento de projetos usando o PMBOK e o Scrum.

24.2.1. A Metodologia Ágil para a Gestão de Riscos

O RisAgi – Riscos Ágeis, é uma metodologia ágil concebida para possibilitar o gerenciamento das incertezas envolvidas em projetos de desenvolvimento de software. Essas incertezas podem ser entendidas como riscos e/ou impedimentos.

A Figura 271 mostra o ciclo de vida do RisAgi. Este ciclo foi definido com base no ciclo de vida dos processos de gestão de riscos do PMBOK. O modo de execução nessas fases segue os valores das metodologias ágeis e se baseia, principalmente, nas práticas encontradas nos métodos ágeis Scrum e APM (*Agile Project Management*) [Highsmith, 2004]. O RisAgi não tem como foco entregar software, mas gerenciar as incertezas dos projetos, visando a melhor qualidade possível do produto final.

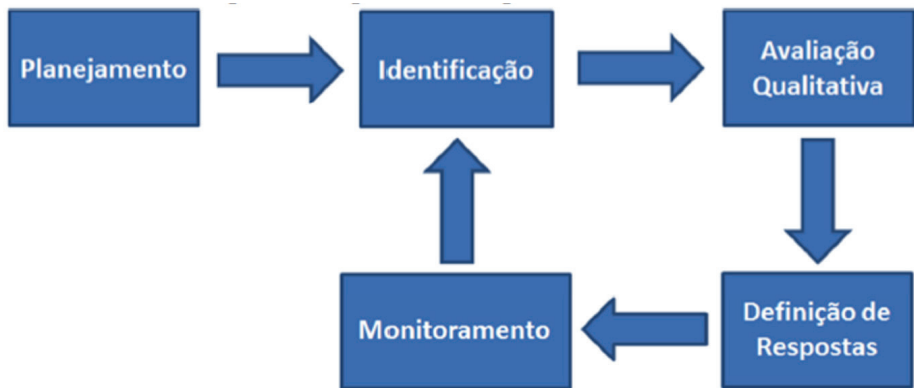


Figura 271. Ciclo de Vida do RisAgi

As fases neste ciclo são:

- Planejamento: trata da apresentação do RisAgi à equipe com intuito de explicar como acontecerá o processo de gestão de riscos no decorrer do projeto de software;
- Identificação: foco no levantamento das incertezas do projeto junto à equipe;
- Avaliação Qualitativa: possibilita a análise dos riscos identificados por meio de critérios subjetivos;
- Definição de Respostas: permite a definição da estratégia de como proceder antes e depois dos riscos tornarem-se uma realidade no projeto;
- Monitoramento: agrupa práticas que possibilitam o acompanhamento do processo de gestão de riscos ao longo do projeto.

a) Papéis

O RisAgi é composto por dois papéis, o Guardião e a Equipe de Risco, cada um com responsabilidades bem definidas, a saber:

- Guardião: responsável por garantir que o RisAgi seja entendido e aplicado. Atua organizando, direcionando e objetivando as cerimônias do RisAgi, além de acompanhar e monitorar as atividades de gerenciamento de risco realizadas pela Equipe de Risco. É a interface entre os *stakeholders*. Características deste papel: conhecer os interesses do cliente; conhecer os interesses da organização; disseminar a visão, o objetivo e os requisitos do produto para a Equipe de Risco; garantir que o processo seja seguido; orientar a equipe de risco; definir a lista de Categorias dos Riscos no projeto; criar e monitorar o *Burnup* de Riscos, gráfico responsável por demonstrar a severidade dos riscos ocorridos ao longo do tempo;

- **Equipe de Risco:** consiste de profissionais que realizam o trabalho de identificar, avaliar e prover respostas para mitigar e contingenciar os riscos que impactam nos objetivos do projeto e suas iterações. A Equipe de Risco é estruturada e autorizada pela organização para organizar e gerenciar seu próprio trabalho. A sinergia resultante aperfeiçoa a eficiência e a eficácia da Equipe de Risco como um todo. Este equipe deve ter as seguintes características: deve ser multifuncional, possuindo todas as habilidades necessárias, enquanto equipe, para gerenciar riscos; todos os membros desta equipe devem possuir o mesmo poder e nível hierárquico independente dos títulos de cada membro da Equipe de Risco; não deve conter subequipes dedicadas a domínios específicos de conhecimento, tais como identificação ou avaliação ou resposta aos riscos, assim, os membros devem participar de todo o processo; definir a lista de Categorias dos Risco no projeto; identificar os riscos; avaliar os riscos; prover respostas aos riscos; assumir a responsabilidade pela contingência e/ou mitigação de riscos no projeto.

b) Cerimônias

O ciclo do RisAgi é composto por eventos prescritos que criam uma rotina e minimizam a necessidade de reuniões não definidas no *framework*. O RisAgi usa eventos *time-boxed*, onde todo evento tem uma duração máxima. Isto garante que uma quantidade adequada de tempo seja gasta no planejamento.

b.1) Iniciação

A Iniciação é uma cerimônia que ocorre na fase de Planejamento do RisAgi, com duração máxima de quatro horas, objetivando apresentar à Equipe de Riscos suas responsabilidades, as cerimônias e os produtos de trabalho que devem ser gerados a partir do uso do RisAgi. Responsabiliza-se por esta

cerimônia o Guardião, que inicia a cerimônia apresentando à Equipe de Riscos toda a metodologia do RisAgi; os principais interesses da organização concernente ao projeto; os interesses do cliente; e uma visão geral das outras áreas que compõem o plano do projeto (escopo, prazo, custo, recursos, qualidade, etc.).

Como entradas para a cerimônia de Iniciação têm-se a descrição da metodologia, o plano do projeto e quaisquer outros ativos da organização que possam influenciar esta etapa do RisAgi, por exemplo uma lista de categorias de riscos iniciais a qual a organização mantém.

Ao final, a equipe gera uma Lista de Categorias de riscos para o presente projeto, disposto no Quadro de Riscos conforme a Figura 272.

b.2) Jogo de Risco

Esta cerimônia ocorre ao longo das fases de Identificação, Avaliação Qualitativa e Definição de Resposta, possui um *time-box* de um a dois dias para cada release de um mês ou de dois a quatro dias para releases de dois meses. Tem como objetivo responder a três questões: Quais são os riscos do projeto?; Qual a severidade destes riscos?; Como estes riscos serão tratados? Estes questionamentos são respondidos em etapas da mesma cerimônia, a saber:

- Etapa I : Quais são os riscos do projeto?: tem como objetivo identificar os riscos do projeto e registrá-los. O Guardião irá conduzir a primeira fase da cerimônia apresentando de forma mais detalhada o plano do projeto, os interesses do cliente e da organização para a Equipe de Riscos. O intuito é estimular uma discussão que possibilite identificar riscos do projeto com base nas categorias definidas anteriormente na reunião de iniciação. A técnica utilizada para essa fase da cerimônia pode ser um *brainstorm*, onde cada risco deve ser justificado pelo membro da equipe que identificou o mesmo. Dessa forma, somente encerrará essa fase após se obter um consenso

sobre os riscos identificados entre todos os membros da Equipe. Ao fim dela é produzido o *Backlog* de Riscos do projeto, disposto no Quadro de Riscos apresentado na Figura 272;

- Etapa II : Qual a severidade destes riscos?: a severidade dos riscos é obtida pela multiplicação do impacto com a probabilidade dos mesmos. A equipe irá avaliar cada item do *Backlog* de Risco de forma qualitativa quanto à probabilidade e impacto. Assim, no decorrer da avaliação o Quadro de Riscos será preenchido. A avaliação é realizada para dar prioridade aos riscos mais importantes, geralmente os que possuem maior severidade aos olhos da equipe de desenvolvimento, visando agregar oportunidades que ofereçam mais retorno de investimento tanto para o cliente quanto para o fornecedor e minimizar ou extinguir ameaças e fraquezas que possam vir a acontecer no decorrer do projeto. Esta etapa utiliza uma adaptação da técnica *Planning Poker* [Cohn, 2005], uma técnica ágil de estimativa e planejamento baseada em consenso [Cohn, 2005]. Esta técnica combina a opinião de especialistas, a analogia e a desagregação em uma abordagem agradável para estimar resultando em estimativas rápidas, mas de confiança [Cohn, 2005]. Esta adaptação segue os passos descritos abaixo:
 - Para pontuar os itens do *Backlog* de Riscos será utilizado um baralho de *Planning Poker* customizado, que deverá conter as seguintes qualificações: baixo, médio e alto;
 - Caso seja necessário obter um valor mais detalhado de severidade para cada risco, poderá ser utilizado um baralho numérico que represente as qualificações, por exemplo: 1, 2 e 3 (baixo); 4, 5 e 6 (médio); 7, 8 e 9 (alto);
 - Procedimentos do *Planning Poker*: (1) a equipe em conjunto lê a descrição de todos os riscos do *Backlog* de Risco para ter uma visão geral do que deverá ser avaliado; (2) a equipe

seleciona o risco que julga ser mais simples de todos, isto é, o que apresenta menor severidade ao projeto. Para este risco a estimativa será de Baixo impacto e probabilidade, que representa a menor qualificação do baralho; (3) o primeiro risco avaliado servirá de guia para todo o processo de avaliação, ou seja, será uma referência para a avaliação dos demais riscos; (4) seguindo a ordem apresentada no *Backlog* de Risco, cada item será relido e avaliado. É necessário ressaltar que um risco que tenha uma severidade maior que a do risco-guia não necessariamente será avaliado com o valor seguinte na escala de qualificações. A severidade deve ser avaliada seguindo uma proporção da severidade que foi definida para o risco-guia; (5) à medida que os riscos sejam avaliados, os mesmos devem ser colados no artefato Quadro de Risco (ver Figura 2) em seu devido quadrante;

- Etapa III : Como estes riscos serão tratados?: realizada a avaliação qualitativa, e com o quadro preenchido com os riscos, a Equipe deverá definir as respostas de mitigação e contingência dos riscos. Essas respostas devem classificar os riscos como oportunidades ou ameaças. Para melhor visualização, as oportunidades devem ser evidenciadas por meio de *post-its* na cor verde e as ameaças na cor vermelha. Nesse momento, realiza-se uma reunião onde todos estão de pé e em frente ao Quadro de Riscos. Utilizando-se da técnica de Opinião Especializada, cada membro da Equipe de Riscos define as respostas de mitigação e contingência relacionadas àquele risco sob sua responsabilidade. Essas respostas devem ser inseridas ao lado dos seus respectivos riscos, assim como, cada membro da Equipe deve se responsabilizar por essas respostas, evidenciando seu comprometimento ao escrever seu nome nos *post-its*. Ao final dessa fase, têm-se os cartões de contingência e mitigação preenchidos.

b.3) Monitorar e Revisar

Esta cerimônia ocorre na fase de Monitoramento, e deve ser realizada pelo Guardião e pela Equipe de Risco, os quais devem monitorar e revisar os riscos levantados na iteração. O *Burnup* de Risco (ver Figura 274) é utilizado como um artefato de apoio para esta prática de acompanhamento.

À medida que um risco ocorrer ao longo do projeto, o Guardião deve atualizar o gráfico do *Burnup* de Riscos. Para isto, ele deverá registrar no gráfico a data de ocorrência do risco e o número da severidade do risco, severidade esta obtida pela multiplicação da probabilidade com o impacto do risco. Assim, registra-se em um ponto no gráfico a severidade do risco que acabara de ocorrer. Dessa forma, esta prática pode transparecer para todos a ocorrência de um risco previsto ao longo do tempo. As informações que podem ser extraídas do gráfico são úteis para avaliar a incidência de riscos, o trabalho da equipe quanto à contingência e a mitigação para que estes não ocorram ou não se repitam.

c) Produtos de Trabalho

A metodologia RisAgi gera produtos de trabalho a partir da execução de suas cerimônias. Esses artefatos estão fortemente baseados nos princípios dos métodos ágeis, os quais têm como foco as pessoas e a iteração entre elas. Dessa forma, cada artefato foi minuciosamente gerado para agilizar o andamento do processo e não travá-lo. Os produtos de trabalho são:

- **Quadro de Riscos:** o Quadro de Riscos (ver Figura 272) tem o propósito de dar transparência a todos os envolvidos no processo de gerenciamento de riscos, facilitar o monitoramento e a visualização da severidade dos riscos ao longo do projeto. A Equipe de Risco é responsável por inserir os riscos analisados qualitativamente quanto à probabilidade e ao impacto em seus respectivos quadrantes. Em

um dos lados do Quadro existe uma seção complementar, o *Backlog* de Riscos (a ser apresentado a seguir); enquanto que do outro lado há a Lista de Categorias (a ser apresentado a seguir) dos riscos;

Backlog de Risco	Impacto	Alto	Médio	Baixo	Lista de Categorias
	Probabilidade				
	Alto				
	Médio				
	Baixo				

Figura 272. Quadro de Riscos do RisAgi

- **Lista de Categorias:** é uma lista que contém todas as categorias, as quais servem para agrupar os riscos, relevantes à Equipe de Risco para analisar as incertezas do projeto de software. A Lista de Categorias é criada, mantida e disponibilizada pela Equipe de Risco. O principal objetivo é nortear as análises realizadas durante o ciclo do RisAgi de acordo com os interesses da organização e do cliente. Cada categoria deverá seguir o formato apresentado na Figura 273(a) e assim possuir um título e uma descrição objetiva. Esse artefato fica disposto no Quadro de Riscos (ver Figura 272). É aconselhável à organização manter uma lista de categorias pré-estabelecidas, ou melhor, padronizadas com base em projetos anteriores como um ativo organizacional disposto numa ferramenta *Wiki*, por exemplo. Assim, esse artefato serviria como insumo para a cerimônia de Iniciação do RisAgi, com o objetivo de ser adaptado à realidade de cada projeto;
- **Backlog de Risco:** trata-se de uma lista que contém todas as incertezas de um projeto identificadas pela Equipe de Risco ainda não avaliadas qualitativamente. É uma origem única dos riscos do projeto.

Este artefato geralmente não está completo no início de um projeto, pode-se começar com uma lista de riscos iniciais para aquele primeiro momento. Com o tempo, o *Backlog* de Risco cresce e muda à medida que se aprende mais sobre o projeto, o produto e seus usuários. A Equipe de Risco deve manter o *Backlog* de Risco atualizado e disponível a todos os interessados no projeto. O risco é registrado num *post-it* conforme o *template* evidenciado pela Figura 273(b). Nesse *post-it* deve-se, inicialmente, registrar um identificador, um nome, uma descrição e uma categoria para o risco. E, posteriormente, um membro da Equipe de Risco deve assinalar com seu nome qual risco fica sobre sua responsabilidade;

Titulo da Categoria	ID	Titulo do Risco	Categoria
Descrição da Categoria	Descrição do Risco		

(a)

(b)

Figura 273. (a) *Template* para Registro de uma Categoria de Riscos; (b) *Template* de Registro dos Riscos

- *Burnup* de Risco: é uma representação gráfica da severidade dos riscos, utilizado para monitorar a severidade dos riscos ao longo do projeto. A partir do início de um projeto, e à medida que os riscos tornam-se realidade, deve-se atualizar o *Burnup*. O Guardião é o responsável pela criação e manutenção deste artefato. Qualquer significativa mudança no gráfico, o Guardião deve acionar a Equipe de Risco para realizar uma análise do ocorrido. No eixo horizontal têm-se os dias em que cada risco ocorreu e no eixo vertical têm-se a severidade dos riscos ocorridos, bem como a tolerância aos riscos e a soma das estimativas de severidade dos riscos. Este artefato pode ser visto na Figura 274;

- **Cartão de Contingência e de Mitigação:** descreve as ações a serem tomadas para contingenciar os riscos e para diminuir ou aumentar a probabilidade ou impacto da ocorrência dos riscos, respectivamente. A Equipe de Risco é responsável por escrever os Cartões de Contingência, dando maior detalhamento àqueles que possuem maior severidade. Os Cartões de Contingência visam responder a três questões fundamentais: Quando o plano deve ser acionado?; O que será feito para contingenciar o risco?; Como será contingenciado? Já os Cartões de Mitigação visam responder a três questões fundamentais: O que será feito?; Como será feito?; Porque será feito? Os Cartões de Contingência e de Mitigação devem estar disponíveis e ser de conhecimento de todos os membros da Equipe de Risco, porém é de responsabilidade de um único membro da Equipe, quem se comprometeu com o risco que essa resposta pertence. Os *templates* dos cartões podem ser vistos na Figura 275.

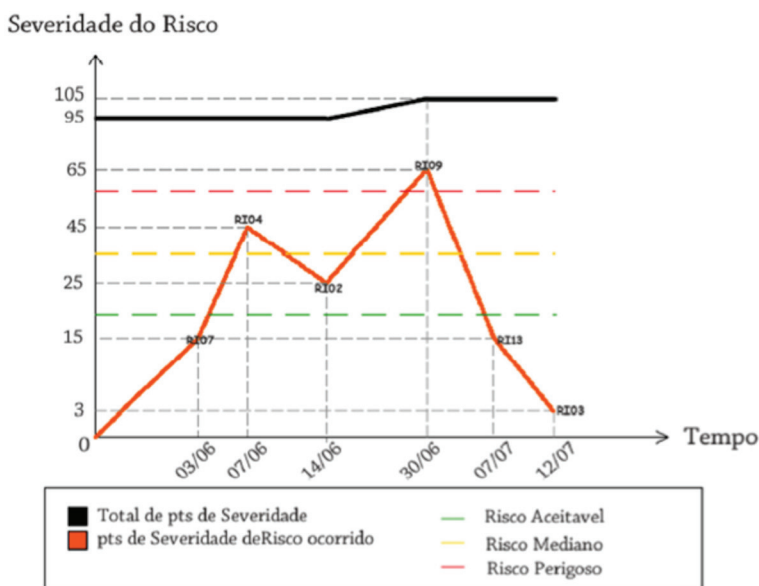


Figura 274. Burnup para Registro e Acompanhamento da Severidade dos Riscos

ID	#Contingência	ID	#Mitigação
	Quando o plano deve ser acionado?		O que será feito?
	O que será feito para contingenciar o risco?		Como será feito?
	Como será contingenciado?		Porque será feito?

Figura 275. Cartões de Contingência e Mitigação como Resposta aos Riscos



Capítulo 25

Um *Framework* de Processo para Gerência de Portfólio de Projetos de Software de Acordo com Padrões de Qualidade

em [Souza e Oliveira, 2013]

Maurício Ronny de Almeida Souza

Sandro Ronaldo Bezerra Oliveira

É comum a adoção de normas, padrões e modelos de qualidade para apoiar a definição de práticas gerenciais em organizações que procuram a melhoria de seus processos, aumento de eficiência, controle sobre seus meios de produção, ou que almejam diferenciais competitivos para fortalecimento do negócio. No cenário da indústria de software brasileira, esse movimento é crescente, uma vez que grande parte das empresas partiu em busca destas certificações e avaliações, para se manterem competitivas no mercado [Guerra e Alves, 2004].

No entanto, as práticas gerenciais não podem estar restritas ao escopo de projetos. Além de gerenciar de forma eficiente seus projetos, os executivos de organizações têm o desafio de garantir que os recursos organizacionais estejam sendo direcionados para os projetos que melhor contribuam ao alcance dos objetivos estratégicos de suas organizações [Yelin, 2007], uma vez que tempo, recursos e orçamento são variáveis, que entre outras, limitam o número de projetos que podem ser conduzidos.

Desta forma, as organizações devem preocupar-se também com questões ligadas à lucratividade, retorno sobre investimento, entrega de benefícios e vantagens sobre janelas de oportunidade [Levine, 2005]. Assim, cabe à Gerência de Portfólio de Projetos (GPP) manter um nível de controle mais abrangente sobre os componentes que constituem a carteira ou portfólio de projetos das organizações, definindo quais projetos devem ser conduzidos, e acompanhando o sucesso dos mesmos, à luz dos objetivos e estratégia de negócio da organização [Rad e Levin, 2006], destinando recursos para projetos que tragam maiores benefícios [Yelin, 2007].

Um reflexo da crescente importância desta atividade gerencial é o crescente número de modelos criados, ou que incluíram a gestão de portfólio de projetos entre suas práticas, para apoiar e recomendar práticas de gestão de portfólio, conforme identificados por [Costa *et al.*, 2010a]: (i) *Project Portfolio Management Maturity Model* (PPMMM); (ii) *Portfolio, Programme, and Project Management Maturity Model* (P3M3); (iii) *Programme, and Portfolio Management Maturity Model* (P2M3); (iv) Padrão para Gerência de Portfólio do PMI; (v) *Organizational Project Management Maturity Model* (OPM3); (vi) ISO 12207:2008; e o (vii) Programa de Melhoria do Processo de Software Brasileiro (MPS.BR).

Sabe-se, contudo, que a implantação da melhoria de processo de software a partir da adoção de padrões de qualidade pode ser custosa, tanto em relação a investimento financeiro quanto ao tempo necessário [Oliveira *et al.*, 2011].

Conforme observado em [Souza *et al.*, 2012a] e [Costa *et al.*, 2010a], o cenário nacional ainda é marcado pela baixa maturidade no que se refere à gestão de portfólio de projetos, considerando o contexto da indústria de software. Nos resultados de aplicação de questionários da pesquisa conduzida e publicada em [Souza *et al.*, 2012a], pelo mesmo grupo da pesquisa, observa-se que, embora práticas relacionadas às atividades de gestão de portfólio de projetos estejam presentes no cotidiano de organizações de diferentes tamanhos, independente de serem organizações avaliadas ou não em modelos de qualidade, estas práticas carecem de padronização e formalização.

Nesse contexto, este trabalho tem o objetivo de apresentar um *Framework* de Processo para Gestão de Portfólio de Projetos de Software, em conformidade com práticas recomendadas, para área gerencial em questão, pelos padrões: MR-MPS-SW, ISO/IEC 12207 e Padrão para Gestão de Portfólio do PMI [PMI, 2008]. Desenvolvido no contexto de um trabalho de dissertação de Mestrado da Universidade Federal do Pará, este framework de processo consolida práticas relacionadas ao processo em questão, sugeridas pela literatura técnica especializada e pelos padrões mencionados, sob a forma de uma ferramenta de apoio à definição dos processos organizacionais para a gestão de portfólio de projetos, no contexto específico de projetos de desenvolvimento de software.

Por *framework* entende-se um modelo básico de processo a ser configurado e adaptado para os projetos [Falbo, 2000], a exemplo do *framework* do RUP (*Rational Unified Process*) e o proposto pelo Padrão para Gestão de Portfólio de Projetos [10], compostos por um conjunto de atividades, tarefas e recomendações que podem ser utilizados na concepção de um processo, adaptando-os conforme as necessidades organizacionais ou de projeto.

Inserido no contexto do Projeto SPIDER – Software Process Improvement: DEvelopment and Research [Oliveira *et al.*, 2011], este *framework* de processo configura como um instrumento proposto para facilitar a implantação de programas de melhoria de processo (neste caso, no que tange ao processo de gerência de portfólio de projetos) e como objeto de estudo para pesquisadores ou profissionais interessados no aprendizado a respeito da gestão de portfólio de projetos, apresentando uma estrutura de referência.

25.1. Trabalhos Relacionados

Os tópicos centrais deste trabalho estão relacionados à especificação de uma estrutura de apoio à definição de processos de gerência de portfólio de

projetos, e à adequação destes processos às recomendações dos padrões: MR-MPS-SW, ISO/IEC 12207 e Padrão para Gestão de Portfólio do PMI.

Assim, entre os trabalhos relacionados identificados nas etapas iniciais da definição da pesquisa, foram identificados principalmente trabalhos cujo foco era a concepção de processos ou metodologias para apoio à gestão de portfólio de projetos em conformidade com modelos de qualidade. Não foram identificadas outras propostas de definição de estruturas semelhantes ao framework aqui proposto.

Em [Correia, 2005] é proposto um modelo para Gestão de Portfólio de Projetos de Software, o “Portfolius”, que atua na seleção e priorização de projetos de desenvolvimento de software, considerando variáveis provenientes dos níveis estratégico, tático e operacional da organização.

A pesquisa conduzida em [Correia, 2005] é relevante para a pesquisa aqui proposta por fazer um levantamento de diversos modelos de gerenciamento de portfólio, porém, devido à época de sua realização, não foi possível realizar um mapeamento com modelos e normas específicos para processos de desenvolvimento de software.

Em [Souza *et al.*, 2009] é proposto um processo para apoiar a gerência estratégica de portfólio, por meio da identificação das regras de governança do portfólio, identificação e categorização dos projetos, seleção, priorização e balanceamento do portfólio, manutenção do pipeline (projetos em execução) e encerramento dos projetos. O processo foi construído com base nas melhores práticas de gerência de projetos, programa e portfólio encontradas na literatura e está aderente ao processo de gerência de portfólio de projetos da norma ISO/IEC 12207:2008. Este artigo foi fruto da dissertação de mestrado intitulada “Uma Abordagem para Gerência Estratégica de Portfólio: Foco da Seleção de Projetos” [Souza, 2008], onde propõe também o ferramental para apoiar o processo desenvolvido.

Uma limitação destes trabalhos é a inexistência do processo de GPP no MPS.BR, à época do desenvolvimento dos mesmos. Portanto, seria necessário realizar uma análise de aderência em relação à Gerência de Portfólio

de Projetos no contexto do MR-MPS-SW. Além disso, a ferramenta que fora proposta para implementar e validar o modelo, a Taba *Portfolio Management* (desenvolvida para a Estação TABA, um ambiente centrado no processo), encontra-se descontinuada.

Em [Costa *et al.*, 2010b] e [Costa, 2011] é descrito um modelo de processo (e uma ferramenta de apoio para validar o modelo) para a seleção e otimização de portfólios baseada na Moderna Teoria de Portfólio de Markowitz [Markowitz 1952].

A proposta do *framework* aqui definida tem os diferenciais:

- Contexto de organizações desenvolvedoras de software: as recomendações propostas no detalhamento dos componentes do *framework* apresentam apenas abordagens que foram propostas ou aplicadas para o contexto da gestão de portfólio de projetos de software;
- Aderência aos padrões MR-MPS-SW, ISO/IEC 12207 e Padrão para Gestão de Portfólio do PMI;
- Abrangência às principais atividades da gestão de portfólio: seleção, avaliação e seleção de projetos; balanceamento de portfólio; alocação estratégica de recursos; revisão do portfólio e outros.

25.2. Metodologia

A definição do *framework* de processo foi conduzida em conformidade com as seguintes etapas:

- a) Identificação e análise de padrões, modelos e normas de qualidade para a Gestão de Portfólio de Projetos: Neste contexto, foram selecionados o modelo MR-MPS-SW e a Norma ISO/IEC 12207, devido a sua relevância no contexto da melhoria de processo de software, e o Padrão para Gerência de Portfólio mantido pelo PMI,

por ser um padrão específico para a gestão de portfólio e que condensa os conhecimentos disseminados no corpo de conhecimentos do PMBOK (*Project Management Body of Knowledge*) e no modelo de maturidade OPM3, mantidos pelo PMI, instituto referência em gestão de projeto;

- b) Mapeamento de práticas sugeridas para a gestão de portfólio de projetos: A partir da análise da recomendação dos padrões citados, fez-se um mapeamento de boas práticas sugeridas, identificando um conjunto de práticas comuns aos padrões e práticas que fossem complementares entre estes. Este mapeamento foi definido e publicado em [Souza *et al.*, 2011], totalizando 24 Boas Práticas, que serviram de base para a definição do *framework* de processo proposto. Este mapeamento foi definido pelo mesmo grupo da pesquisa;
- c) Definição das fases, tarefas e atividades do *Framework* de Processo: A partir das Boas Práticas Identificadas em (b), foi proposta uma estrutura para o *Framework* de Processo;
- d) Condução de um Mapeamento Sistemático da Literatura para identificar abordagens de implementação para as Boas Práticas identificadas: A partir de um estudo baseado em Revisão Sistemática, foram identificadas abordagens (ferramentas, metodologias, técnicas, ativos de processo, entre outros) para apoiar a implementação das atividades de Gestão de Portfólio de Projetos de Software;
- e) Detalhamento dos componentes do *Framework* de Processo: A partir dos resultados da etapa (d), foi definido um conjunto de recomendações para a implementação das tarefas do *framework* de processo.

Assim, o *Framework* de Processo para a Gerência de Portfólio de Projetos de Software definido neste trabalho é composto por fases, atividades

e tarefas, organizadas em um fluxo sugerido. Cada componente é caracterizado por um objetivo e, no caso das tarefas, por um conjunto de recomendações dos próprios padrões de qualidade mapeados e dos resultados do estudo baseado em Revisão Sistemática da Literatura. Diferente de um processo, este *Framework* não é detalhado em termos de passos, procedimentos, atores e artefatos, mas sim por um conjunto de recomendações que possam apoiar a definição de um processo de gerência de portfólio de projetos de software.

25.3. Framework de Processo para Gestão de Portfólio de Projetos de Software

Conforme a metodologia definida, o *Framework* de Processo para Gerência de Portfólio de Projetos de Software foi definido, observando as seguintes características nas subseções a seguir.

25.3.1. Objetivos

Este *Framework* de Processo tem o objetivo de fornecer uma estrutura de referência para a definição de processos de gerência de portfólio de projetos, especificamente associada ao contexto de projetos de desenvolvimento de software.

Esta estrutura tem como principal característica a flexibilidade, isto é, a possibilidade de adequação conforme necessidades organizacionais ou ao contexto para o qual se pretende definir um processo de gestão de portfólio. Isto é alcançado por dois fatores:

- a) Uma vez que o *framework* foi definido a partir de um mapeamento de boas práticas de padrões de qualidade, para cada tarefa, existe uma descrição a respeito da importância desta em relação a cada

padrão. Assim existem detalhamentos a respeito dos impactos de omissão de tarefas;

- b) O *Framework* aponta um conjunto de recomendações de diferentes abordagens para implementação de cada tarefa, oriundas das sugestões dos documentos oficiais de cada padrão e de uma análise da literatura, por meio de um estudo baseado em revisão sistemática. Assim, o *Framework* não define uma única estratégia para a implantação de um processo, mas sim aponta diretrizes que podem ser utilizadas para tal;
- c) O fluxo definido para o *Framework*, bem como suas tarefas e atividades, são sugestões. Cada componente é descrito em relação aos seu objetivo no *framework*, havendo possibilidade de omiti-los, uni-los ou dividi-los, conforme a necessidade de processos mais ou menos complexos.

25.3.2. Limitações

O *Framework* de Processo descrito neste capítulo está limitado às atividades de gestão de portfólio de projetos de software.

Muito embora sejam apresentadas recomendações dos padrões e da literatura a respeito de formas de implementar os componentes do *framework*, estas são apenas direcionamentos e não são necessariamente suficientes para o entendimento de determinada abordagem sugerida. Um exemplo do detalhamento de uma tarefa é apresentado na seção a seguir.

Não é possível utilizar diretamente este *framework* como um processo, haja vista que não são descritos passos, procedimentos, atores e artefatos. Este *framework* limita-se a fornecer uma visão das atividades e tarefas que podem compor um processo de gestão de portfólio, quais os resultados e entradas esperadas de cada tarefa, e sugere um fluxo (que pode ser alterado) para os componentes.

25.3.3. Estrutura

O *Framework* de Processo definido foi organizado em quatro fases: (i) Planejamento, (ii) Identificação e Seleção de Projetos Candidatos, (iii) Seleção do Portfólio e (iv) Monitoramento do Portfólio (vide Figura 276). Cada um destes elementos é composto por um fluxo de atividades, que por sua vez são compostas por fluxos de tarefas.

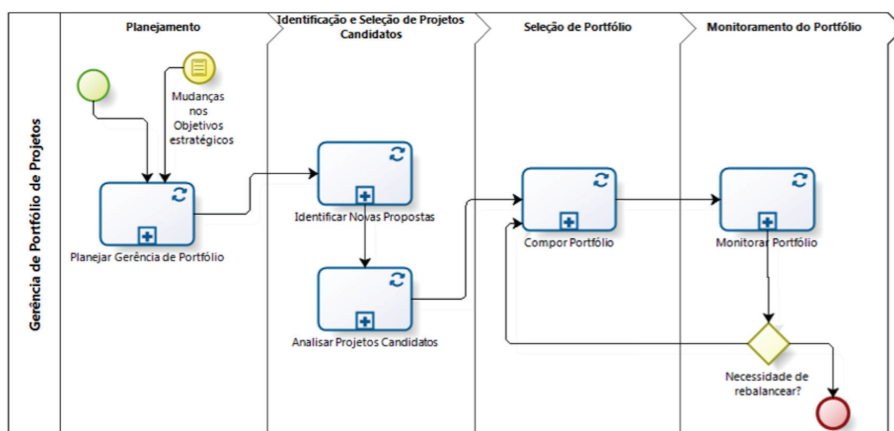


Figura 276. Macro fluxo do *Framework* de Processo para Gerência de Portfólio de Projetos de Software

Além da descrição de seus objetivos, cada tarefa é descrita em termos de:

- Boas Práticas Relacionadas: mapeamento em relação às Boas Práticas [Souza *et al.*, 2011] que originaram a tarefa;
- Entradas e Resultados Sugeridos: Indicações de possíveis dados e informações consumidas e produzidas pela tarefa, de forma que seja possível propor artefatos, a partir destas recomendações;
- Recomendações segundo os Padrões e segundo a Revisão Sistemática da Literatura: Uma síntese de abordagens sugeridas para implementação das tarefas;

- d) Observações sobre o uso do item do Framework: Descrição da relevância do componente ao *framework* e um detalhamento da importância da tarefa em relação a conformidade aos padrões aos quais este *framework* é baseado.

Para fins deste trabalho, as fases serão descritas nas Seções a seguir, em termos dos objetivos das atividades e tarefas. Posteriormente, é apresentado apenas um exemplo de detalhamento de uma tarefa, conforme os itens supracitados (o detalhamento completo do *framework* encontra-se disponível em http://www.spider.ufpa.br/projetos/spider_portfolio/PPGCC_DissertacaoMauricioSouza.pdf).

a) Fase “Planejamento”

A fase “Planejamento” corresponde à etapa de definição das diretrizes do processo de gestão de portfólio de projetos. Esta fase é constituída pela atividade “Planejar Gestão de Portfólio”.

a.1) Planejar Gestão de Portfólio:

O objetivo desta atividade é planejar todo o processo de GPP quanto a ferramentas, métodos, critérios e responsáveis que irão nortear todas as demais fases. A partir desta tarefa pode ser definido um plano ou um conjunto de documentos definindo as diretrizes de execução da gestão de portfólio da organização. Suas tarefas incluem: “Analisar Objetivos e Estratégias Organizacionais Para a Gestão de Portfólio”, “Definir Categorias de Projetos”, “Identificar Recursos Organizacionais”, “Definir Procedimentos e Critérios Para Avaliação de Projetos Candidatos”, “Definir Procedimentos e Critérios Para Seleção de Projetos Candidatos”, “Definir Procedimentos e Critérios Para Priorização de Projetos Selecionados”, “Definir Procedimentos e Critérios Para o Balanceamento do Portfólio” e “Definir Procedimentos e

Critérios Para o Acompanhamento do Portfólio”. Cada uma destas tarefas é descrita a seguir:

- 1) Tarefa “Analisar Objetivos e Estratégias Organizacionais Para a Gerência de Portfólio”: esta tarefa tem o propósito de extrair das fontes apropriadas os objetivos organizacionais que devem ser levadas em consideração para a avaliação e seleção de projetos e para a composição do portfólio de projetos da organização. Assim, esta tarefa é importante para promover a ligação entre a seleção de projetos e a estratégia organizacional;
- 2) Tarefa “Definir Categorias de Projetos”: esta tarefa tem o objetivo de definir as categorias de projetos da organização. Esta tarefa está diretamente relacionada à tarefa “Categorizar Projetos”, definindo diretrizes suficientes para possibilitar a distinção de projetos em diferentes categorias. As categorias e os critérios para categorização de projetos devem ser registrados (um plano específico ou um segmento de um plano para a gerência de portfólio de projetos);
- 3) Tarefa “Identificar Recursos Organizacionais”: esta tarefa é relevante para estabelecer um levantamento dos recursos organizacionais, possibilitando o posterior mapeamento dos recursos disponíveis e alocados para fins de definir estratégias de alocação de recursos e para verificação da capacidade de execução de novos projetos;
- 4) Tarefa “Definir Procedimentos e Critérios Para Avaliação de Projetos Candidatos”: o objetivo desta tarefa é definir a estratégia para avaliar os projetos candidatos conforme forem identificados. É importante que esta estratégia permita que diferentes projetos sejam avaliados através de um sistema padrão de tal forma que seja possível compará-los, e estabelecer um mecanismo que permita aos tomadores de decisão identificar e justificar o grau de relevância de um projeto para a organização. É importante lembrar que a estratégia de avaliação dos projetos candidatos deve estar em harmonia

com a estratégia organizacional, garantindo que a análise dos projetos reflita sua relevância em relação à viabilidade de apoiar o alcance dos objetivos estratégicos. Nesta tarefa deve ser definido o mecanismo pelo qual os projetos serão avaliados (ferramentas, técnicas ou procedimentos) e, caso necessário, os critérios que deverão ser aplicados ou especialistas consultados durante a avaliação. Também é relevante que seja descrito como a informação proveniente da avaliação deve ser interpretada, permitindo o entendimento dos resultados desta de forma clara e não ambígua;

- 5) Tarefa “Definir Procedimentos e Critérios Para Seleção de Projetos Candidatos”: esta tarefa tem o objetivo de definir as diretrizes para filtrar os projetos que de fato apresentam real potencial de atender as demandas estratégicas da organização. Assim, devem ser definidas as estratégias, ferramentas, técnicas e procedimentos envolvidos na seleção dos projetos candidatos avaliados, e, caso necessário, definir os critérios e especialistas consultados para realizar esta seleção. Novamente, a seleção está condicionada aos objetivos estratégicos organizacionais, e não caracteriza a inclusão no portfólio, mas sim uma seleção prévia dos projetos que tem potencial de serem inclusos conforme a capacidade de execução da organização;
- 6) Tarefa “Definir Procedimentos e Critérios Para Priorização de Projetos Seleccionados”: esta tarefa tem o objetivo de definir as diretrizes para priorizar os projetos selecionados sempre que necessário. Para isto, devem ser definidas as estratégias, ferramentas, técnicas e procedimentos envolvidos na priorização dos projetos selecionados, definindo os critérios, e especialistas consultados se for o caso, para tal. A priorização do projeto deve estar em harmonia com a estratégia da organização e levar em consideração os resultados da etapa de avaliação dos projetos candidatos. É importante definir em que momentos esta priorização deve ocorrer, e quaisquer parâmetros

necessários para estabelecer a prioridade entre projetos (como ordem de importância, escalas de prioridade, ou qualquer forma de distinguir a prioridade de um projeto em relação ao outro);

- 7) Tarefa “Definir Procedimentos e Critérios Para o Balanceamento do Portfólio de Projetos”: esta atividade tem por objetivo definir a estratégia, procedimentos, ferramentas e técnicas a serem aplicadas no balanceamento do portfólio de projetos. É importante considerar que é necessário estabelecer critérios para avaliar quando é necessário rebalancear o portfólio e qual a capacidade de execução de projetos da organização. Deve ser estabelecido o quadro de responsáveis por analisar e propor as mudanças no portfólio;
- 8) Tarefa “Definir Procedimentos e Critérios Para o Acompanhamento do Portfólio”: esta tarefa tem o objetivo de estabelecer estratégias para revisar o portfólio periodicamente para verificar possíveis desvios, inclusive conflitos sobre recursos e riscos. É importante definir a periodicidade dessas revisões e critérios que devem ser levados em consideração e possíveis diretrizes que permitam julgar até que ponto os desvios do portfólio são aceitáveis ou quando medidas são necessárias (inclusive a possibilidade de rebalanceamento).

b) Fase “Identificação e Seleção de Projetos Candidatos”

Esta fase corresponde à etapa que se estende da captação de novas oportunidades de projeto à pré-seleção dos projetos com maior potencial estratégico para a organização. É composta pelas atividades de “Identificar Novas Propostas” e “Analisar Projetos Candidatos”.

b.1) Atividade “Identificar Novas Propostas”:

Esta atividade tem o objetivo de estabelecer uma lista de propostas de projetos para serem avaliadas posteriormente. São tarefas pertencentes a esta

atividade: “Identificar Projetos Candidatos” e “Categorizar Projetos”. A seguir estas atividades são descritas:

- 1) Tarefa “Identificar Projetos Candidatos”: o objetivo desta tarefa é fazer o registro e a descrição das informações necessárias de novas propostas que configuram possibilidades de projetos para a organização. Esta identificação pode ser consequência de propostas de negócio, solicitações de cliente, demandas internas, entre outras atividades. É importante registrar a maior quantidade de informações possíveis a respeito da proposta, e é relevante utilizar os parâmetros de avaliação do projeto como guia para a identificação das principais informações do projeto. Entre informações relevantes a serem registradas pode-se citar: identificador; cliente; escopo do projeto; prazo; tamanho; complexidade. Outro fator relevante é que ao iniciar um processo de gerência de portfólio de projetos em uma organização, esta atividade deve ocorrer de forma retroativa, isto é, além de identificar as novas propostas que surgem, é importante fazer a identificação dos projetos que já estão em desenvolvimento, possibilitando futuras ações de gerenciamento do portfólio;
- 2) Tarefa “Categorizar Projetos”: o objetivo desta tarefa é analisar as novas propostas de negócio e enquadrá-las nas categorias de projeto/investimento da organização, conforme os procedimentos e critérios mencionados na Tarefa “Definir Categorias de Projetos”. Uma vez categorizados, a organização pode aplicar procedimentos ou critérios de avaliação, seleção e priorização diferenciados, conforme a categoria de projeto, ou ainda estabelecer prioridades entre as categorias para fins de balanceamento de portfólio, entre outras aplicações práticas para a categorização de projetos.

b.2) Atividade “Analisar Projetos Candidatos”

Esta atividade tem o objetivo de conduzir a análise das propostas de projetos a fim de avaliar e filtrar as propostas mais relevantes para a organização, conforme procedimentos e critérios planejados. Como resultado, esta atividade fornece uma lista de projetos candidatos que estão aptos a serem priorizados para, quando necessário, compor o portfólio de projetos da organização. São tarefas desta atividade: “Avaliar Projetos Candidatos” e “Selecionar Projetos Candidatos”. Estas tarefas são descritas a seguir:

- 1) Tarefa “Avaliar Projetos Candidatos”: o objetivo desta tarefa é aplicar os procedimentos organizacionais previamente planejados para avaliar os projetos candidatos, fornecendo informações que propiciem a comparação de projetos através de uma linguagem comum, seja por meio de ferramentas, técnicas de avaliação, aplicação de critérios, *checklists*, entre outros;
- 2) Tarefa “Selecionar Projetos Candidatos”: o objetivo desta tarefa é aplicar os procedimentos e critérios definidos na Tarefa “Definir Procedimentos e Critérios Para Seleção de Projetos Candidatos” sobre os projetos candidatos avaliados, estabelecendo uma lista apenas com os projetos estrategicamente interessantes para a organização.

c) Fase “Seleção do Portfólio”

Esta fase compreende a etapa de composição do portfólio de projetos, utilizando as informações provenientes das fases “Identificação e Seleção de Projetos Candidatos” e “Monitoramento do Portfólio de Projetos”. Esta fase é composta pela atividade “Compor Portfólio de Projetos”.

c.1) Atividade “Compor Portfólio de Projetos”:

Esta atividade tem o objetivo de definir o portfólio de projetos da organização, isto é, o conjunto de projetos para serem executados em um determinado ciclo. Isto envolve priorizar os projetos selecionados, propor um conjunto ideal de projetos a serem executados (ou apenas serem mantidos), analisando os riscos destes projetos em conjunto (em relação aos objetivos estratégicos da organização), e autorizar o início destes projetos conforme diretrizes organizacionais.

Esta atividade é recorrentemente executada, sempre que houver necessidade de reestruturar o conjunto de projetos em execução, ou seja, sempre que for necessário balancear o portfólio de projetos. Isto pode ser consequência de projetos que foram concluídos e existe a possibilidade de iniciar novos projetos, de projetos que precisam ser redirecionados ou interrompidos por não mais serem sustentáveis ou viáveis para organização, e até mesmo em função de mudanças no ambiente de negócio que promovam mudanças nos objetivos estratégicos organizacionais, levando a necessidade de rever as prioridades do portfólio (BP23 – Identificar mudanças estratégicas que requeiram rebalanceamento do portfólio [Souza *et al.*, 2011]).

São tarefas desta atividade: “Priorizar Projetos”; “Balancear Portfólio”; “Identificar Resultados Esperados dos Projetos”; “Definir Estratégia de Alocação de Recursos”; “Identificar Riscos do Portfólio”; “Analisar Riscos do Portfólio”; “Comunicar Alterações no Portfólio”; “Aprovar Portfólio”; “Propor Recomendações”; e “Desenvolver Resposta aos Riscos do Portfólio”. Além disso esta atividade é composta pelas subatividades “Autorizar Projetos” e “Redirecionar Projetos”. Tais tarefas são descritas a seguir:

- 1) Tarefa “Priorizar Projetos”: esta tarefa tem o objetivo de determinar a prioridade dos projetos disponíveis, tanto em termos de repriorizar os projetos que já estão em execução, quanto em relação a projetos que foram avaliados e pré-selecionados. Os resultados desta

tarefa permitem análise dos projetos que são estrategicamente mais relevantes para a organização para fins de balanceamento do portfólio de projetos, alocação de recursos e resolução de conflitos;

- 2) Tarefa “Balancear Portfólio”: esta tarefa tem o objetivo de compor a carteira de projetos da organização, procurando estabelecer uma combinação viável de projetos que melhor favoreça o alcance dos objetivos estratégicos. Esta tarefa é uma das atividades chave da gestão de portfólio de projetos, e ocorre sempre que houver a necessidade de adicionar novos projetos ao portfólio ou rever a carteira atual, por recomendações ou mudanças na organização, como a atualização dos objetivos estratégicos por exemplo. O balanceamento do portfólio deve levar em consideração a capacidade de execução de projetos da organização (como a disponibilidade de recursos ou a existência de infraestrutura necessária) e os possíveis benefícios esperados da execução de projetos. Resultados da fase “Monitoramento do Portfólio” trazem recomendações sobre projetos que devem ser mantidos, descontinuados ou redirecionados. Estas recomendações devem ser consideradas na seleção/alteração do portfólio de projetos. O resultado desta tarefa é uma lista de projetos indicados para a execução, considerando a avaliação e a priorização dos projetos realizados nas tarefas anteriores, bem como a indicação de projetos que devem ser descontinuados, redirecionados ou encerrados;
- 3) Tarefa “Identificar Resultados Esperados dos Projetos”: esta tarefa tem o objetivo de determinar quais os benefícios e resultados esperados da execução dos projetos (ou atualizar os resultados esperados dos projetos já em execução, caso necessário), para possibilitar o acompanhamento do desempenho do projeto nos pontos de revisão do portfólio;
- 4) Tarefa “Definir Estratégia de Alocação de Recursos”: esta tarefa tem como objetivo definir uma estratégia viável de alocação de recursos

para os projetos que compõem a portfólio de projetos proposto na tarefa anterior. Isto envolve recursos financeiros, humanos e físicos (infraestrutura, equipamentos, alocações) e deve considerar não apenas as necessidades dos projetos, mas também a sua relevância para a organização, de forma que os projetos mais importantes tenham prioridade na alocação de recursos. Esta tarefa tem uma perspectiva organizacional, então devem ser observados os possíveis conflitos entre projetos que compartilham recursos e traçar uma estratégia que minimize e previna a ocorrência de tais conflitos;

- 5) Tarefa “Identificar Riscos do Portfólio”: esta tarefa tem como objetivo identificar os possíveis riscos da proposta de portfólio de projetos em formação. Os riscos de portfólio têm uma perspectiva mais abrangente do que os riscos de projeto, devendo ser investigado: riscos oriundos da combinação dos projetos e as dependências entre estes; riscos relacionados a conflitos na alocação de recursos; riscos estratégicos para a organização; e demais riscos que fogem do escopo dos projetos individuais. A identificação de riscos compreende a identificação de possíveis causas e de responsáveis;
- 6) Tarefa “Analisar Riscos do Portfólio”: esta tarefa tem o objetivo de analisar os riscos identificados do portfólio para quantificá-los em termos de probabilidade, impacto e demais informações relevantes para tomadas de decisão e acompanhamento do portfólio. A análise de riscos considera os riscos em conjunto e seus possíveis impactos no sucesso do portfólio. Os riscos devem ser priorizados para posterior acompanhamento e ações para os conter ou mitigar;
- 7) Tarefa “Comunicar Alterações no Portfólio”: esta tarefa tem o objetivo de comunicar aos tomadores de decisão da organização as propostas de mudanças na composição do portfólio, alocação de recursos, repriorização de projetos, mudança nos riscos e outras informações que apoiem a tomada de decisão ou que necessitem de aprovação de instâncias superiores;

- 8) Tarefa “Aprovar Portfólio”: esta tarefa tem o objetivo de oficializar as mudanças propostas sobre o portfólio, obtendo a autorização para que as propostas sejam executadas ou sinalizando que a proposta de portfólio não está adequada e necessita de adequações;
- 9) Tarefa “Propor Recomendações”: esta tarefa tem o objetivo de sugerir direcionamentos ou correções que devem ser atendidas para a aprovação de uma proposta de portfólio;
- 10) Tarefa “Desenvolver Resposta aos Riscos do Portfólio”: esta tarefa tem o objetivo de apontar mecanismos de contingência e mitigação aos riscos identificados do portfólio de projetos aprovado.

c.1.1) Atividade “Autorizar Projetos”

Uma vez aprovada uma mudança ou uma nova proposta de portfólio de projetos, esta atividade tem o objetivo de determinar o início oficial de novos projetos e realizar os procedimentos necessários para tal, como alocar recursos, estabelecer responsáveis e identificar metas e pontos de controle e marcos para os projetos.

São tarefas desta atividade: “Alocar Recursos para os Projetos”; “Definir Responsabilidades sobre Projetos”; “Estabelecer Marcos e Pontos de Revisão”; e “Autorizar Início da Execução de Projetos”. Estas tarefas são descritas a seguir:

- 1) Tarefa “Alocar Recursos Para os Projetos”: esta tarefa tem o objetivo de executar a estratégia de alocação de recursos proposta para o portfólio de projetos. Assim, recursos humanos, financeiros e físicos são aportados aos projetos por meios oficiais da organização, e o remanejamento dos recursos dos projetos já em execução é executado. A situação dos recursos organizacionais deve ser atualizada para fins de controle;
- 2) Tarefa “Definir Responsabilidades sobre Projetos”: esta tarefa tem o objetivo de estabelecer a responsabilidade, autoridade e autonomia

sobre projetos, designando papéis específicos (normalmente papéis de liderança como líder ou gerente de projeto) a recursos humanos, estabelecendo a autonomia destes sobre os projetos e mantendo o controle sobre quem deve ser consultado sobre informações do projeto;

- 3) Tarefa “Estabelecer Marcos e Pontos de Revisão”: esta tarefa tem o objetivo de determinar a periodicidade dos pontos de revisão dos projetos, para que o processo de gerência de portfólio tenha visibilidade sobre o seu desempenho;
- 4) Tarefa “Autorizar Início da Execução dos Projetos”: esta tarefa tem o objetivo de determinar o início oficial dos projetos, conforme os procedimentos oficiais da organização. Neste momento, os novos projetos que foram inseridos no portfólio de projeto são iniciados.

c.1.2) Atividade “Redirecionar ou Encerrar Projetos”

Esta atividade tem o objetivo de executar ações de encerramento ou redirecionamento de projetos. Esta tarefa acontece quando, por exemplo, um projeto de performance comprometida foi suspenso temporariamente ou cancelado, ou quando um projeto chegou a sua conclusão. Neste momento a organização deve identificar as lições aprendidas e resultados alcançados, e atualizar o estado do projeto.

Nenhum dos padrões que serviram de base para a definição deste framework aponta tarefas ou procedimentos para guiar a etapa de conclusão de projetos. Assim, as tarefas aqui definidas foram inspiradas na última fase do modelo IDEAL [Mcfeeley, 1996], a fase de Aprendizado (*Learning*), onde são identificados os objetivos alcançados e registradas as lições aprendidas.

- 1) Tarefa “Identificar Objetivos Alcançados”: esta tarefa tem o propósito de analisar quais os resultados alcançados a partir da execução do projeto. Em relação ao Modelo IDEAL esta

tarefa está relacionada ao passo de “Análise e Validação” da fase de “Aprendizado” (*Learning*), através do questionamento: “de que maneira o esforço alcançou os propósitos pretendidos?”. Este registro é importante para medir o sucesso do projeto e estabelecer um histórico para apoiar futuras tomadas de decisão a respeito de projetos similares;

- 2) Tarefa “Registrar Lições Aprendidas”: esta tarefa tem o propósito de identificar pontos positivos e negativos a respeito do andamento do projeto e sobre as tomadas de decisão realizadas, a fim de estabelecer uma base histórica que permita a melhoria do processo de gestão de portfólio de projetos em iterações futuras. Em relação ao Modelo IDEAL esta tarefa está relacionada ao passo de “Análise e Validação” da fase de “Aprendizado” (*Learning*), através dos questionamentos: “O que funcionou bem?” e “O que poderia ser feito de forma mais eficaz ou eficiente?”. As lições são coletadas, analisadas, sumarizadas e documentadas;
- 3) Tarefa “Atualizar Estado do Projeto”: esta tarefa tem o propósito de manter registro de qual ação foi tomada sobre este projeto, para fins de controle sobre a situação dos projetos e do portfólio como um todo. Deve-se identificar se um projeto foi cancelado, concluído ou redirecionado, bem como o motivo de tal ação.

d) Fase “Monitoramento do Portfólio”

Esta fase ocorre constantemente e garante o acompanhamento do portfólio e a produção de informações a respeito da situação do portfólio e seus componentes para tomadas de decisão sobre o balanceamento e a alocação de recursos organizacionais. Esta fase é composta pela atividade “Monitorar Portfólio”.

d.1) Atividade “Monitorar Portfólio”

Esta atividade tem o objetivo de acompanhar o portfólio de projetos, em relação a riscos e desempenho, em pontos de revisão, coletando informações para apoiar tomadas de decisão a respeito da continuidade de projetos e balanceamento de portfólio, bem como executar ações para mitigação de riscos.

É composta pelas tarefas: “Monitorar Desempenho do Portfólio”; “Monitorar Gatilhos de Riscos e Conflitos no Portfólio”; “Identificar Desvios do Portfólio”; “Realizar Ações para Tratar Desvios do Portfólio”; e “Relatar o Desempenho do Portfólio”. Estas tarefas são descritas a seguir:

- 1) Tarefa “Monitorar Desempenho do Portfólio”: esta tarefa tem o objetivo de coletar informações a respeito da situação dos projetos em atividade, e analisar o desempenho do portfólio em pontos de revisão. Os projetos são avaliados para averiguar se continuam justificáveis e viáveis, bem como identificar resultados alcançados e próximas metas, correlacionando-os aos objetivos estratégicos planejados e planejamentos definidos;
- 2) Tarefa “Monitorar Gatilhos de Risco e Conflitos no Portfólio”: esta tarefa tem o objetivo de identificar a ocorrência e a iminência de riscos que ameaçam o portfólio de projetos. Os riscos incluem ameaças aos objetivos estratégicos, ameaça ao sucesso de projetos, conflitos sobre recursos e quaisquer outros riscos identificados previamente e, caso necessário, comunica a identificação de novos riscos que podem ameaçar o portfólio e que possam influenciar no balanceamento do portfólio;
- 3) Tarefa “Identificar Desvios do Portfólio”: o objetivo desta tarefa é sumarizar os riscos e os problemas de desempenho do portfólio observados nas revisões de portfólio, e, assim, permitir a tomada de ações corretivas e a comunicação de problemas que possam comprometer o alcance dos objetivos estratégicos;

- 4) Tarefa “Realizar Ações para Tratar Desvios do Portfólio”: esta tarefa tem o objetivo de aplicar ações corretivas que possam mitigar os desvios identificados nas revisões de portfólio. Durante a atividade “Compor Portfólio”, ações de resposta a riscos foram previamente definidas, e caso se apliquem devem ser acionadas neste momento;
- 5) Tarefa “Relatar Desempenho do Portfólio”: o objetivo desta tarefa é comunicar o estado do portfólio de projetos, em relação ao seu desempenho, situação dos projetos, desvios e resultados de ações corretivas aplicadas. A partir destas informações, recomendações podem ser definidas para o balanceamento do portfólio, apontando projetos que estão comprometendo os resultados esperados do portfólio ou projetos que se mantêm justificáveis e viáveis.

25.4. Exemplo de Detalhamento da Tarefa “Definir Procedimentos e Critérios Para Avaliação de Projetos Candidatos”

Para fins de entendimento da estrutura do *framework* de processo, o detalhamento da tarefa “Definir Procedimentos e Critérios Para Avaliação de Projetos Candidatos” é apresentado a seguir. Uma vez que já fora apresentado o objetivo desta na subseção anterior, o detalhamento a seguir é em relação aos itens “Boas Práticas Relacionadas”, “Entradas e Resultados Sugeridos”, “Recomendações” e “Observações sobre o uso do item do *Framework*”.

25.4.1. Boas Práticas Relacionadas

A tarefa “Definir Procedimentos e Critérios Para Avaliação de Projetos Candidatos” foi definida a partir da Boa Prática “BP3 – Avaliar Projetos Candidatos”, conforme identificada em [Souza *et al.*, 2011].

25.4.2. Entradas e Resultados Sugeridos

Uma vez que se pretende avaliar projetos para a posterior comparação e seleção, sugestões de entradas para esta tarefa compreendem o conjunto de procedimentos e critérios de avaliação de projetos candidatos, conforme proposto na tarefa “Definir Procedimentos e Critérios Para Avaliação de Projetos Candidatos”, e a lista de projetos candidatos, previamente identificados, descritos e categorizados.

Ao término da tarefa, são esperados os resultados da avaliação dos projetos com os respectivos critérios aplicados, conforme definido em procedimentos. A lista de projetos avaliados, poderá ser posteriormente utilizada para seleção e priorização de projetos no portfólio organizacional.

25.4.3. Recomendações

Quanto às recomendações observadas nos padrões analisados, o modelo MR-MPS-SW sugere que as demandas deverão ser qualificadas, ou seja, deverão ser identificados os atributos que a caracterizam e que serão utilizados como critérios de seleção e priorização. Em geral estes critérios estão relacionados: ao retorno sobre o investimento (ROI); ao alinhamento com os planos estratégicos e táticos; ao balanceamento da carteira; ao uso mais efetivo de recursos; à probabilidade de sucesso (prazo, custo e escopo); à oportunidade de mercado; ao risco; entre outros. A escolha dos critérios a serem utilizados deve refletir os objetivos estratégicos da organização, de modo que a seleção e priorização possam contribuir para o sucesso da organização.

Em “*The Standard for Portfolio Management*”, são sugeridos os seguintes critérios para avaliação de projetos:

- Critérios de negócios: alinhamento estratégico; produtividade; melhoria de processos; vantagem competitiva; impacto nos negócios;

satisfação dos funcionários; satisfação do cliente; propriedade intelectual; impacto da não realização do projeto;

- Critérios financeiros: crescimento da receita; economia de custos; eliminação de custos; Taxa Interna de Retorno (TIR); Valor Presente Líquido (VPL); Retorno sobre o Investimento (ROI); período de retorno; custo; geração de fluxo de caixa;
- Critérios relacionados a riscos: riscos de negócios; riscos tecnológicos; riscos de gerenciamento de projetos; riscos de implementação; riscos de aceitação do mercado; riscos de relações públicas; riscos de imagem da marca;
- Critérios legais ou regulatórios;
- Critérios relacionados a Recursos Humanos (RH): competência específica; satisfação dos funcionários; disponibilidade de recursos; capacidade de RH; capacidade de RH para integrar a solução; impacto na condição de trabalho;
- Critérios de mercado: impacto no mercado; probabilidade de sucesso; tempo para mercado; impacto sobre as linhas de produtos existentes; vida útil estimada;
- Critérios técnicos: alinhamento de arquitetura; fornecimento de informações; probabilidade de sucesso (inverso de risco); confiabilidade, disponibilidade e sustentabilidade do sistema; conformidade com as normas.
- Quanto às ferramentas, o padrão sugere:
- Modelo de Pontuação com critérios chave ponderados: nesta abordagem é importante definir uma lista de critérios, agrupá-los em indicadores chave específicos, estabelecer o peso relativo dos critérios e definir o sistema ou escala de pontuação a ser utilizada (0 a 10, 1 a 5, etc);
- Representações Gráficas: histogramas, gráficos de pizza, gráficos de linha, gráficos de bolhas, grades de comparação de dois critérios (um muito utilizado é o gráfico Risco x Benefício);

- Análise de Especialista.

A partir da análise dos resultados do Mapeamento Sistemático da Literatura, as seguintes recomendações foram definidas como detalhamento para a tarefa em questão:

- [Xiaoguang *et al.*, 2008] define um *framework* de melhoria de processo, onde o “Processo de Balanceamento de Projetos” realiza a avaliação de projetos em potencial, através do “Sistema de Índice de Avaliação Multi Projeto”, onde são definidos os métodos de coleta de informações, métodos de medição, fatores de peso e critérios de decisão para avaliação de projetos;
- [Yi *et al.*, 2008] define um modelo para avaliação de projetos a partir de uma avaliação multicritérios com lógica Fuzzy. Para isto utiliza dois passos relevantes para este item do *Framework*: “Projetar estrutura de hierarquia de decisão” (onde são definidos os critérios para avaliação dos projetos) e “Avaliação fuzzy para pesos de critérios de decisão” (onde são definidos os pesos dos critérios de decisão);
- [Yi *et al.*, 2008] menciona critérios como: Benefícios econômicos; Competitividade; Conexão tecnológica; Chance de sucesso; e Viabilidade;
- [Balikuddembe e Bagula, 2009] propõe a definição dos limiares de valor de negócio para direcionar a avaliação dos projetos por meio do *Project Value Evaluator* (Avaliador de Valor do Projeto). Um diagrama de quadrantes que objetiva destacar o estado de negócio de qualquer projeto candidato em comparação com a estratégia de negócio estabelecida na organização, por meio da relação Benefício x Investimento;
- [Zhang e Fan, 2011] propõe uma abordagem de tomada de decisão em grupo baseada em múltiplos atributos linguísticos incertos.

Para isto é necessário definir previamente os critérios de avaliação dos projetos (ex: análise de risco, análise tecnológica, análise da equipe do projeto, análise de mercado), e as variáveis linguísticas para avaliação dos critérios (ex: muito fraco, levemente fraco, razoável, levemente bom, muito bom). Além disso, cada critério deve ter um peso associado;

- [Nascimento *et al.*, 2011] define, conforme a metodologia P³, a tarefa “Estabelecer um método de classificação de projetos (A.1.2)”, cujo objetivo é fornecer um meio comum para classificar os projetos, visto que estes nunca são criados de formas iguais;
- [Nascimento *et al.*, 2011] descreve e aplica um modelo de pontuação, onde são definidas perspectivas e seus respectivos indicadores aos quais são determinadas pontuações máximas e mínimas a serem concedidas durante atividade de avaliação dos projetos, bem como uma fórmula para normalização dos resultados para a obtenção de um score para os projetos;
- [Souza *et al.*, 2009] descreve a abordagem metodológica para o gerenciamento de portfólio, onde, na fase “Definir Gerência Estratégica de Portfólio”, é definido um conjunto de critérios de avaliação dos projetos, com seus respectivos pesos, chamados na estratégia de categorias de benefícios padrão e de benefícios padrão;
- [Souza *et al.*, 2009] trata-se de ferramental em desenvolvimento (na época do estudo) na Estação Taba.

25.4.4. Observações sobre o uso do item do *Framework*

Esta tarefa está diretamente relacionada à tarefa “Avaliar Projetos Candidatos” cujos resultados são utilizados vastamente ao longo das demais atividades sugeridas por este *Framework* de Processo, logo a importância deste item é

fundamental. O não uso deste item pode comprometer a aderência à Norma ISO/IEC 12207 e ao Resultado Esperado GPP1 do MR-MPS-SW.

25.5. Aplicabilidade do *Framework* de Processo

O *Framework* de Processo apresentado foi aplicado em uma organização de software paraense, conforme relato de experiência publicado em [Souza *et al.* 2012b], cuja pesquisa foi realizada pelo mesmo grupo. Como já dito, este *framework* não foi projetado para ser utilizado como um processo, mas sim estabelecer uma estrutura de referência para o planejamento e definição de processos de gerência de portfólio. Assim, como no relato de experiência descrito em [Souza *et al.*, 2012b] a aplicação do *framework* pode adotar os seguintes passos e questionamentos para nortear a definição de novos processos:

- Estabelecer metas: Identificar que tipo de processo deverá ser definido (quais as limitações e especificidades da organização que devem ser atendidas pelo processo a ser definido?, busca-se a aderência a algum padrão de qualidade específico?). No caso do relato definido em [Souza *et al.*, 2012b], houve uma reunião inicial, entre os consultores de melhoria e os representantes da organização alvo, para a definição destas metas, a partir de uma entrevista estruturada realizada pelos consultores;
- Estudo dos componentes do *Framework*: Os componentes do *framework* foram definidos a partir do entendimento das recomendações dos modelos, assim seus componentes retratam as sugestões de boas práticas destes. No relato descrito em [Souza *et al.*, 2012b], uma apresentação realizada pelos consultores de melhoria, expôs o *framework* de processo aos representantes da organização, utilizando-o como instrumento para explicação dos principais objetivos e atividades da gestão de portfólio de projetos;

- Identificação dos componentes relevantes para as metas definidas: A partir da análise dos componentes, devem ser identificados os componentes que representam e atendem as metas definidas no primeiro passo e as práticas complementares que são necessárias para o processo definido (isto inclui a identificação de práticas necessárias para a conformidade a algum padrão específico);
- Identificação das limitações do processo: Identificar práticas que não deverão compor o processo. No relato [Souza *et al.*, 2012b], as práticas relacionadas à gestão de risco foram suprimidas, para serem incorporadas apenas em futuras versões do processo;
- Adequação dos componentes selecionados: Alguns componentes podem ser decompostos ou unidos formando novas tarefas para o processo a ser definido. Os objetivos e sugestões de entradas e resultados permitem um entendimento do papel da prática no processo, facilitando esta etapa. No relato descrito em [Souza *et al.*, 2012b], os objetivos da tarefa “Identificar Resultados Esperados dos Projetos” foram incorporados ao contexto da tarefa “Identificar Projetos Candidatos”, para traduzir ao processo a forma de captação de novas propostas;
- Definição do fluxo de atividades/tarefas: O fluxo das tarefas definido no *framework* de processo configura uma sugestão, mas pode ser adaptado, sobretudo se houver adequações ou exclusão de componentes do *framework*;
- Definição de estratégias de implementação das tarefas: A partir das recomendações dos padrões e da literatura, descritas no detalhamento dos componentes do *framework* de processo, é possível identificar abordagens (ativos de processo, ferramentas, técnicas, papéis, procedimentos, entre outros) mais apropriadas para a implementação das tarefas selecionadas para o processo. No caso do processo definido no relato [Souza *et al.*, 2012b], por exemplo, foi adaptado o modelo de pontuação definido em [Souza *et al.*, 2009],

conforme listado nas recomendações da literatura para a tarefa “Definir Procedimentos e Critérios Para Avaliação de Projetos Candidatos”.

Assim, a aplicação realizada em [Souza *et al.*, 2012b] foi conduzida no ambiente de uma Fábrica de Software, a FabSoft, vinculada a uma instituição de ensino superior. Foi definido um processo de gestão de Portfólio, seguindo os passos definidos nesta seção. A definição do processo de gestão de portfólio na instituição tinha como meta a adequação ao respectivo processo do MPS.BR, além de atender as necessidades específicas da organização que incluíam a alta rotatividade de pessoas, a conciliação dos ciclos de desenvolvimento com as atividades acadêmicas da instituição de ensino e a categorização dos diversos tipos de projetos fornecidos pela organização.

A condução da atividade foi realizada em menos de dez horas de consultoria, divididas em reuniões semanais de uma hora conduzida com um consultor de melhoria de processo e dois representantes da Fabsoft: O gerente de negócios (e também coordenador) e um colaborador Gerente de Projetos. A utilização do *Framework* facilitou a comunicação entre as partes favorecendo o entendimento das práticas de gestão de portfólio de forma compatível com as recomendações do modelo MR-MPS. A estrutura do *Framework* permitiu visibilidade das atividades chaves com recomendações de como implementá-las, tendo observações sobre quais atividades eram obrigatórias no contexto de uma avaliação oficial MPS.BR, através das “Observações de Uso” dos seus componentes.

Esta visibilidade permitiu a maior participação dos representantes da organização na concepção do processo, tornando-os menos dependentes do consultor, que passou a atuar como facilitador do entendimento.

Em abordagens anteriores feitas para outros processos do MPS.BR na organização era feito um workshop com os representantes da organização para o entendimento dos Processos e em seguida a consultoria para implementar a melhoria. Nesta abordagem, os membros da organização atuaram de

forma prática no entendimento dos objetivos gerais da gestão de portfólio sem um treinamento específico nas recomendações do MPS.BR para o seu referido processo de Gerência de Portfólio de Projetos.

O processo resultante foi analisado por um segundo consultor, experiente em avaliações oficiais MPS.BR, para analisar a conformidade com os Resultados Esperados do modelo. A avaliação teve resultado positivo [Souza *et al.*, 2012b].

Um desafio encontrado foi a mudança da dinâmica da consultoria de melhoria de processo a partir da utilização desta ferramenta, necessitando um esforço inicial maior para o entendimento da estrutura do próprio *Framework*, por parte dos membros da FabSoft. Adicionalmente, durante a condução desta aplicação, o *Framework* sofreu atualizações decorrentes de necessidades observadas durante o experimento realizado. As atualizações ocorridas foram no sentido de permutação de tarefas entre atividades, para refletir melhor o entendimento das práticas.



Capítulo 26

Um *Framework* de Processo para Gerência de Recursos Humanos de Acordo com Padrões de Qualidade

em [Lira et al., 2012] e [Lira e Oliveira, 2013]

Wallace Michel Pinto Lira

Fledison Barbosa

Ehilton Yoshidome

Elder de Jesus Ferreira da Silva

Paulo Robson Campelo Malcher

Sandro Ronaldo Bezerra Oliveira

A Gestão de Recursos Humanos (GRH) pode ser descrita como uma abordagem coerente, estratégica e integrada no que tange o emprego, desenvolvimento e bem estar das pessoas nas organizações [Armstrong, 2010]. Este conceito é corroborado no Guia do PMBOK, o qual estabelece quatro processos relacionados à GRH: Desenvolver o Plano de Recursos Humanos; Mobilizar a Equipe de Projeto; Desenvolver a Equipe de Projeto; e Gerenciar a Equipe de Projeto.

Gerir os recursos humanos é importante para organizações desenvolvedoras de software, haja vista que modelos e normas de qualidade do processo de software apontam esta necessidade. Entre estes modelos e normas, destacam-se o MR-MPS-SW; CMMI-DEV; e ISO/IEC 12207.

A indústria, portanto, apresenta vários guias de boas práticas para Gerência de Recursos Humanos. Com o objetivo de investigar se estas

referências possuem sinergias e pontos de conflito, foi realizado um mapeamento entre as práticas propostas por estas referências. Posteriormente, definiu-se um framework de processo aderente a este mapeamento. Portanto, é objetivo deste trabalho descrever um Framework de processo aderente aos principais padrões da área já estabelecidos na indústria.

Framework de processo, neste contexto, é um “alicerce para um processo de software completo através de um pequeno número de atividades as quais são aplicáveis a todos os projetos” [Pressman, 2011]. É necessário definir um framework para integrar as boas práticas das referências em um único conjunto de práticas.

26.1. Fundamentação Teórica

Como base para a elaboração do framework de processo de apoio à Gerência de Recursos Humanos, foi necessário desenvolver um conjunto de referências apropriadas.

Os modelos MR-MPS-SW, CMMI-DEV e a norma de qualidade ISO/IEC 12207 formam a base do referencial teórico. Estes guias apontam boas práticas consolidadas na indústria, definindo padrões de qualidade para processos de software baseados na experiência de organizações de sucesso comprovado no setor. O CMMI-DEV e a ISO/IEC 12207 estão consolidadas internacionalmente, ao passo que o MR-MPS está consolidado no Brasil, com perspectivas de internacionalização [SOFTEX, 2012b].

Posteriormente, as boas práticas propostas pelas referências foram analisadas em relação aos artefatos e técnicas propostas no PMBOK. O PMBOK foi incluído no estudo por ser um guia de boas práticas para Gerência de Projetos em geral, abrangendo, assim, o desenvolvimento de software.

As subseções a seguir apresentam mais detalhes sobre os guias de boas práticas que compõem a fundamentação teórica do framework proposto neste artigo.

26.1.1. ISO/IEC 12207

A norma ISO/IEC 12207 foi desenvolvida pela *International Organization for Standardization* em conjunto com a *International Electrotechnical Commission*. Esta norma fornece um conjunto abrangente de processos, atividades e tarefas do ciclo de vida de software. Atualmente, a norma é utilizada como uma referência de termos técnicos e técnicas para processos de software.

Apresenta, em seu Processo de Gestão de Recursos Humanos, orientações acerca de: treinamento de recursos humanos; avaliação de recursos humanos; e gerência de conhecimento. Abrange, também, orientações para o gerenciamento da alocação dos recursos humanos nos setores e projetos da organização.

26.1.2. CMMI-DEV

O CMM, ou *Capability Maturity Model*, é um modelo de maturidade de processo baseado na capacidade da organização de definir, gerenciar e melhorar seus processos. A experiência do SEI (*Software Engineering Institute*) com os modelos CMM possibilitou a evolução para o modelo CMMI, que apresenta os modelos: CMMI for Services (CMMI-SVC), CMMI for Development (CMMI-DEV) e CMMI for Acquisition (CMMI-ACQ).

Dentre estes modelos, o trabalho irá focar no CMMI-DEV por seu foco em processos de desenvolvimento. No que se refere à Gerência de Recursos Humanos, o CMMI-DEV apresenta orientações para a área de processo de Treinamento Organizacional. Não existem, no entanto, orientações para avaliação de recursos humanos ou para práticas de Gerência de Conhecimento.

26.1.3. MR-MPS-SW

É um modelo de referência para melhoria gradual do processo de software adaptado à realidade das empresas brasileiras. O MR-MPS-SW é fundamentado, principalmente, no CMMI-DEV e nas normas ISO/IEC 15504 e ISO/IEC 12207.

O MR-MPS-SW apresenta um processo específico para GRH, semelhante à ISO/IEC 12207. O propósito deste processo é prover à organização e aos projetos os recursos humanos necessários e manter suas competências adequadas às necessidades do negócio. É composto por 11 (onze) Resultados Esperados que possuem como objetivo orientar as organizações acerca deste processo.

26.1.4. PMBOK

O Guia do Conhecimento em Gerenciamento de Projetos, ou PMBOK, é “uma norma reconhecida para a profissão de gerenciamento de projetos”, a qual descreve normas, métodos, técnicas e práticas estabelecidas.

Apresenta em seu escopo uma área de conhecimento denominada Gerenciamento de Recursos Humanos, a qual inclui processos que organizam e gerenciam a equipe do projeto: Desenvolver o Plano de Recursos Humanos, Mobilizar a Equipe do Projeto, Desenvolver a Equipe do Projeto e Gerenciar a Equipe do Projeto.

26.2. Mapeamento

Foi realizado um mapeamento entre boas práticas sugeridas pelos guias para GRH apresentados na seção anterior com o intuito de condensar estas boas práticas em um único conjunto de referência [Lira e Oliveira, 2011]. Para tal,

foi necessário estabelecer uma relação de equivalência entre os elementos dos modelos. Sendo assim, foram mapeadas: as Tarefas da ISO/IEC 12207; as Práticas Específicas do CMMI-DEV; os Resultados Esperados do MR-MPS; e os Artefatos e Técnicas do PMBOK.

A comparação entre os elementos citados no parágrafo anterior, referentes aos padrões para Gerência de Recursos Humanos, resultou no mapeamento. No mapeamento foram identificadas 23 (vinte e três) práticas. A Tabela 25 apresenta as boas práticas identificadas e os modelos em que se identificou a presença destas práticas.

Analisando o mapeamento, foi possível notar que todas as práticas elencadas são recomendadas por pelo menos a metade dos guias selecionados.

A divulgação deste mapeamento na íntegra, incluindo a especificação dos elementos mapeados e a justificativa de cada item do mapeamento, está disponível em http://www.spider.ufpa.br/projetos/spider_grh/mapeamento_grh.pdf.

Tabela 25. Mapeamento de Boas Práticas para Gerência de Recursos Humanos [Lira e Oliveira, 2011]

ID	Boas Práticas	ISO/IEC 12207	CMMI-DEV	MR-MPS-SW	PMBOK
1	Identificar/Rever Necessidades Organizacionais	S	S	S	S
2	Planejar Recrutamento	S	N	S	S
3	Definir Critérios de Avaliação de Recrutamento	S	N	S	S
4	Recrutar Recursos Humanos	S	N	S	S
5	Desenvolver Sistema de Recompensas	S	N	N	S
6	Determinar Necessidades de Treinamento	S	S	S	S
7	Planejar Treinamentos Necessários	S	S	S	S
8	Planejar Estratégia de Treinamento	S	S	S	S
9	Executar Treinamento de Recursos Humanos	S	S	S	S
10	Avaliar Efetividade do Treinamento	S	S	S	S
11	Definir Critérios de Avaliação de Efetividade dos Recursos Humanos	S	N	S	S
12	Avaliar Efetividade dos Recursos Humanos	S	N	S	S
13	Difundir Resultado das Avaliações de Efetividade	S	N	S	S

ID	Boas Práticas	ISO/IEC 12207	CMMI-DEV	MR-MPS-SW	PMBOK
14	Registrar Avaliações de Efetividade de Recursos Humanos	S	N	S	S
15	Definir Conceito de Ativos de Conhecimento	S	N	S	N
16	Realizar Treinamento em Ativos de Conhecimento	S	N	S	N
17	Estabelecer Rede de Especialistas	S	N	S	N
18	Estabelecer Mecanismo de Apoio à Troca de Informações	S	N	S	N
19	Apoiar Aquisição de Conhecimento	S	N	S	N
20	Definir Estrutura da Equipe	S	S	S	S
21	Revisar Planos Pertinentes com a Equipe	S	S	S	S
22	Gerenciar Conflitos da Equipe	S	N	S	S
23	Avaliar Disponibilidade de Recursos Humanos	S	N	S	S

26.3. Framework

No mapeamento foram propostas 23 (vinte e três) práticas para Gerência de Recursos Humanos, conforme apresentado anteriormente. No entanto, para definir o *framework*, foi necessário, também, agrupar estas práticas por afinidade e estabelecer um fluxo de execução entre elas. As subseções a seguir descrevem em maiores detalhes o *framework* elaborado.

26.3.1. Macro-fluxo do Framework

As atividades elicitadas através do mapeamento foram agrupadas em macro-atividades, as quais compreendem: Planejar Gerência de Recursos Humanos; Desenvolver Recursos Humanos; Acompanhar Recursos Humanos; Gerenciar Conhecimento. Estas atividades foram ordenadas em um macro-fluxo, conforme apresenta a Figura 277.

A primeira macro-atividade a ser executada é “Planejar a Gerência dos Recursos Humanos”. As outras atividades do framework devem seguir as diretrizes propostas no planejamento.

Vale ressaltar que o macro-fluxo de Gerência de Recursos Humanos é executado quando ocorre o evento “Nova Necessidade Organizacional”. Esta nova necessidade pode ser identificada em projetos e também em setores de apoio da organização. Uma necessidade organizacional pode incluir: carência de conhecimento em uma determinada tecnologia de desenvolvimento; falta de recursos humanos com perfil para liderar equipes; aumentar a motivação da equipe através de recompensas; entre outras.

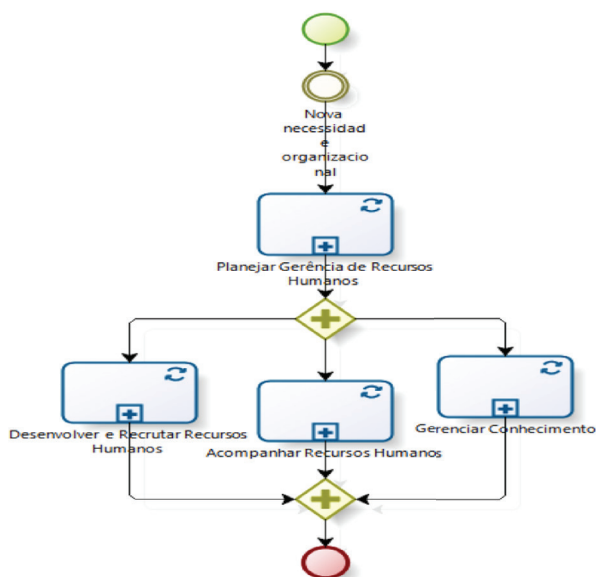


Figura 277. Macro-fluxo de Atividades do *Framework*

26.3.2. Planejar Recursos Humanos

A macro-atividade Planejar Recursos Humanos foi decomposta em 10 (dez) atividades, como evidencia a Figura 278.

A atividade de Identificar/Rever Necessidades Organizacionais é chave na execução do framework de processo. Durante seu escopo, é necessário definir se a organização necessita de treinamentos, recrutamentos ou identificar, ainda, a necessidade de melhorar a remuneração e/ou cargos dos recursos humanos. Portanto, nesta etapa é importante que o alto nível hierárquico da organização esteja em sintonia com as necessidades dos recursos humanos.

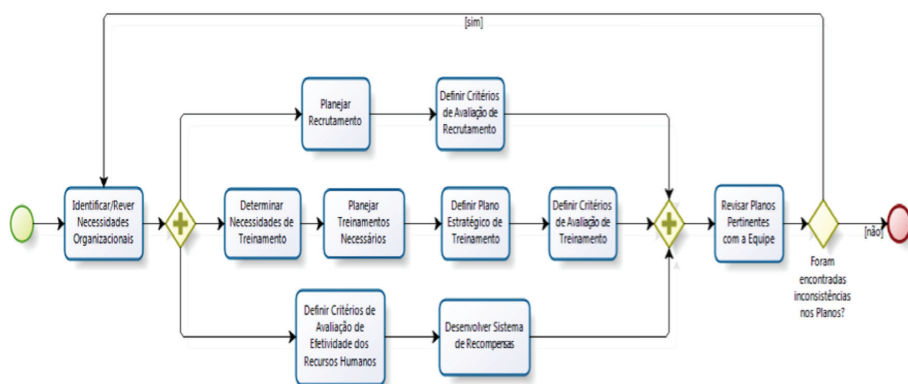


Figura 278. Detalhamento de Planejar Recursos Humanos

O fluxo de execução é posteriormente dividido em 3 (três) fluxos paralelos. A bifurcação superior aponta a necessidade de Planejar o Recrutamento e, em seguida, Definir Critérios de Avaliação de Recrutamento. Planejar Recrutamento é necessário, pois as necessidades organizacionais devem ser analisadas e uma estratégia de captação de Recursos Humanos deve ser elaborada. A atividade posterior é importante, pois a organização precisa manter o controle sobre a qualidade de seu processo de recrutamento para, a partir destas informações, identificar pontos fracos e oportunidades de melhoria.

A bifurcação intermediária trata do planejamento dos treinamentos a serem realizados na organização. Durante a execução da atividade Determinar Necessidades de Treinamento, é investigado se o treinamento a ser realizado é de responsabilidade da organização ou de projetos específicos.

Este passo é importante, pois o custo para realizar treinamentos a nível de projeto deve usar o orçamento do mesmo, o que pode inviabilizar projetos. Em seguida, durante a atividade Planejar Treinamentos Necessários, as necessidades de treinamento devem ser revisadas, e a partir disso a estratégia de treinamento deve ser definida. Esta estratégia de treinamento deve analisar principalmente quais serão os tipos de treinamento que melhor enquadram-se à necessidade de treinamento em questão. Posteriormente, deve ser executada a atividade Definir Plano Estratégico de Treinamento, na qual informações pertinentes a respeito do planejamento do treinamento são elicitadas. Estas informações incluem, mas não se limitam a: cronogramas de implementação, requisitos de recurso e necessidade dos treinamentos. Finalmente, há a necessidade de desenvolver a atividade Definir Critérios de Avaliação de Treinamento.

A bifurcação inferior trata da necessidade de promover um sistema de recompensas dentro da organização. A primeira atividade, Definir Critérios de Avaliação de Efetividade de Recursos Humanos, deve ser realizada primeiro para, com base nestes critérios de análise, Desenvolver um Sistema de Recompensas. Estas recompensas podem envolver bônus salarial ou promoção de cargo na organização.

Finalmente, é apontada a necessidade de dar visibilidade à equipe a respeito de todo o planejamento de recursos humanos realizado. É importante que, caso inconsistências sejam encontradas nesta etapa, elas sejam sanadas. Apesar do fluxo retornar à primeira atividade do Planejamento de Recursos Humanos, apenas as atividades que apresentaram inconsistências e as subsequentes precisam ser revistas.

26.3.3. Desenvolver Recursos Humanos

A macro-atividade Desenvolver Recursos Humanos foi decomposta em 5 (cinco) atividades. Segue na Figura 279 o detalhamento deste fluxo.

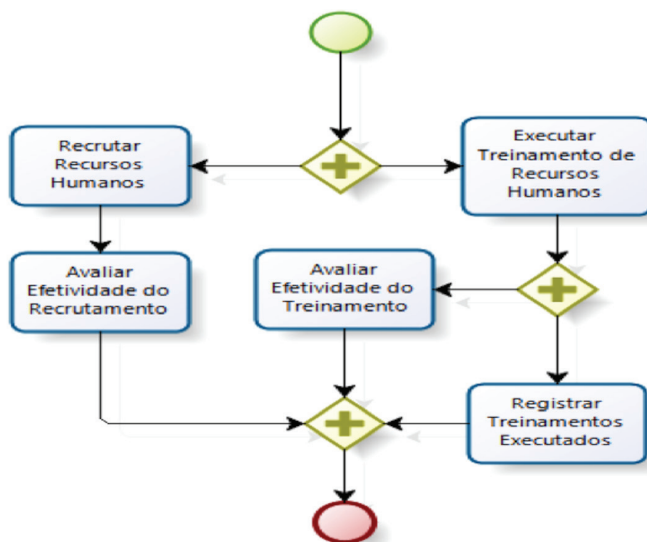


Figura 279. Detalhamento de Desenvolver Recursos Humanos

Desenvolver Recursos Humanos engloba recrutar e treinar os recursos humanos da organização, conforme a necessidade desta organização. Ambas as atividades ocorrem em paralelo no framework.

Treinar os recursos humanos engloba as atividades Executar Treinamento de Recursos Humanos, Registrar Treinamentos Executados e Avaliar Efetividade do Treinamento. Executar Treinamento de Recursos Humanos tem como objetivo aplicar a estratégia de treinamento conforme planejado e, em seguida, a atividade Registrar Treinamentos Executados garante o armazenamento das informações de treinamentos pertinentes. A atividade Avaliar Efetividade do Treinamento deve avaliar o desempenho do treinamento por meio de atividades e métodos previamente planejados. Avaliar os treinamentos é importante para identificar oportunidades de melhoria no treinamento dos recursos humanos.

Recrutar recursos humanos engloba, além da atividade homônima, a atividade Avaliar Efetividade do Recrutamento. A atividade Recrutar Recursos Humanos deve aplicar a estratégia de recrutamento, conforme planejado, e manter registros desse recrutamento para posterior avaliação

do mesmo. A atividade Avaliar Efetividade do Recrutamento não foi identificada como uma boa prática no mapeamento proposto. No entanto, foi inserida para aumentar o autoconhecimento da organização, provendo meios para identificar, no recrutamento, eventuais pontos fortes, fracos e oportunidades de melhoria de forma semelhante ao que ocorre no treinamento de recursos humanos.

26.3.4. Acompanhar Recursos Humanos

A macro-atividade Acompanhar Recursos Humanos foi decomposta em 6 (seis) atividades, conforme apresenta a Figura 280.

Alocar recursos humanos tem duas finalidades principais: alocar e gerir as equipes da organização.

Para alocar recursos humanos efetivamente, são necessárias as atividades Avaliar Disponibilidade de Recursos Humanos e Definir Estrutura da Equipe. A atividade Avaliar Disponibilidade de Recursos Humanos tem como meta verificar disponibilidade de recursos humanos para a execução do projeto e funcionamento das equipes de apoio, definindo papéis e responsabilidades. A atividade Definir a Estrutura da Equipe deve garantir que estes recursos humanos possuem o perfil de habilidades e competências necessárias para atender as demandas do papel o qual está relacionado. Adicionalmente, podem ser levadas em consideração as relações interpessoais entre os recursos humanos para diminuir os conflitos na equipe e aumentar sua produtividade. Finalmente, é necessário garantir que não haja conflito de horários na alocação dos recursos humanos.

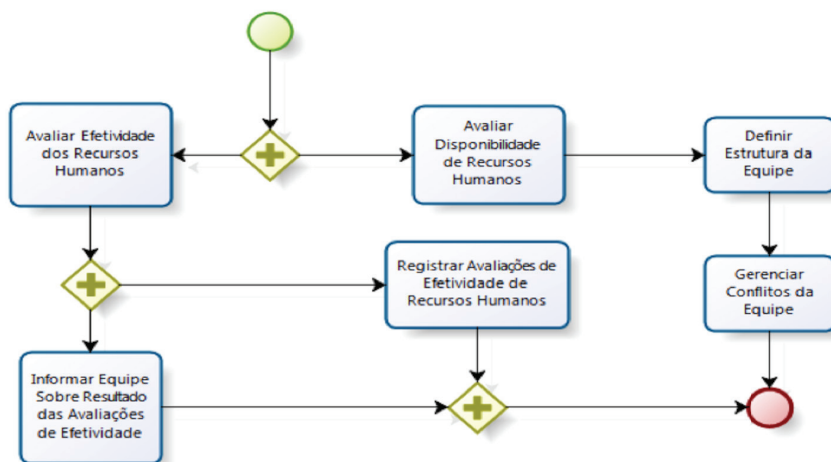


Figura 280. Detalhamento de Acompanhar Recursos Humanos

A Gerência de Conflitos da Equipe possui como objetivo manter as relações interpessoais compatíveis com as necessidades do ambiente de trabalho. Para tal, é necessário desenvolver ações preventivas e solucionar conflitos nas equipes conforme estes aconteçam.

Paralelamente, a atividade Avaliar Efetividade dos Recursos Humanos deve ser realizada para fornecer à organização subsídios para desenvolver um processo de Gerência de Recursos Humanos efetivo. A avaliação de efetividade deve utilizar os critérios e as diretrizes planejadas na atividade Definir Critérios de Avaliação de Recursos Humanos.

26.3.5. Gerenciar Conhecimento

A macro-atividade Gerenciar Conhecimento foi decomposta em 5 (cinco) atividades, conforme mostra a Figura 281.

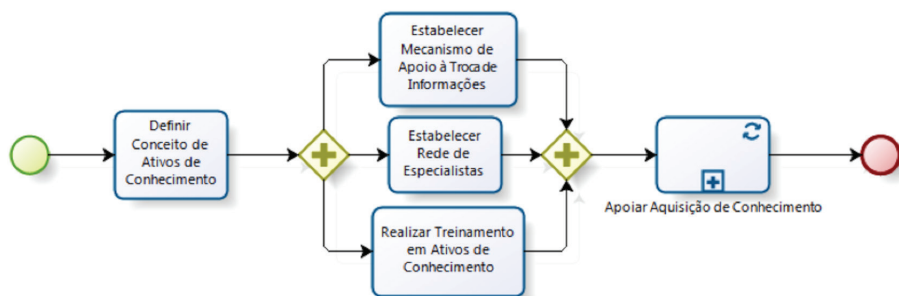


Figura 281. Detalhamento de Acompanhar Recursos Humanos

A primeira atividade, Gerenciar Conhecimento, inclui a definição de ativos de conhecimento. Esta atividade é necessária porque conceito de ativo de conhecimento pode variar conforme a organização [SOFTEX, 2012b].

A atividade Estabelecer Mecanismo de Apoio à Troca de Informações é uma necessidade, haja vista que a Gestão de conhecimento depende fortemente da troca de conhecimento entre os recursos humanos. Mecanismos de apoio à troca de informações podem incluir *wikis*, ferramentas de *chat*, reuniões periódicas para troca de informações, entre outras.

É necessário, também, Estabelecer uma Rede de Especialistas, os quais são indivíduos que atuarão como referência em uma ou mais áreas estrategicamente importantes para a organização. Estes especialistas podem ser recursos humanos da organização ou consultores externos. Os especialistas podem ser consultados utilizando os mecanismos de apoio à troca de informações estabelecidos.

No contexto desta pesquisa, a atividade Apoiar Aquisição de Conhecimento tem como objetivo estabelecer um sistema de gestão de ativos de conhecimento na organização. A aquisição do conhecimento foi estabelecida no framework com uma abordagem derivada de um “modelo que apoia, também, a filtragem do conhecimento relevante para a organização” [Oliveira, 2007].

26.4. Considerações sobre o *Framework*

O maior diferencial do *framework* de processo de apoio à Gerência de Recursos Humanos elaborado é evidenciar o seu apoio completo às recomendações dos guias de boas práticas para GRH. Este trabalho contribui, também, para evidenciar a inexistência de inconsistências entre as propostas destes guias de boas práticas. Dada a sua natureza, o *framework* pode ser adaptado para diferentes situações. Sua instanciação é flexível para suportar diferentes técnicas para desenvolver uma determinada atividade.

Adicionalmente, caso apenas a implantação de apenas um modelo de qualidade seja o objetivo a ser atingido por uma organização, o *framework* possui flexibilidade necessária para tal. Por exemplo, na implantação apenas do CMMI-DEV, deve ser consultado o mapeamento para investigar quais são as atividades requeridas para tal. Neste caso, são elas: Identificar/Rever Necessidades Organizacionais; Determinar Necessidades de Treinamento; Planejar Treinamentos Necessários; Planejar Estratégia de Treinamento; Executar Treinamento de Recursos Humanos; Avaliar Efetividade do Treinamento; Definir Estrutura da Equipe; e Revisar Planos Pertinentes com a Equipe. Note que, neste exemplo, apesar de uma parte significativa do *framework* ter sido excluída, as dependências das atividades restantes foram satisfeitas.

Este trabalho foi avaliado por um especialista em qualidade de software com experiência na implantação de práticas do MR-MPS-SW, do CMMI-DEV e da norma ISO/IEC 12207. Uma das dificuldades encontradas na avaliação foi a necessidade de consultar o mapeamento para determinar a relação entre as atividades do *framework* e os modelos de qualidade. Por este motivo, um comentário foi adicionado a cada atividade do *framework* com esta informação, a qual pode ser visualizada na versão integral do mapeamento. O avaliador constatou, ainda, que o mapeamento e *framework* em questão podem ser estendidos para comportar outros modelos e normas de qualidade, tais como: *Personal Software Process* (PSP), *Team Software Process* (TSP)

e *People Capability Maturity Model* (People-CMM). Finalmente, foi relatado que as atividades propostas podem atender às boas práticas especificadas no mapeamento, validando a proposta deste trabalho.

Durante o desenvolvimento desta pesquisa, foram encontrados trabalhos relacionados no que tange a definição de processos para GRH. Uma destas [Shnaider, 2003] pesquisas não leva em consideração no seu desenvolvimento o mesmo conjunto de guias de boas práticas para Gerência de Recursos Humanos, apenas o MR-MPS. Outra [Morais, 2009] descreve um processo de software que teve como fundamentação teórica as normas ISO/IEC 12207, ISO 10015, ISO 9001, ISO 9004, ISO 10006 e os modelos CMMI-DEV e MR-MPS. No entanto, seu trabalho não evidencia a aderência do processo proposto a estes modelos e normas, além de não mapeá-los para evidenciar as relações entre os escopos dos modelos e normas.



REFERÊNCIAS BIBLIOGRÁFICAS

- Abadi, A., Nisenson, M., Simionovici, Y., 2008, A Traceability Technique for Specifications, Proc. 16th IEEE Int. Conf. Program Comprehension, pp. 103-112.
- ABNT - Associação Brasileira de Normas Técnicas, 1994, NBR ISO 8402/1994 – Gestão da Qualidade e Garantia da Qualidade – Terminologia, Brasil.
- Ackerman, A., Buchwald, L., Lewski, F., 1989, Software Inspections: An Effective Verification Process, IEEE Software, vol. 6, no. 3, p.31-37.
- ACM/AIS/IEEE-CS, 2005, The Joint Task Force for Computing Curricula, Computing Curricula 2005, The Overview Report.
- Acuña, S. T., Juristo, N., 2004, Assigning people to roles in software projects, Software: Practice and Experience, 34(7), John Wiley & Sons, Inc, New York, pp 675–696.
- Albrecht, A. J., Gaffney, J., 1983, Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation, IEEE Transaction on Software Engineering, SE-9, 6.
- Alho, F. M., Macedo, K. O., Oliveira, S. R. B., 2011, Spider-Reuse: Um Ferramental para Apoio Sistêmico ao Processo de Gerência de Reuso de Ativos de Software aderente ao MR-MPS, ISO/IEC 12207 e IEEE 1517, Anais do WSL 2011, Porto Alegre-RS.
- Almeida *et al.*, 2011, Apoio aos Processos de Gerência de Requisitos e Verificação e Validação em um Ambiente Integrado, VII Workshop Anual do MPS (WAMPS 2011), Campinas-SP.
- Ambrósio *et al.*, 2008, Experiência em Projetos e Uso de Técnicas de Verificação e Validação de Software em Aplicações Espaciais no INPE, INPE-15182-NTC/374,

Ministério da Ciência e Tecnologia – Instituto Nacional de Pesquisas Espaciais, São José dos Campos-SP.

- Anda, B., Dreiem, H., Sjöberg, D. I. K., Jørgensen, M., 2001, Estimating Software Development Effort based on Use Cases - Experiences from Industry, Fourth International Conference on the UML.
- Anderson, L. W., Krathwohl, D. R., 2001, A Taxonomy for Learning, Teaching, and Assessing: a Revision of Bloom's Taxonomy of Educational Objectives, Longman, New York.
- Anohina, A., Graudina, V., Grundspenkis, J., 2007, Using Concept Maps in Adaptive Knowledge Assessment, Advances in Information Systems Development, New Methods and Practice for the Networked Society, v. 1, Springer, pp 469-479.
- Anohina, A., Strautmane, M., Grundspenkis, J., 2010, Development of the scoring mechanism for the concept map based intelligent knowledge assessment system, Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies, Nova York.
- Antoniol, G. *et al.*, 2002, Recovering traceability links between code and documentation, IEEE Transactions on Software Engineering, 28(10):970–983.
- Araújo, L. L., Mello, A. B., 2009, Implantação do MPS.BR (Melhoria do Processo de Software Brasileiro), Nível F, com TFS (Team Foundation Server) no Desenvolvimento Eficiente de Sistemas, V Workshop Anual do MPS (WAMPS 2009), Campinas-SP.
- Armenise, P., Bandinelli, S., Ghezzi, C., Morzenti, A., 1993, A survey and assessment of software process representation formalism, International Journal of Software Engineering and Knowledge Engineering 3(3), pp 401-426.
- Armstrong, M., 2010, Armstrong's Essential Human Resource Management Practice, 1st edition, editora Krogan Page.
- Bagheri, S., Ratakonda, K., Mohan, R., 2010, Variable Productivity Adjustment Estimation for Function Point Project Delivery, IEEE International Conference on Data Mining Workshops (ICDMW 2010).
- Baldez, G. P., Barbosa, S. O., Oliveira, S. R. B., 2010, Spider-APF e Spider-UCP: Apoios Sistêmicos à Estimativa de Projetos no Contexto da Qualidade de Software, Anais do ERIN 2010, Manaus-AM.

- Balikuddembe, J. K., Bagula, A., 2009, Business viability assessment of potential software projects: An empirical study with the CASSE framework, Communications in Computer and Information Science.
- Barbalho, R. A., Teles, M. P., Oliveira, S. R. B., 2011, Uma Implementação do Processo de Garantia da Qualidade usando a Spider-QA, a Spider-CL e o Mantis, Anais do ENACOMP 2011, Catalão – GO.
- Barcellos, M. P., 2009, Uma Estratégia para Medição de Software e Avaliação de Bases de Medidas para Controle Estatístico de Processos de Software em Organizações de Alta Maturidade, Tese de Doutorado. UFRJ, Rio de Janeiro-RJ.
- Barreto, A. O. S., 2006, Apoio à Verificação de Software em Ambientes de Desenvolvimento de Software, Dissertação de Mestrado - COPPE/UFRJ, Rio de Janeiro-RJ.
- Barros, R. S., Oliveira, S. R. B., 2010a, Spider-CL: Uma Ferramenta de Apoio ao Uso de Critérios Objetivos no Contexto da Qualidade de Software, II Escola Regional de Informática – ERIN 2010, Manaus-AM.
- Barros, R. S., Oliveira, S. R. B., 2010b, SPIDER_ML: Uma Linguagem de Modelagem de Processos de Software, II Escola Regional de Informática – ERIN 2010, Manaus-AM.
- Barros, R. S., Oliveira, S. R. B., 2010c, Spider-PM: Uma Ferramenta de Apoio à Modelagem de Processos de Software, Anais do ENACOMP 2010, Catalão-GO.
- Barros, R. S., 2010, Spider-PM: Um Ferramental de Apoio para Modelagem de Processos de Software, Trabalho de Conclusão de Curso, Curso de Bacharelado em Ciência da Computação, UFPA, Belém-PA.
- Bartié, A., 2002, Garantia da qualidade de software, Campus, Rio de Janeiro-RJ.
- Basili, V. R., 1985, Quantitative Evaluation of Software Methodology, Keynote Address, First Pan Pacific Computer Conference, Melbourne, Australia.
- Basili, V. R., Kan, S. H., Shapiro, L. N., 1994, Software Quality: An Overview from the Total Quality Management Perspective, IBM Systems Journal.
- Bassi, G., 2013, Gestão de risco ágil: alguma ideias, Disponível em <http://blog.lambda3.com.br/2011/01/gesto-de-risco-gil-alguma-ideias/>, Último Acesso em Dezembro de 2015.
- Beck, K., 2003, Test Driven Development: By Example, Pearson Education.
- Beck, K., 2004, Programação extrema explicada: acolha as mudanças, Bookman.

- Beck, K., Andres, C., 2005, *Extreme Programming explained: embrace change*, Second Edition, Upper Saddle River: Addison-Wesley.
- Becker-Kornstaedt, U., Belau, W., 2000, *Descriptive process modeling in an industrial environment: Experience and guidelines*, Software Process Technology. Disponível em: <http://www.springerlink.com/index/t43q70180l337w16.pdf>, Último Acesso em Novembro de 2015.
- Bendraou, R. *et al.*, 2010, *A Comparison of Six UML-Based Languages for Software Process Modeling*.
- Bertollo, G., 2006, *Definição de Processos Em Um Ambiente de Desenvolvimento De Software*, Universidade Federal do Espírito Santo – Departamento De Informática, Mestrado Em Informática, Vitória-ES.
- Boehm, B. W., 1981, *Software Engineering Economics*, Prentice Hall.
- Boria, J., Rubinstein, V., Rubinstein A., 2013, *A história da Tahini-Tahini - Melhoria de processos de software com métodos ágeis e modelo MPS*, Brasil.
- Boyle, E. A., Connolly, T. M., Hainey, T., 2011, *The role of psychology in understanding the impact of computer games*, Entertainment Computing, 2, pp 69–74.
- Brito Neto, O. N., Oliveira, S. R. B., 2013, *Utilizando Ferramentas de Software Livre para Implementação de Testes a partir do Processo de Verificação constante no Modelo de Referência do MPS.BR*, Anais do Computer on the Beach 2013, Florianópolis-SC.
- CaliberRM, 2015, *Gerenciamento de Requisito para Engenharia de Software*, Disponível em: <http://www.borland.com/br/products/caliber/index.htm>, Último Acesso em Novembro de 2015.
- Cardias Junior, A. B., Bentes, L. N., Souza, M. R. A., Yoshidome, E. Y. C., Oliveira, S. R. B., 2010, *Uma Análise Avaliativa de Ferramentas de Software Livre no Contexto da Implementação do Processo de Gerência de Requisitos do MPS. BR*, Anais do WER10 - Workshop em Engenharia de Requisitos (pp 75-84), Cuenca-Ecuador.
- Carneiro, S. N. C., Oliveira, S. R. B., 2011, *Uma Implementação do Processo de Medição usando a Spider-MPlan*, Anais do ENACOMP 2011, Catalão-GO.
- Carvalho, A., Lima, G., Costa, H., 2013, *Gerência Ágil de Projetos de Software Apoiada por uma Ferramenta Interativa Computacional com Suporte ao MPS.BR*, Anais do SBQS 2013, Salvador-BA.

- Castro, V. S., Oliveira, S. R. B., 2014, MR-MPS-SW e Métodos Ágeis: Um Apoio à Implementação do Processo de Projeto e Construção do Produto, Anais do Computer on the Beach 2014, Florianópolis-SC.
- Catunda, E., Nascimento, C., Cerdeiral, C., Santos, G., Nunes, E., Schots, N., Schots, M., Rocha, A., 2011, Implementação do Nível F do MR-MPS com Práticas Ágeis do Scrum em uma Fábrica de Software, Anais do SBQS 2011, Belém-PA.
- Chaves, R. O., Tavares, E. M. C., Oliveira, S. R. B., Favero, E. L., 2011, A Software Process Simulator Machine for Software Engineering Simulation Games, Anais do SBGames 2010, Florianópolis-SC.
- Chaves, R. O., Lobato, W. A. L., Tavares, E. M. C., Miranda, T. C., Oliveira, S. R. B., Favero, E. L., 2011a, Intelligent Behavior Simulation Module for Software Process Elements, Anais do SBGames 2011, Salvador-BA.
- Chaves, R. O., Tavares, E. M. C., Miranda, T. C., Oliveira, S. R. B., Favero, E. L., 2011b, DesigMPS: Um Jogo de Apoio ao Ensino de Modelos de Qualidade de Processos de Software baseado em Mapas Conceituais, Anais do COBENGE 2011, Blumenau-SC.
- Chen, X., 2010, Extraction and visualization of traceability relationships between documents and source code, Proceedings of the IEEE/ACM international conference on Automated software engineering - ASE '10, 505.
- Cleland-Huang, J., Chang, C. K., Ge, Y., 2002, Supporting event based traceability through high-level recognition of change events, In Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment, Washington, DC, IEEE Computer Society, pp. 595-602.
- Cleland-Huang, J. *et al.*, 2007, Best Practices for Automated Traceability, Computer (Volume: 40 , Issue: 6), (June), 27–35.
- Cleland-Huang, J. *et al.*, 2012, Breaking the big-bang practice of traceability: Pushing timely trace recommendations to project stakeholders, 20th IEEE International Requirements Engineering Conference, 231–240.
- CoEST: Center of excellence for software traceability, 2015, Disponível em: <http://www.CoEST.org>, Último Acesso em Novembro de 2015.
- Cohn, M., 2005, Agile Estimating and Planning”, Prentice Hall.
- Cohn, M., 2010, Managing Risk on Agile Projects with the Risk Burndown Chart, Disponível em <http://www.mountangoatsoftware.com/blog/>

managing-risk-on-agile-projects-with-the-risk-burndown-chart, Último Acesso em Dezembro de 2015.

- Connolly, T. M., Boyle, E. A., MacArthur, E., Hainey, T., Boyle, J. M., 2012, A systematic literature review of empirical evidence on computer games and serious games, *Computers & Education*, 59(2), pp 661–686.
- Conradi, R., Liu, C., 1994, *Process Modelling Languages: One or Many?*.
- Correia, B. C. S., 2005, *Portfolius: um modelo de gestão de portfólio de projetos de software*, Dissertação de Mestrado apresentada à Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco – CIN/UFPE, Recife-PE.
- Costa, C. S., 2010, *Uma abordagem baseada em evidências para o gerenciamento de projetos no desenvolvimento distribuído de software*, Dissertação de Mestrado Universidade Federal de Pernambuco, Recife-PE.
- Costa, H. R., Barros, M. O., Rocha, A. R. C., 2010a, *Maturidade em gerencia de portfólio de projetos de software: um estudo experimental*”, IX Simpósio Brasileiro de Qualidade de Software – SBQS 2010, Belém-PA.
- Costa, H. R., Barros, M. O., Rocha, A. R., 2010b, *Software project portfólio selection: A modern portfólio theory based technique*, *Proceedings of the 22nd International Conference of Software Engineering and Knowledge Engineering*.
- Costa, H. R., 2011, *Apoio à seleção de portfólio de projetos de software baseado na moderna teoria do portfólio*, Tese doutorado Rio de Janeiro. UFRJ/COPPE.
- Costa, T. S. A., Silva, B. W. F. V., Teixeira, D. J. S., Silva, G. P., Souza, P. J. S., Oliveira, S. R. B., Vasconcelos, A. M. L., 2015, *Spider-MSControl: Uma Ferramenta para Apoio ao Processo de Medição usando a Abordagem GQIM*, *Anais do WAMPS 2015*, Curitiba-PR.
- Deerwester, S. *et al.*, 1990, *Indexing by latent semantic analysis*, *J. Society for Information Science*, 41(6), 391–407.
- De Lucia, A., Oliveto, R., Tortora, G., 2008, *ADAMS Re-Trace: Traceability Link Recovery via Latent Semantic Indexing*, *ICSE’08, ACM/IEEE 30th International Conference on Software Engineering*, 839 – 842.
- Demarco, T., 1982, *Controlling software projects*, Yourdon Press Prentice-Hall.
- DOORS, 2005, *Dynamic Object Oriented Requirements System*, Disponível: em: <http://www.telelogic.com/doors>, Último Acesso em Novembro 2015.

- Dybå, T., Dingsoyr, T., Hanssen, G. K., 2007, Applying Systematic Reviews to Diverse Study Types: An Experience Report, in Proceedings of the 1st Empirical Software Engineering and Measurement (ESEM-07), Madrid-Spain: IEEE Computer Society, pp. 225-234.
- Estácio, B., Oliveira, S. R. B., 2010, Spider-MPlan: Uma Ferramenta para Apoio ao Processo de Medição do MPS.BR, XI Workshop de Software Livre (WSL 2010), Porto Alegre-RS.
- Falbo, R. A., 1998, Integração de Conhecimento em um Ambiente de Desenvolvimento de Software, Orientadora: Ana Regina Cavalcanti da Rocha. Tese de Doutorado, COPPE/UFRJ.
- Falbo, R. A., 2000, A experiência na definição de um processo padrão baseado no Processo Unificado, II Simpósio Internacional de Melhoria de Processo de Software, São Paulo-SP.
- Fenton, N., Pfleeger, S. L., 1997, Software Metrics. A rigorous and practical approach, PWS Pub.
- Ferreira, P., 2010, Rastreabilidade de ponto de função, Dissertação de Mestrado. UNIRIO, Rio de Janeiro-RJ.
- Frakes, W. B., Kang, K., 2005, Software Reuse Research: Status and Future, IEEE Transactions on Software Engineering, v. 31, n. 7, pp. 529-536.
- Fuggetta, A., 2000, Software Process: A Roadmap. In: Proceedings of The Future of Software Engineering, (ICSE'2000), Limerick-Ireland.
- Furtado, J. C. C., Oliveira, S. R. B., 2011a, Apoio Ferramental à Implementação do Processo de Aquisição de S&SC constante em Modelos de Qualidade, Anais da Sessão de Ferramentas do CBSOFT 2011, São Paulo-SP.
- Furtado, J. C. C., Oliveira, S. R. B., 2011b, Uma Implementação de Práticas de Gerência de Projetos de Aquisição Utilizando Ferramentas de Software Livre, Anais do WGPS 2011, Curitiba-PR.
- García-Borgoñon, L., Garcia-Garcia, J. A., Alba, M., Escalona, M. J., 2013, Software Process Management: A Model-Based Approach, 21st International Conference on Information Systems Development (ISD2012).
- Gibiec, M., Czauderna, A., Cleland-Huang, J., 2010, Towards minin replacement queries for hard-to-retrieve traces, Proceedings of the IEEE/ACM international conference on Automated software engineering - ASE '10, 245.
- GNU Operating System, 2015, Free Software Foundation, Disponível em: www.gnu.org. Último acesso em: Novembro/2015.

- Goold, A., Horan, P., 2002, Foundation Software Engineering Practices for Capstone Projects and Beyond, Proceedings of the Fifteenth Conference on Software Engineering Education and Training. IEEE: Covington, KY, USA, pp 140-146.
- Gotel, O., Finkelstein, A., 1997, Extended requirements traceability: Results of an industrial case study, In Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, Washington, DC, p. 169.
- Governo Federal Brasileiro, 2015, Portal Software Livre, Disponível em: www.softwarelivre.gov.br, Último acesso em: Novembro/2015.
- Greese, C., Rodrigues, R., 2000, Planejamento de programas de mensuração baseados em reutilização, XI Conferência Internacional de Qualidade de Software, Curitiba-PR.
- Groth, D. P., Robertson, E. L., 2001, It's All About Process: Project-Oriented Teaching of Software Engineering, in Proceedings of the Fourteenth Conference on Software Engineering Education and Training. Charlotte, USA, pp 7-17
- Guerra, A., Alves, A., 2004, Aquisição de Produtos e Serviços de Software, Campus, Rio de Janeiro-RJ.
- Gusmão, C. M. G., Moura, H. P., 2004, Gerência de Riscos em Processos de Qualidade de Software: uma Análise Comparativa, In: SBQS 2004 – III Simpósio Brasileiro de Qualidade de Software.
- Hawker, J. S., 2009, A Software Process Engineering Course, Proceedings of the 2009 American Society for Engineering Education Annual Conference, Austin, TX.
- Hayes, J. H., Dekhtyar, A., Osborne, J., 2003, Improving requirements tracing via information retrieval, In Proceedings of the 11th IEEE International Conference on Requirements Engineering, Washington, DC, p. 138.
- Hayes, J. H., Dekhtyar, A., Sundaram, S. K., 2005, Improving after-the-fact tracing and mapping: Supporting software quality predictions, IEEE Software, 22(6), 30-37.
- Hayes, J. H., Dekhtyar, A., Sundaram, S. K., 2006, Advancing candidate link generation for requirements tracing: the study of methods, IEEE TSE, 32/1, pp. 4-19.
- Hexsel, R., 2002, Propostas de Ações de Governo para Incentivar o Uso de Software Livre, Relatório Técnico 004/2002, Universidade Federal do Paraná – Departamento de Informática, Brasil.
- Highsmith, J., 2004, Agile Project Management: Creating Innovative Products, Addison-Wesley.
- Humphrey, W. S., 1989, Managing the Software Process, Addison-Wesley Professional.

- Hunter, R., Robinson, G., Woodman, I., 1997, Tool Support for Software Process Assessment and Improvement, University of Strathclyde, Department of Computer Science.
- IBM, 2010, IBM rational requisitepro software, Disponível em: <http://www-01.ibm.com/software/awdtools/reqpro/>, Último Acesso em Novembro de 2015.
- IEEE-CS/ACM, 2004, The Joint Task Force on Computing Curricula, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering 2004.
- Institute of Electrical and Electronics Engineers, 1990, Std 610.12 - IEEE Standard Glossary of Software Engineering Terminology, USA.
- Institute of Electrical and Electronics Engineers, 1999, Std 1517-1999 IEEE Standard for Information Technology—Software Life Cycle Processes—Reuse Processes –Description, IHS Standards, USA.
- Institute of Electrical and Electronics Engineers, 2004, SWEBOK – Guide to the Software Engineering Body of Knowledge, USA.
- Institute of Electrical and Electronics Engineers, 2006, ISO/IEC 16085 - IEEE Std 16085-2006 - Systems and software engineering - Life cycle processes - Risk management, USA.
- the International Organization for Standardization and the International Electrotechnical Commission, 1995, ISO/IEC 12207 Information Technology – Software Life Cycle Processes, Geneva: ISO.
- the International Organization for Standardization and the International Electrotechnical Commission, 1998, Information Technology – Software Product Evaluation - Part 5: Process for Evaluators, Geneva: ISO.
- the International Organization for Standardization and the International Electrotechnical Commission, 2001, ISO/IEC 12207 Amendment: Information Technology – Amendment 1 to ISO/IEC 12207, Geneva: ISO.
- the International Organization for Standardization and the International Electrotechnical Commission, 2004a, ISO/IEC 15504-1 Information Technology – Process Assessment – Part 1 – Concepts and Vocabulary, Geneva: ISO.
- the International Organization for Standardization and the International Electrotechnical Commission, 2004b, ISO/IEC 12207 Amendment: Information Technology – Amendment 2 to ISO/IEC 12207, Geneva: ISO.

- Itin, C. M., 1999, Reasserting the philosophy of experiential education as a vehicle for change, in the 21st century, *The Journal of Experiential Education* 22, pp 91–98.
- Jaccheri, M. L., Lago, P., 1997, Applying Software Process Modeling and Improvement in Academic Setting, *Proceedings of the 10th Conference on Software Engineering Education & Training*, Virginia Beach, Virginia, IEEE Computer Society Press, pp 13-27.
- Jiang, H. *et al.*, 2007, Traceability Link Evolution Management with Incremental Latent Semantic Indexing, *31st Annual International Computer Software and Applications Conference, (Compsac)*, 309–316.
- Kafai, Y. B., 2001, The educational potential of electronic games: from games-to-teach to games-tolearn, *Conference on playing by the rules: the cultural policy challenges of video games*, Chicago, Illinois.
- Karner, G. ,1993, Resource Estimation for Objectory Projects, Disponível em: <http://www.bfpug.com.br/Artigos/UCP/Karner%20%20Resource%20Estimation%20for%20Objectory%20Projects.doc>. Último Acesso em Novembro de 2015.
- Kellnet, M. I., Hansen G. A., 1988, *Software Process Modeling*, Pittsburgh.
- Kitchenham, B., 2007, Guidelines for performing Systematic Literature Reviews, In *Software Engineering*, Technical Report EBSE-2007-01, Departament of Computer Science Keele University, Keele.
- Kotonya, G., Sommerville, I., 1998, *Requirements engineering: processes and techniques*. Chichester, England: John Wiley.
- Koscianski, A., Soares, M. S., 2007, *Qualidade de Software – Aprenda as Metodologias e Técnicas Mais Modernas para o Desenvolvimento de Software*, 2ed, Novatec, São Paulo-SP.
- Krishnan, M. S., Kriebel, C. H., Kekre, S., Mukhopadhyay, T., 2000, An Empirical Analysis of Productivity and Quality in Software Products, *Management Science*, 46(6), pp 745-759.
- Kruchten, P., 2003, *Introdução ao RUP – Rational Unified Process*, Ciência Moderna, Rio de Janeiro-RJ.
- Kuhrmann, M., Fernández, D. M., Münch, J., 2013, Teaching Software Process Modeling, *Proceedings of the 35th International Conference on Software Engineering (ICSE'2013)*, San Francisco, United States, pp 18-26.

- Kusumoto, S., Matukawa, F., Inoue, K., Hanabusa, S., Maegawa, Y., 2010, Estimating Effort by Use Case Points: Method, Tool and Case Study, 10th International Symposium on Software Metrics.
- Lethbridge, T. C., 2000, What knowledge is important to a software professional?, *Journal Computer*, 33(5), IEEE Computer Society Press, Los Alamitos, CA, USA, pp 44- 50.
- Levine, H. A., 2005, *Project portfolio management: A practical guide to selecting projects, managing portfolios and maximizing benefits*, San Francisco, Jossey-Bass.
- Li, W., Hayes, J. H., 2013, Traceability Challenge 2013: Query + Enhancement for Semantic Tracing (QuEST), (TEFSE), 2013 International Workshop on Traceability in Emerging Forms of Software Engineering, 95–99.
- Lima, A. *et al.*, 2006, WebAPSEE: Um Ambiente Livre e Flexível Para Gerência de Processos de Software, VII Workshop de Software Livre (WSL 2006), Porto Alegre-RS.
- Lin J., *et al.*, 2006, Poirot: adistributed tool supporting enterprise-wide automated traceability, In: 14th IEEE international requirements engineering conference (RE 2006), pp 356–357.
- Lira, W. M. P., Oliveira, S. R. B., 2011, Um mapeamento de boas práticas para gerência de recursos humanos: uma análise de aderência ao PMBOK, Anais do III Congresso de Gerenciamento de Projetos da Amazônia, Manaus–AM.
- Lira, W. M. P., Barbosa, F., Yoshidome, E., Silva, E. J. F, Malcher, P. R. C., Oliveira, S. R. B., 2012, *Framework* de Processo para Gerência de Recursos Humanos: Uma Abordagem Baseada em Modelos e Normas de Qualidade de Processo de Software, Anais do JPC 2012, Puno-Peru.
- Lira, W. M. P., Oliveira, S. R. B., 2013, Um *Framework* de Práticas de Gerência de Recursos Humanos de suporte à Gerência de Projetos de Software, Anais do WGPS 2013, Salvador-BA.
- Macedo, K. O., Oliveira, S. R. B., 2011, Spider-CoCoMo: Uma Ferramenta de Apoio ao CoCoMo no Contexto da Melhoria do Processo de Software, Anais do SEMINF 2011, Belém-PA.
- Mäder, P., Gotel, O. C. Z., Philippow, I., 2009, Getting back to basics: Promoting the use of a traceability information model in practice, TEFSE '09. ICSE Workshop on Traceability in Emerging Forms of Software Engineering, 21–25.
- Mäder, P. *et al.*, 2013, Strategic Traceability for Projects, Software, IEEE, 58–66.

- Malcher, P. R. C., Oliveira, S. R. B., 2014, Indicações de Abordagens para Rastreabilidade de Requisitos no contexto do MR-MPS-SW por meio de uma Revisão Sistemática da Literatura, Anais do WAMPS 2014, Campinas-SP.
- Malcher, R. R. C., 2015, Catálogo de Abordagens de Apoio à Rastreabilidade de Requisitos baseado em uma Revisão Sistemática da Literatura, Dissertação de Mestrado, PPGCC/UFGA, Belém-PA.
- Malcher, P. R. C., Ferreira, D. A. L., Oliveira, S. R. B., Vasconcelos, A. M. L., 2015, Uma Abordagem para a Implementação Multi-Modelos de Qualidade de Software Adotando a CERTICS e o CMMI-DEV, Revista FSMA, Edição 16, Jul-Dez/2015.
- Maletic, J. I., Marcus, A., 2001, Supporting program comprehension using semantic and structural information, In Proc. of 23rd International Conference on Software Engineering, pages 103–112, Toronto.
- Manifesto Ágil, 2001, Manifesto para o desenvolvimento ágil de software, Disponível em <http://manifestoagil.com.br/>, Último Acesso em Dezembro de 2015.
- Markowitz, H., 1952, Portfólio selection, The Journal of Finance, v. 7, pp. 77 -91.
- Martins, A. F. *et al.*, 2006, ReqODE: Uma Ferramenta de Apoio à Engenharia de Requisitos Integrada ao Ambiente ODE, Anais da XIII Sessão de Ferramentas, XX Simpósio Brasileiro de Engenharia de Software, Florianópolis-SC.
- Mcfeeley, B., 1996, IDEALSM: A User's Guide for Software Process Improvement, Software Engineering Institute Handbook. Carnegie Mellon University. CMU/SEI-96-HB-001.
- Mello, M., 2011, Melhoria de Processos de Software Multi-Modelos Baseada nos Modelos MPS e CMMI-DEV, Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro-RJ.
- Mendes, H. J. P. S., Silva, B. W. F. V., Abreu, D. O., Ferreira, D. A. L., Leite, M. V. R., Leal, M. S. B., Oliveira, S. R. B., Spider-RM: Uma Ferramenta para Auxílio ao Gerenciamento de Riscos em Projetos de Software, Anais da Sessão de Ferramentas do CBSOFT 2014, Maceió-AL.
- Mezzomo, L. P., Oliveira, S. R. B., 2011a, Uma Abordagem para Definição, Avaliação e Melhoria do Processo de Software no Contexto de Modelos e Norma de Qualidade de Software, Proceedings X CSPC, Pucallpa-Peru.
- Mezzomo, L. P., Oliveira, S. R. B., 2011b, Framework de Definição, Avaliação e Melhoria de Processos Organizacionais Baseado em Modelos de Qualidade, In: I Simpósio de Informática e Geotecnologia de Santarém. Santarém-PA.

- Morais, S., 2009, Uma Abordagem para a Gerência de Recursos Humanos de Organizações de Software, Dissertação de Mestrado, UNIFOR, Fortaleza-CE.
- Morais, L., 2010, Qualidade de Software – Desvendando um requisito essencial no processo de desenvolvimento, Revista de Engenharia de Software, ed. 29, ano 3.
- Murta, L., Filho, R., Katsurayama, A., Santos, G., Rocha, A., 2008, A Experiência na Implantação do Processo de Gerência de Reutilização no Laboratório de Engenharia de Software da COPPE/UFRJ, In: Simpósio Brasileiro de Qualidade de Software, Florianópolis-SC.
- Nair, S., de La Vara, J. L., Sen, S., 2013, A review of traceability research at the requirements engineering conferencere@21, (RE), 2013 21st IEEE International Requirements Engineering Conference, 222–229.
- Narayan, N., Bruegge, B., Delater, A., Paech, B., 2011, Enhanced traceability in model-based CASE tools using ontologies and information retrieval, Fourth International Workshop on Managing Requirements Knowledge (MARK), 24–28.
- Nascimento, L. M. A. *et al.*, 2007, Uma abordagem para Medição em um Ambiente de Desenvolvimento de Software Centrado em Processos, XXXIII CLEI, San José–Costa Rica.
- Nascimento, H., Palma, J., Parente, R. S., 2011, A metodologia P³ no Gerenciamento de Portfólio de Projetos, Anais do VII WAMPS 2011 - Workshop Anual do MPS, 7., Campinas-SP.
- Navarro, E. O., 2006, SimSE: A Software Engineering Simulation Environment for Software Process Education, Dissertation, Doctoral dissertation, University of California, USA.
- Neiva, J., Oliveira, S. R. B., 2010, Uma Proposta de Apoio Sistêmico à Avaliação de Processos com Base no MA-MPS, SCAMPI e ISO/IEC 15504. In: Simpósio Brasileiro de Qualidade de Software – SBQS / Workshop de Teses e Dissertações em Qualidade de Software – WTDQS, Belém-PA.
- Neiva, J., Aviz, P. A., Oliveira, S. R. B., 2011, Spider-Appraisal: Uma Ferramenta de Apoio à Avaliação Integrada do MPS.BR e CMMI, Anais do WSL 2011, Porto Alegre-RS.
- Object Management Group, 2008, Software & Systems Process Engineering Meta-Model Specification.

- Object Management Group, 2011, BPMN – Business Process Model And Notation, Version 2.0.
- Oliveira, S. R. B., Vasconcelos, A. M. L., Rouiller, A. C., 2005, Uma Proposta de um Ambiente de Implementação de Processo de Software, Infocomp – Journal of Computer Science, Vol. 4, p. 71-78.
- Oliveira, S. R. B., 2007, ProDefiner: Uma Abordagem Progressiva para a Definição de Processos de Software no Contexto de um Ambiente Centrado no Processo, Tese de Doutorado, CIN/UFPE, Recife-PE.
- Oliveira, A., Guimarães, F., Fonseca, I., 2008, Utilizando Metodologias Ágeis para atingir Certificação MPS.BR na Powerlogic, Anais do SBQS 2008, Florianópolis-SC.
- Oliveira, S. R. B. *et al.*, 2011, SPIDER – Uma Proposta de Solução Sistêmica de um SUITE de Ferramentas de Software Livre de Apoio à Implementação do Modelo MPS.BR, Revista do Programa Brasileiro da Qualidade e Produtividade em Software, SEPIN-MCT, 2ª Edição, Brasília-DF.
- Oliveira, S. R. B., 2006, Processo de Software: Ambiente e Mecanismos de Execução, Exame de Qualificação do Doutorado, Centro de Informática, Universidade Federal de Pernambuco, Recife-PE.
- Park, R. E., Goethert, W. B., Florac, W. A., 1996, Goal-driven software measurement – A guidebook, Pittsburgh, PA, CMU/SEI-96- HB-002, Software Engineering Institute, Carnegie Mellon University.
- Paulk, M. C., Weber, C. V., Curtis, B. Chrissis, M. B., 1994, The Capability Maturity Model: Guidelines for Improving the Software Process, Addison-Wesley Professional, USA.
- Pavan, C., Stumpf, I. R. C., 2007, Revistas Brasileiras de Ciência da Computação: procedimentos de avaliação pelos pares, VIII ENANCIB – Encontro Nacional de Pesquisa em Ciência da Computação.
- Pfleeger, S. L., 1998, Software Engineering Theory and Practice, Prentice Hall PTR.
- Prieto-Díaz, R., 1991, Implementing faceted classification for software reuse, Communications of the ACM, Volume 34 Issue 5.
- Pressman, R. S., 2006, Engenharia de software, Tradução José Carlos Barbosa dos Santos, McGraw-Hill, 1056 p, Rio de Janeiro-RJ.
- Pressman, R. S., 2011, Engenharia de Software, uma Abordagem Profissional, AMGH editora Ltda, 7ª edição.

- Project Management Institute, 2008, The Standard for Portfolio Management, PMI Publishing Division.
- Project Management Institute, 2013, A Guide to the Project Management Body of Knowledge, Campus Boulevard, Newton Square, 5th Edition.
- Poppendieck, M., Poppendieck, T., 2011, Implementando o desenvolvimento LEAN de Software - Do conceito ao dinheiro, Bookman.
- Portela, C. S., Gomes, M., 2011, xSPIDER_ML: Especificação Técnica, Disponível em: http://www.spider.ufpa.br/projetos/xspider_ml/xSPIDER_ML.pdf 2011, Último Acesso em Novembro de 2015.
- Portela, C. S., 2012, Spider-PE: Um Framework De Apoio À Execução Flexível De Processos De Software Aderente A Modelos De Qualidade, Dissertação de Mestrado em Ciência da Computação – Centro de Informática, Universidade Federal de Pernambuco, Recife-PE.
- Portela, C. S., Vasconcelos, A. M. L., Oliveira, S. R. B., Gomes, M., Souza, M. R. A., Lira, W. M. P., 2012, xSPIDER_ML: Definition of a Software Processes Enactment Language adhering to the SPEM 2.0, Journal JSEA, 5(6), 375 – 384.
- Portela, C. S., Vasconcelos, A. M. L., Oliveira, S. R. B., Silva, A. A., Silva, E., 2014, Spider-PE: A Set of Support Tools to Software Process Enactment, Proceedings of The Ninth International Conference on Software Engineering Advances, Nice-France.
- Prikladnicki, R., Magalhães, A., 2010, Implantação de Modelos de Maturidade com Metodologias Ágeis: Um Relato de Experiências, Anais do WAMPS 2010, Campinas-SP.
- Rad, P. F., Levin, G., 2006, Project portfolio management, New York: International Institute for Learning Inc.
- Rai, A., Al-Hindi, H., 2000, The effects of development process modeling and task uncertainty on development quality performance, Information & Management, 37(6), pp 335–346.
- Raiol, I. I. S., Lima, L. O. D., Oliveira, S. R. B., 2014, Spider-QA: Uma Ferramenta de Apoio ao Processo de Garantia da Qualidade, Anais do WAMPS 2014, Campinas – SP.
- Ramesh, B., Jarke, M., 2001, Toward Reference Models for Requirements Traceability, IEEE Transactions on Software Engineering, vol. 27, no. 1, 2001, pp. 58-93.

- Reis, C. A. L., 2003, Uma Abordagem Flexível para Execução de Processos de Software Evolutivos, Tese de Doutorado em Ciência da Computação – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre-RS.
- Resende, D. K., Grego, J. B., Pimentel, N., Gonçalves, C. A., Junior, E. N. V., Ferreira, A. C., Kruehl, F., Batista, P. R., Neto, O. C. T., Cavalcanti, W., Godinho, H., Montoni, M., Nunes, E., Barreto, A., Rocha, A. R., 2009, Implementação do MPS.BR Nível F e CMMI-DEV Nível 2 na Red & White IT Solutions, WAMPS – Workshop Anual do MPS, Campinas – SP.
- Ribeiro, L., Gusmão, C. M. G., 2008, Definição de um Processo Ágil de Gestão de Riscos em Ambientes de Múltiplos Projetos, SIMS 2008 - Simpósio de Software e Mostra de Software Acadêmico, PUCRS, Uruguaiana, 2008.
- Rocha, A. R. Cavalcanti, Maldonado, J. C., Weber, K. C., 2001, Qualidade de Software: Teoria e Prática, Prentice Hall, São Paulo.
- Rocha, A. R. *et al.*, 2005, Estação TABA: Uma Infraestrutura para Implantação do Modelo de Referência para Melhoria de Processo de Software, IV Simpósio Brasileiro de Qualidade de Software (SBQS'05), Porto Alegre-RS.
- Rocha, A. R. C., Souza, S. S., Barcellos, M. P., 2012, Medição de Software e Controle Estatístico de Processos, 232 p, Brasília-DF.
- Santos, A., Silva, V., Junior, V., 2008, Desenvolvimento de um Processo de Software Aderente à ISO 9001:2000 Baseado no Processo Ágil Scrum, Anais do SBQS 2008, Florianópolis-SC.
- Santos, G., 2010, Revisão Sistemática, Mini-Curso, Simpósio Brasileiro de Qualidade de Software - SBQS 2010, Belém-PA.
- Santos, K. B. C., Neto, J. B. M. C., Cardoso, P. V. R., Oliveira, S. R. B., 2013, RisAgi: Uma Metodologia Ágil para Gestão de Riscos em Projetos de Desenvolvimento de Software, Anais do WAMPS 2013, Campinas-SP.
- Saskatchewan Education, 1991, Instructional Approaches: A Framework for Professional Practice.
- SBC – Sociedade Brasileira de Computação, 2006, Grandes Desafios de Pesquisa em Computação no Brasil – 2006-2016, São Paulo-SP.
- Schneider, L., Santos, G., Montoni, M., Rocha, A. R., 2004, Uma abordagem para Medição e Análise em Projetos de Desenvolvimento de software, III Simpósio Brasileiro de Qualidade de software, Brasília-DF.
- Schwaber, K., 2009, Guia do SCRUM, Disponível em: http://www.trainning.com.br/download/GUIA_DO_SCRUM.pdf, Último Acesso em Dezembro/2015.

- Schwarz, H., Ebert, J., Winter, A., 2009, Graph-based traceability: a comprehensive approach. *Software & Systems Modeling*, 9(4), 473–492.
- SEI - Software Engineering Institute, 2010a, Capability Maturity Model Integration (CMMI) for Development, Version 1.3, Carnegie Mellon, USA.
- SEI - Software Engineering Institute, 2010b, Capability Maturity Model Integration (CMMI) for Acquisition, Version 1.3, Carnegie Mellon, USA.
- SEI - Software Engineering Institute, 2011, Standard CMMI Appraisal Method for Process Improvement, Version 1.3, Carnegie Mellon, USA.
- Shnaider, L., 2003, Planejamento de Alocação de Recursos Humanos em Ambientes de Desenvolvimento de Software Orientados à Organização, Dissertação de Mestrado, COPPE/UFRJ, Rio de Janeiro-RJ.
- Silva, W. C., Martins, L. D. G., 2008, PARADIGMA: Uma Ferramenta de Apoio à Elicitação e Modelagem de Requisitos Baseados em Processamento de Linguagem Natural, XI Workshop em Engenharia de Requisitos, Catalonia-Espanha.
- Silva, F. G., Hoentsch, S. C. P., Silva, L., 2009, Uma análise das Metodologias Ágeis FDD e Scrum sob a Perspectiva do Modelo de Qualidade MPS.BR, Scientia Plena.
- Silva, A. A. C., Silva, E. J. S., Portela, C. S., Vasconcelos, A. M. L., Oliveira, S. R. B., 2012a, Spider-PE: Uma Ferramenta de Apoio à Implementação da Capacidade do MR-MPS Nível F e CMMI-DEV Nível 2, Anais do WAMPS 2012, Campinas-SP.
- Silva, A. A. C., Silva, E. J. S., Portela, C. S., Vasconcelos, A. M. L., Oliveira, S. R. B., 2012b, Spider-PE: Uma Ferramenta de Apoio à Execução de Processos de Software aderente ao CMMI-DEV e MR-MPS, Anais do WSL 2012, Porto Alegre-RS.
- Smith, H. T. G., 1997, *Software Quality Assurance: A Guide for Developers and Authors*, CRC Press, 1st. Edition.
- SOFTEX - Associação para Promoção da Excelência do Software Brasileiro, 2008a, MPS.BR: Lições Aprendidas, organizadores: Ana Regina Cavalcanti da Rocha e Kival Chaves Weber, Campinas-SP.
- SOFTEX - Associação para Promoção da Excelência do Software Brasileiro, 2008b, iMPS: Resultados de Desempenho de organizações que adotaram o modelo MPS, organizadores: Guilherme Horta Travassos e Marcos Kalinowski, Campinas-SP.

- SOFTEX - Associação para Promoção da Excelência do Software Brasileiro, 2012a, MPS.BR - Guia Geral do MR-MPS-SW, versão 2012 – MPS.BR:2012, Brasil.
- SOFTEX - Associação para Promoção da Excelência do Software Brasileiro, 2012b, Guia de Implementação – Parte 11: Implementação e Avaliação do MR-MPS-SW:2012 em Conjunto com o CMMI-DEV v1.3, Brasil.
- SOFTEX - Associação para Promoção da Excelência do Software Brasileiro, 2013, Melhoria do Processo de Software Brasileiro (MPS.BR) - Guia de Aquisição 2013, Brasil.
- SOFTEX - Associação para Promoção da Excelência do Software Brasileiro, 2015, MPS.BR – Melhoria de Processo do Software Brasileiro - Guia de Avaliação, Brasil.
- Solingen, R., Berghout, E., 1999, The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development, McGraw-Hill.
- Sommerville, I., 2003, Engenharia de Software, Addison Wesley, 6ª. Edição.
- Souza, A. D., 2008, Uma abordagem para gerência estratégica de portfólio: foco na seleção de projetos, Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, UFRJ.
- Souza, A. D., Rocha, A. R. C., Santos, G., Carmo, T. V. P., Alexandre, D. B., 2009, Uma abordagem para gerência estratégica de portfólio com foco na seleção de projetos, VIII Simpósio Brasileiro de Qualidade de Software – SBQS 2009, Ouro Preto-MG.
- Souza, M. R. A., Lira, W. M. P., Oliveira, S. R. B., 2010, Uma Implementação do Processo de Gerência de Configuração usando o Redmine, o SVN e a Spider-CL, Anais do III Workshop de Gerenciamento de Projetos de Software (WGPS), Belém-PA.
- Souza, M. R. A., Pereira, R. L., Araujo, M. V. C., Oliveira, S. R. B., 2011, Um Mapeamento de Boas Práticas de Gerência de Portfólio de Projetos no Contexto da Melhoria do Processo de Software, III Congresso de Gerenciamento de Projetos da Amazônia (PMI-AM. , 2011), Manaus-AM.
- Souza, M. R. A., Silva, A. A. C., Sinimbu, A. S., Oliveira, S. R. B., Vasconcelos, A. M. L., 2012a, Um diagnóstico das práticas do processo Gerência de Portfólio de Projetos em organizações de software de Belém, Anais do XI Simpósio Brasileiro de Qualidade de Software (SBQS 2012). Fortaleza-CE.
- Souza, M. R. A., Silva, A. A. C., Pereira, R. L., Araujo, M. V. C., Oliveira, S. R. B., 2012b, A experiência de uma fábrica de software com foco acadêmico na

- definição de um processo para gerência de portfólio de projetos. XI Simpósio Brasileiro de Qualidade de Software/V Workshop de Gerenciamento de Projetos de Software (SBQS 2012 - WGPS), Fortaleza-CE.
- Souza, M. R. A., Oliveira, S. R. B., 2013, A Process Framework for Software Project Portfolio Management in Accordance to Quality Standards, Proceedings of CLEI 2013, Vargas-Venezuela.
- Staples, M. *et al.*, 2007, An Exploratory Study of Why Organizations Do Not Adopt CMMI, In Journal of Systems and Software, Elsevier, Vol. 80, Issue 6, pp. 883-895.
- Sussman, B. C. *et al.*, 2009, Version Control with Subversion: For Subversion 1.6: (Compiled from r3649), e-Book, O'Reilly Media.
- Teles, M. P., Oliveira, S. R. B., 2010, Spider-QA: A Systematic Approach for Quality Assurance Process, Proceedings of the IASTED International Conference Software Engineering and Applications (SEA 2010), Marina Del Rey – USA.
- Teles, M. P., 2011, SPIDER-QA: Um Ferramental de Apoio ao Processo de Garantia da Qualidade no Contexto de Modelos e Norma para Processo de Software, Dissertação de Mestrado do Programa de Pós-Graduação em Ciências da Computação, Universidade Federal do Pará, Belém-PA.
- Travassos, G. H., 1994, O Modelo de Integração de Ferramentas da Estação TABA, Orientadora Ana Regina Cavalcanti da Rocha, Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, Brasil.
- Veethil, S. T., 2013, Risk Management in agile, disponível em <http://www.scrumalliance.org/community/articles/2013/2013-may/risk-management-in-agile>, Último Acesso em Dezembro de 2015.
- Wangenheim, C. G. V., Shull, F., 2009, To Game or Not to Game?, IEEE Software, 26 (2), IEEE Computer Society Press Los Alamitos, CA, USA, pp 92–94.
- Wangenheim, C. G., Thiry, M., Kochanski, D., 2009, Empirical evaluation of an educational game on software measurement, Empirical Software Engineering, 14(4), Kluwer Academic Publishers Hingham, MA, USA, pp 418 – 452.
- Wangenheim, C. G. V., Hauck, J. C. R., 2010, Teaching Software Process Improvement and Assessment, Proceedings of 17th European Systems & Software Process (EuroSPI'2010), Grenoble, France.
- Weber, K. C. *et al.*, 2004a, Modelo de Referência para Melhoria de Processo de Software: uma abordagem brasileira, In: XXX Conferencia Latinoamericana de Informatica (CLEI2004), Sesión. 2004a. p. 20-10.

- Weber, K. C. *et al.*, 2004b, Uma estratégia para melhoria de processo de software nas empresas brasileiras, Proceedings of QUATIC, p. 73-78, 2004b.
- Xiaoguang, Y., Xiaogang, W., Linpin, L., Zhuoning, C., 2008, Research on organizational-level software process improvement model and its implementation, Proceedings - International Symposium on Computer Science and Computational Technology, ISCSCT 2008.
- Yelin, K. C., 2007, Linking strategy and project portfolio management, in Project Portfolio Management: A practical guide to selecting projects, managing portfolios and maximizing benefits, First edition, San Francisco, Jossey-Bass.
- Yi, C., Ning, Y., Jin, Q., 2008, A fuzzy multi-criteria evaluation approach for RD project selection, International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008.
- Yoshidome, E. Y. C., Souza, M. R. A., Lira, W. M. P., Oliveira, S. R. B., 2010, Uma Implementação do Processo de Gerência de Projetos Usando Ferramentas de Software Livre, Anais do WAMPS 2010, Campinas-SP.
- Yoshidome, E. Y. C., Souza, M. R. A., Lira, W. M. P., Oliveira, S. R. B., 2011, Apoio à Implementação de Estimativas no Contexto do Processo de Gerência de Projetos do MPS.BR e CMMI, Anais do ENACOMP 2011, Catalão-GO.
- Yoshidome, E. Y. C., Souza, M. R. A., Lira, W. M. P., Oliveira, S. R. B., Vasconcelos, A. M. L., 2012, Um Apoio Sistematizado à Implementação do Processo de Desenvolvimento de Requisitos do MPS.BR e CMMI a partir do Uso de Ferramentas de Software Livre, Anais do WER 2012, Bueno Aires-Argentina.
- Zamli, K. Z., Lee, P. A., 2001, Taxonomy of Process Modeling Languages.
- Zaniro, D., Fabbri, S., 2008, Um Processo Guiado para o Levantamento e Modelagem de Requisitos de Aplicações Web Baseado em Objetivos e Casos de Uso, XI Workshop em Engenharia de Requisitos, Catalonia-Espanha.
- Zanlorenci, E. P., Burnett, R. C., 2000, Ferramenta de Apoio aos Processos da Engenharia de Requisitos, na Fase de Projetos, III Workshop em Engenharia de Requisitos, Rio de Janeiro-RJ.
- Zhang, Y., Fan, Z., 2011, Uncertain linguistic multiple attribute group decision making approach and Its application to software project selection, Journal of Software.

<i>Título</i>	Abordagens para Apoio à Implementação da Melhoria do Processo de Software
<i>Autores</i>	Sandro Ronaldo Bezerra Oliveira e Alexandre Marcos Lins de Vasconcelos
<i>Projeto Gráfico/Capa</i>	Bruna Andrade
<i>Revisão de Texto</i>	Prof. Dr. Sandro Ronaldo Bezerra Oliveira

<i>formato</i>	16,0 x 23,0 cm
<i>fontes</i>	Minion Pro, Myriad Pro

<i>Finalização</i>	EdUFPE Rua Acadêmico Hélio Ramos, 20, Várzea Recife, PE CEP: 50.740-530 Fone: (0xx81) 2126.8397 Fax: (0xx81) 2126.8395 www.ufpe.br/edufpe livraria@edufpe.com.br
--------------------	--

