



**Programa Brasileiro da
Qualidade e Produtividade
em Software
Projetos Ciclo 2010**

Presidente da República

Dilma Vana Rousseff

Ministro da Ciência e Tecnologia

Aloizio Mercadante Oliva

Secretário de Política de Informática

Virgílio Augusto Fernandes Almeida

Diretora de Políticas e Programas Setoriais em Tecnologias da Informação

Marylin Peixoto da Silva Nogueira

Coordenador Geral de Serviços e Programas de Computador

Rafael Henrique Rodrigues Moreira

PBQP Software

Coordenador Geral

Kival Chaves Weber

Coordenador de Projetos e Indicadores

Diva da Silva Marinho

Compilação, organização e revisão

Diva da Silva Marinho

Ministério da Ciência e Tecnologia
Secretaria de Política de Informática



Revista do Programa Brasileiro da Qualidade e Produtividade em Software

2ª edição
Junho/2011

Editorial

Essa é mais uma publicação que a Secretaria de Política de Informática - SEPIN elabora, com foco na qualidade do setor de software brasileiro. Trata-se da segunda edição da Revista - Programa Brasileiro da Qualidade e Produtividade em Software, que contém artigos que descrevem os projetos concluídos que foram submetidos ao Ciclo de Projetos 2010 do Programa.

Os projetos concluídos foram avaliados por um Comitê de Avaliadores do Programa com objetivo de conceder o Prêmio Dorgival Brandão Junior da Qualidade e Produtividade em Software aos três melhores avaliados.

Os vinte e seis artigos reunidos nessa segunda edição da revista representam uma parte significativa do trabalho que a comunidade brasileira envolvida com o tema de qualidade de software desenvolve com vistas a alcançar a melhoria dos processos e produtos do software brasileiro. Congratulo os autores e as instituições que contribuíram para mais uma edição da revista.

Rafael Henrique Rodrigues Moreira
Especialista em Políticas Públicas e Gestão Governamental
Coordenador Geral de Software e Serviços
MCT/SEPIN

Sumário

Artigos dos Projetos 2010

Agile Brazil 2010 - Conferência Brasileira sobre Métodos Ágeis de Desenvolvimento de Software

Rafael Prikladnicki _____

Elaboração de Processo para Fiscalização de Contratos do Domínio da Tecnologia da Informação

Carlos Constantino Moreira Nassur _____

Um Estudo Preliminar sobre Tipos de Personalidade Adequados a Inspectores de Software

Tayana Conte, Nayane Maiaa, Anna Beatriz Marquesa, Emília Mendesb,Guilherme Horta Travassos _____

Fatores Críticos de Sucesso em Programas de Melhoria de Processos de Software - Um Estudo Qualitativo

Davi Viana, Olavo Matos, Vinícius Secatti, Tayana Conte _____

GPO - Gestão de Processos Organizacionais uma Metodologia para Promoção da Qualidade

Renato Luiz Della Volpe, Ana Cecília Peixoto Zabeu, Sergio Massao Jomori, Edvar Pera Junior _____

Melhoria do Processo de Aquisição de Software de 2008 a 2010

Kival Chaves Weber, Danilo Scalet _____

Qualidade em Software como Desenvolvimento Sustentável para o Pólo de Software AmazonSoft

Paulino W. P. Viana, Tayana U. Conte _____

Curso de Residência em Arquitetura de Software: Uma Parceria UFPA, UNAMA e CESUPA com AmazonCorp, CTIC-UFPA e PRODEPA

Carla Alessandra Lima Reis, Claudio Roberto de Lima Martins, Marcos Venícios Araújo, Rodrigo Quites Reis

O Uso de PBL no Ensino de Engenharia de Software

Simone C. dos Santos, Jones O. Albuquerque, Silvio R. L. Meira _____

Desenvolvimento de um Método e uma Ferramenta para a Reutilização de Processos de Desenvolvimento de Software

André Luiz Peron Martins Lanna, Carlos Alberto Marques Pietrobon _____

Definição do Processo APD e Certificação no Nível F do MR-MPS Utilizando Práticas do Processo Unificado Associadas a Práticas SCRUM

Tatiane Silva _____

Apoio Ferramental de Software Livre para Implementação do Processo de Gerência de Configuração do MPS.BR

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

Projeto CMMI-DTP: Institucionalização do Modelo CMMI na Dataprev

Guilherme Tavares Motta, Rosana Fernandes Osório _____

Apoio Ferramental de Software Livre para Implementação do Processo de Medição do Modelo MPS.BR

Bernardo José da Silva Estácio, Wallace Michel Pinto Lira, Sandro Ronaldo Bezerra Oliveira _____

Apoio Sistematizado de Software Livre para Implementação do Processo de Gerência de Requisitos do MPS.BR

Ewelton Yoshio Chiba Yoshidome, Alexandre Brito Cardias Junior, Luciana Neves Bentes, Sandro Ronaldo Bezerra Oliveira _____

SPIDER: Uma Proposta de Solução Sistêmica de um SUITE de Ferramentas de Software Livre de Apoio à Implementação do Modelo MPS.BR

Sandro Ronaldo Bezerra Oliveira, Ewelton Yoshidome, Wallace Lira, Júlio Furtado, Jñane Neiva, Marília Paulo2, Renan Barros, Maurício Ronny, Leonardo Mezzomo, Bernardo Estácio, Kaio Valente, Gleise Baldez, Pedro Afonso, Kleverton Macedo, Fabrício Alho, Sivaldinho Oliveira _____

Uma Investigação sobre os Fatores Críticos de Sucesso em Iniciativas de Melhoria de Processos de Software

Mariano Angel Montoni, Ana Regina Cavalcanti da Rocha _____

Teste de Software no Contexto do Software Público Brasileiro

Adalberto Nobiato Crespo, Paulo Bueno, Miguel Argollo, Celso Barros, Mario Jino _____

Definição de um Guia de Apoio às Decisões em Definições de Arquiteturas de Software

Júlio César de Paula Silva, André Luiz Peron Martins Lanna _____

Estimativa de Esforço de Teste no Auxílio da Garantia da Qualidade de Software

Priscila Pereira de Souza, Marcelo Werneck Barbosa, Anybal Rocha da Silva _____

FAMP: Ferramenta de Avaliação para Melhoria de Processos de Desenvolvimento de Software

Elson Cardoso Siquara, Fábio Britto de Carvalho Almeida, Laís do Nascimento Salvador _____

Projeto MDA4Eclipse

Fabio P. Marzullo, Jano M. de Souza, Marcos M. Müller _____

Institucionalização de um Processo Padrão de Desenvolvimento de Software

Amélia de Lima Guedes, João Carlos Testi Ferreira, Anderson Ricardo Frezza, Sidinei José Rossoni _____

Spider-PM: Ferramenta de Software Livre de Apoio à Modelagem de Processos de Software

Renan Sales Barros, Sandro Ronaldo Bezerra Oliveira _____

Spider-QA: Uma Ferramenta de Software Livre de Apoio à Garantia da Qualidade

Marília Paulo Teles, Sandro Ronaldo Bezerra Oliveira _____

Utilizando *Grounded Theory* para Compreender a Aceitação de uma Técnica de Elicitação de Requisitos

Amadeu Anderlin Neto, Cristina Araújo¹, Tayana Conte _____

Prêmio Dorgival Brandão Júnior _____

Artigos dos Projetos Concluídos Ciclo 2010

Agile Brazil 2010 - Conferência Brasileira sobre Métodos Ágeis de Desenvolvimento de Software

Rafael Prikladnicki

Faculdade de Informática - PUCRS - POA - RS - Brasil

rafaelp@pucrs.br

Resumo. Neste artigo apresenta-se o relatório final do projeto 1.03, cujo objetivo foi planejar e realizar a Conferência Brasileira sobre Métodos Ágeis de Desenvolvimento de Software (Agile Brazil 2010). A Agile Brazil é uma conferência nacional sem fins lucrativos organizada por representantes das principais comunidades ágeis brasileiras. O evento tem como objetivo promover a comunicação e a colaboração entre seus integrantes visando à disseminação coordenada da cultura ágil por todo o país. O evento foi realizado em Porto Alegre, contando com a participação de mais de 800 inscritos, 17 patrocinadores, diversas palestras e uma trilha acadêmica.

1. Introdução

Há mais de 10 anos, consultores renomados do mercado de Tecnologia de Informação vêm trabalhando de uma forma diferente para desenvolver software. Suas equipes buscam melhorar processos de comunicação, com foco em entrega de software para os clientes poderem usar e avaliar, buscam mais colaboração dos clientes e, principalmente, buscam responder a mudanças para estar alinhado as necessidades de negócio dos clientes que atendem (Beck et al, 2001).

Estas equipes possuem processos, usam ferramentas para apoiar estes processos, mas acima de tudo o foco nas pessoas é que dá o diferencial no funcionamento do dia a dia. Práticas de modelagem, estimativas, planejamento, desenvolvimento iterativo, qualidade e teste de software são características que se tornam cada vez mais importantes para minimizar riscos no desenvolvimento de software.

O desenvolvimento de software é uma atividade criativa, de excelência técnica e que se realiza em equipe. Um trabalho de pura sinergia. Qualidade é um valor percebido por clientes, um valor subjetivo muitas vezes, e alvo de constante busca por parte das empresas que prestam serviços de desenvolvimento de software. Qualificar, mostrar diferenciais, buscar modelos de qualidade passa a ser relevante. Mais que isto, criar formas de gerar retorno de investimento (*Return of Investment* - ROI) para clientes se torna um assunto cada vez mais presente para empresas no mercado de TI.

Desta forma, em 2010 foi organizada, em Porto Alegre, a **Agile Brazil**, Conferência Brasileira de Metodologias Ágeis para Desenvolvimento de Software, cujo objetivo foi tratar todas estas questões, reunindo profissionais interessados no tema. Foi um evento de retomada da mobilização acerca deste tema no Brasil, organizado a partir da união de quatro eventos que vinham sendo organizados no país nos últimos anos, quais sejam: Maré de Agilidade, organizado em diversas partes do Brasil; Encontro Ágil, em São Paulo; Agile Brazil, no Rio de Janeiro; e Agile Weekend, em Porto Alegre, além de lições aprendidas da II Jornada Latino Americana de Metodologias Ágeis, organizada em Florianópolis no ano de 2009. Desta forma, este artigo relata os resultados alcançados com este evento, como um projeto do ciclo 2010 do PBQP-SW.

2. Objetivos e Justificativa

Nos últimos anos, é visível o crescente interesse por Métodos Ágeis no mundo, fato este corroborado pelas diversas conferências específicas, como *Agile* e *XP Conference*. Esse interesse também ocorre no Brasil com diversas iniciativas, tais como o Encontro Ágil, Agile Brazil, Maré de Agilidade e Agile Weekend, entre outros (Figura 1). Apenas entre os anos de 2008 e 2009, por exemplo, os eventos organizados no Brasil receberam aproximadamente 2.800 participantes.



Figura 1. Histórico dos eventos de métodos ágeis no Brasil – 2008 e 2009

Mas ainda não existia no país um evento de abrangência nacional. Desta forma, no dia 27 de Agosto de 2009, um grupo de brasileiros que estava participando da *Agile Conference* em Chicago se reuniu e decidiu propor a organização da *Agile Brazil 2010*, um evento nacional sobre métodos ágeis para desenvolvimento de software, que contemplaria ainda o primeiro Workshop Brasileiro de Métodos Ágeis (WBMA), um evento acadêmico sobre métodos ágeis no desenvolvimento de software.

O objetivo do projeto foi então a realização da Conferência Brasileira sobre Métodos Ágeis de Desenvolvimento de Software, ou *Agile Brazil*. A *Agile Brazil* foi definida como uma conferência nacional sem fins lucrativos organizada por representantes das principais comunidades ágeis brasileiras. O evento teve como objetivo promover a comunicação e a colaboração entre seus integrantes visando à disseminação coordenada da cultura ágil por todo o país.

3. Metodologia de Execução

A organização da *Agile Brazil* foi definida em algumas etapas, a saber: definição da cidade sede e do local de realização; formação das equipes nacional e local de organização, planejamento do evento, incluindo trilhas e palestrantes convidados; promoção de um concurso de logo; submissão e avaliação de trabalhos; execução do evento. Para a definição da cidade sede, foram consideradas três propostas, sendo que Porto Alegre foi escolhida por apresentar a melhor relação custo-benefício, com a PUCRS como local de realização. Definiu-se uma equipe de organização distribuída nacionalmente, com 14 integrantes, conforme apresenta a Figura 2.

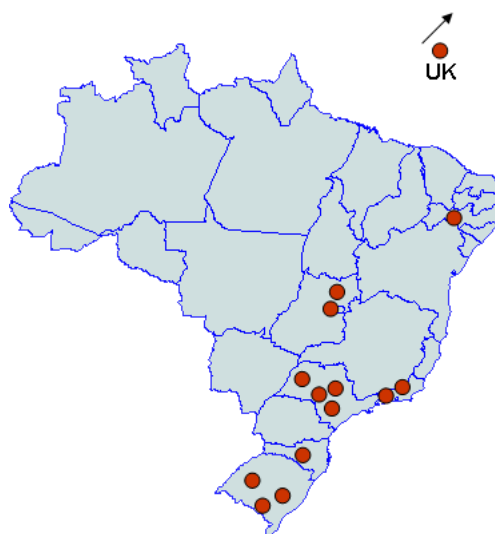


Figura 2. A distribuição da equipe de organização

A equipe de organização local contou com representantes da comunidade ágil local bem como professores da Faculdade de Informática e profissionais do Centro de Eventos da PUCRS. Para o planejamento do evento, foram definidas trilhas de palestras para iniciantes, praticantes e experientes, em temas técnicos e de gestão.

Além disso, foram definidos os tipos de apresentação, tais como tutoriais, workshops e palestras. Além disso, foram definidos os nomes de três palestrantes internacionais, Martin Fowler (um dos autores do Manifesto Ágil publicado em 2001), Philippe Kruchten (um dos autores do RUP – processo unificado, coordenador da Agile Vancouver e nome forte na comunidade ágil mundial) e David Hussman (ganhador do *Gordon Pask Award* em 2009, concedido anualmente pela *Agile Alliance*). Para a submissão de trabalhos à trilha principal foi desenvolvido um sistema *Open Source*, que inovou ao permitir que as propostas de apresentação pudessem ser visualizadas por todos os cadastrados, bem como receber comentários, antes mesmo do processo de avaliação (Figura 3).



Figura 3. Sistema de submissão desenvolvido

Finalmente, o evento foi então executado, com resultados bastante positivos, apresentados a seguir.

4. Resultados Obtidos

O projeto produziu os seguintes resultados:

- **Software:** conforme ilustrado na Figura 3, a equipe nacional de organização desenvolveu um sistema *Open Source* para a submissão de trabalhos na trilha da indústria (trilha principal). Este sistema está sendo utilizado para edição 2011 do evento e foi solicitada para ser utilizada em outros eventos no país.

- **Trabalho acadêmico:** um resultado indireto da organização do evento foi a possibilidade de alunos de graduação e pós-graduação (44% dos participantes eram estudantes) utilizarem o espaço para aprimorar suas idéias, coletar dados para seus trabalhos e consolidar parcerias acadêmicas. Desta forma, diversos trabalhos acadêmicos desenvolvidos no país acabaram sendo beneficiados pelo evento.

- **Portal/site na web:** foi desenvolvido um portal para o evento (www.agilebrazil.com), bem como um site para o sistema de submissões que está sendo utilizado para o evento de 2011.

- **Publicação técnica:** foi desenvolvido um relatório sobre o evento, bem como anais do workshop acadêmico.

- **Certificação:** o evento contou com dois cursos de certificação de Scrum (método ágil com foco em gestão de projetos). São eles: *Certified Scrum Master (CSM)* e *Certified Scrum Product Owner (CSPO)*. Os dois cursos mobilizaram um total de 60 participantes.

- **Cursos de capacitação:** diversos cursos foram oferecidos para os participantes do evento, totalizando mais de 100 vagas.

- **Qualificação de pessoas:** o evento contou com a participação de mais de 800 pessoas, sendo o maior evento de métodos ágeis já organizado na América Latina.

- **Evento:** o evento em si se desdobrou em diversos resultados, apresentados a seguir:

a. Concurso de logo

Um concurso foi planejado de forma a mobilizar a comunidade ágil nacional para propor a logomarca do evento. Ao todo, quatorze propostas foram recebidas, conforme a Figura 4. O vencedor foi contemplado com cinco inscrições para o evento.



Figura 4. Propostas de logo recebidas.

b. Colaboradores envolvidos

O evento mobilizou 14 voluntários na equipe nacional de organização, 6 voluntários na equipe local de organização, 12 monitores, 10 profissionais do Centro de Eventos da PUCRS, 4 colaboradores, 19 profissionais atuando no comitê de programa do WBMA, 25 profissionais atuando no comitê de programa do evento principal e 70 palestrantes (incluindo o WBMA).

c. Propostas de trabalhos submetidas

Ao todo, foram submetidas 150 propostas de trabalhos para a trilha da indústria e 33 propostas para a trilha acadêmica. Foram selecionados 36 trabalhos para a trilha da indústria (24%) e 11 trabalhos para a trilha acadêmica (33%). Todos os trabalhos foram revisados por três membros dos comitês de programa, seguindo-se de uma etapa de consenso. Para a trilha acadêmica utilizou-se a ferramenta *EasyChair*.

d. Abrangência dos trabalhos submetidos e selecionados

São Paulo, Rio Grande do Sul, Rio de Janeiro, Distrito Federal e Pernambuco foram os Estados com o maior número de trabalhos submetidos para a trilha da indústria (trilha principal), sendo a grande maioria propostas do tipo praticante, seguido de iniciante e experiente. Já para a trilha acadêmica a maior concentração de trabalhos submetidos foi em São Paulo, seguido de Rio Grande do Sul e Pernambuco. No total, 29 empresas ou instituições estavam representadas nas propostas de trabalhos aceitos.

e. Número de inscritos e abrangência dos participantes

O evento contou com a participação de 820 pessoas, sendo 44% de estudantes. Além disso, foi um dos maiores eventos de Engenharia de Software organizados no Brasil em 2010 e o maior evento de métodos ágeis já organizados na América Latina. Um aspecto interessante que merece destaque em relação ao número de participantes é que o evento havia sido planejado inicialmente para 550 participantes (um aumento de 20% em relação ao maior evento de métodos ágeis organizado no Brasil em 2009). Mas o evento alcançou sua lotação um mês antes da data de início, o que motivou uma busca por um novo local dentro da própria PUCRS, para então abrir para novas inscrições. A Figura 5 ilustra a evolução das inscrições, com um período de espera durante as negociações para o novo espaço (entre 23 de Maio e 02 de Junho de 2010).

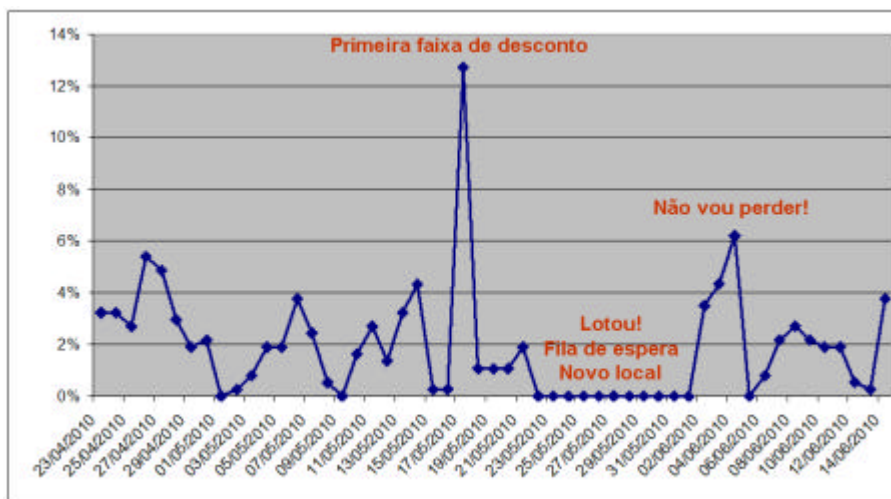


Figura 5. A evolução das inscrições

A Figura 6 ilustra a abrangência dos participantes do evento, totalizando 18 Estados do Brasil, além de participantes do Canadá, Estados Unidos e Austrália.

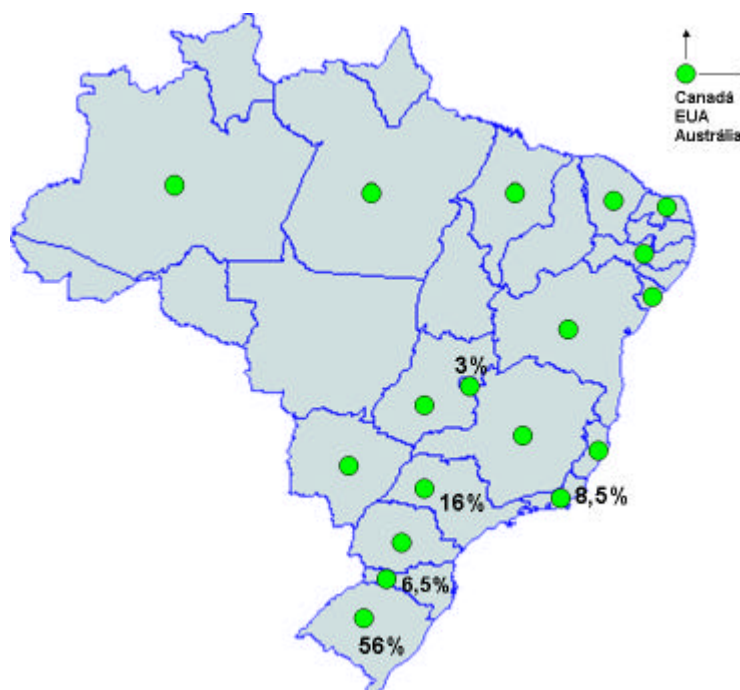


Figura 6. Abrangência dos participantes do evento

f. Patrocinadores e apoiadores

O Agile Brazil 2010 contou com o patrocínio de 17 empresas e o apoio de diversas entidades e grupos regionais, tais como o Grupo de Usuários de Métodos Ágeis do Rio Grande do Sul (GUMA-RS), a Sociedade Brasileira de Computação (SBC), o Ministério da Ciência e Tecnologia (MCT), a AgilCoop, a Sucesu-RS, entre outros.

5. Aplicabilidade dos resultados

O objetivo do evento foi plenamente alcançado, e muito do que se desenvolveu na edição de 2010 foi utilizado na edição de 2011. O evento contribuiu para estimular discussões sobre o tema de forma mais estruturada, além de fomentar a colaboração e o aumento do conhecimento sobre métodos ágeis nas empresas, Universidades e profissionais interessados. Como o projeto desenvolvido não tem uma forma única de ser avaliado em relação à aplicabilidade de seus resultados, utilizamos os feedbacks fornecidos pelos próprios participantes de forma espontânea através de blogs, sites pessoais, *twitter*, entre outros. Como exemplo, além de diversos comentários no *twitter*, identificou-se pelo menos 45 páginas na internet com relatos de impressões sobre o evento, tais como <http://onca.st/blog/?p=840>. Entre diversos feedbacks, reproduzimos dois deles neste relatório:

“A equipe de organização foi composta por 14 pessoas e o que mais me impressionou foi que a primeira reunião presencial desta equipe ocorreu no hall do hotel 1 dia antes do evento. Isto demonstra o poder que indivíduos auto-organizados que trabalham em busca de um mesmo objetivo têm, mesmo estando todos trabalhando remotamente. Eu particularmente, só conhecia metade desta equipe. Foi um sentimento muito interessante encontrar as pessoas que eu não conhecia e dar rostos aos nomes. A propósito, trabalhar com estas pessoas foi muito prazeroso e me proporcionou um aprendizado muito grande. Estão todos de parabéns por terem feito um evento desta qualidade.”

“... o nível técnico do evento foi excelente e os mais de 800 participantes em geral se mostraram muito satisfeitos com o que viram.”

Finalmente, talvez a melhor demonstração da aplicabilidade dos resultados do evento de 2010 foi o início das atividades para o evento de 2011. Uma chamada pública de propostas foi divulgada, e três cidades se candidataram para ser a sede. Além da decisão da sede de 2011, também decidiu-se a sede de 2012, garantindo assim um processo mais sistemático e tranquilo para a organização das edições futuras. Além disso, foi formalizado o comitê nacional de organização e outros comitês de apoios, bem como a criação de regras para futuras edições e o padrão de relatório final do evento para ser utilizado nas edições seguintes.

6. Características Inovadoras

O evento organizado como resultado deste projeto foi inovador em diversos aspectos. Primeiro, houve a retomada da mobilização nacional sobre métodos ágeis. Até então havia diversos eventos locais, latino-americanos ou globais, e faltava um evento nacional. Desta forma, o evento acabou se consolidando como uma ação inovadora e integrada da comunidade ágil nacional, colocando o Brasil definitivamente no mapa deste importante tema para a indústria de desenvolvimento de software. Além disso, o desenvolvimento de um sistema *Open Source* para a submissão dos trabalhos, com a possibilidade de consultar e comentar os trabalhos submetidos criou um mecanismo muito interessante e transparente do ponto de vista de montagem do programa do evento. Finalmente, também pode ser destacado o concurso de logo e a articulação com diversas empresas, entidades e governo.

7. Conclusões e Perspectivas Futuras

O término da Agile Brazil 2010 foi seguido de uma seção de retrospectiva e lições aprendidas. Esta seção é ilustrada na Figura 6.



Figura 7. Sessão de retrospectiva

As principais lições foram documentadas de forma a serem incorporadas no evento de 2011, cuja organização já está em andamento. Ao final do evento houve um sentimento de dever cumprido, de um projeto bem sucedido, mas mais do que isso, a certeza de que o evento contribuiu para disseminar ainda mais os conceitos relacionados com os métodos ágeis de desenvolvimento de software no Brasil, cumprindo com o objetivo definido no início: a criação de espaços de discussão em nível nacional acerca deste importante tema para a indústria de software nacional.

8. Referências Bibliográficas

Beck et al. (2001). “Agile Manifesto”, disponível em <http://agilemanifesto.org/>.

Elaboração de Processo para Fiscalização de Contratos do Domínio da Tecnologia da Informação

Carlos Constantino Moreira Nassur
Centro de Informática da Câmara dos Deputados
carlos.nassur@camara.gov.br

Abstract. *This paper reports Project 2.03, submitted by the Chamber of Deputies' Department of Information Technology for 2010's Brazilian Quality and Productivity in Software Program (PBQP-S) cycle. The Project was conceived to produce an effective inspection process to increment the House's Information Technology agreements results and to mitigate their inherent execution risks. The following text describes the problem context, the goals that should be aimed, and the justification for the developed work, the methodology adopted along the project phases, the results achieved and the future perspectives for the matter.*

Resumo. *Este artigo relata o Projeto 2.03, submetido pelo Centro de Informática da Câmara dos Deputados ao ciclo de 2010 do Programa Brasileiro de Qualidade e Produtividade em Software (PBQP-S), do Ministério da Ciência e Tecnologia. O projeto foi concebido para produzir um processo de fiscalização e acompanhamento, com vistas em maximizar os resultados dos contratos administrativos firmados pela Câmara dos Deputados e, assim, mitigar os riscos inerentes à sua execução. O texto a seguir descreve o contexto do problema, os objetivos e as justificativas para o desenvolvimento do trabalho, a metodologia adotada na sua execução, os resultados já alcançados e as perspectivas futuras.*

1. Introdução

O Centro de Informática (Cenin) é o órgão da estrutura administrativa da Câmara dos Deputados responsável por realizar as atividades normativas e executivas relacionadas à informática [1]. É, portanto, a unidade responsável por disponibilizar à Casa serviços, recursos, sistemas, soluções e demais facilidades do domínio da Tecnologia da Informação (TI).

Os contratos administrativos compõem o rol de meios e recursos com os quais o Centro de Informática conta para adequadamente cumprir suas atribuições. Assim, o Cenin promove ações que visam a buscar na iniciativa privada bens, insumos, recursos e serviços necessários à operação de sua infraestrutura de TI, bem como à execução de projetos diversos sob sua responsabilidade e à prestação de serviços a seus clientes e usuários. A correta execução dos contratos, desse modo, é condição basilar para a satisfatória operação do próprio Centro de Informática, o que lhe impõe rigoroso acompanhamento.

Outro aspecto relevante é que tanto a Lei [2, 3, 8] quanto a boa prática técnica [5, 6, 7] impõem à Administração um rigoroso acompanhamento e controle da execução dos contratos. Assim, diversos atos formais [2, 3, 4, 8] devem ser observados e executados pelos servidores lotados no Centro de Informática, quando estes houverem sido designados como fiscais dos contratos.

A execução dos atos, alguns deles simples outros nem tanto, inerentes à fiscalização de contratos, concomitantemente com a execução de tarefas que são rotineiramente encontradas em qualquer organização intensiva em *software*, dá espaço para a perda de eficácia. Esta perda decorre, entre outros, de interpretações particulares das exigências contratuais ou das disposições legais, da grande variação no conhecimento e na experiência particular de cada um dos fiscais, do grau de maturidade do serviço ou tecnologia que formam o objeto do contrato ou da complexidade e do volume envolvidos na contratação.

A perfeita execução contratual, contudo, impõe expurgar quaisquer desses elementos que dão espaço à perda de eficácia. Assim, o desenvolvimento de um processo que contemple os mecanismos capazes de mitigar os riscos, que apresente padrões, que dissemine conhecimentos e que oriente comportamentos e ações comprovadas pode aumentar as chances de se obter plena execução dos contratos do domínio da TI. O projeto que ora se relata foi concebido e realizado neste contexto.

2. Objetivos e Justificativa

A atual prática de fiscalização dos contratos administrativos executados pelo Centro de Informática não conta com um conjunto explícito de procedimentos, modelos e rotinas padronizadas. Desse modo, a eficácia do acompanhamento e da fiscalização é diretamente proporcional ao conhecimento e à experiência da equipe de servidores designada para a sua fiscalização.

As determinações legais [2, 3, 4, 8], por seu turno, apresentam comandos às vezes abstratos, às vezes incompletos e, em alguns casos, até contraditórios. Essa configuração normativa também colabora para que a fiscalização e o acompanhamento da execução de contratos se torne tarefa sujeita a falhas, muitas delas decorrentes de interpretações conflitantes ou equivocadas dos comandos legais.

As recomendações e as determinações do Controle Externo [9, 10] apresentam elementos que igualmente modelam a fiscalização dos contratos. Sua aplicação sistêmica, contudo, impõe aos responsáveis pela

fiscalização realizar contínua busca pelas novas postulações. Às organizações sujeitas ao Controle também compete verificar se todas as recomendações e determinações publicadas estão sendo seguidas e implementadas a contento em todos os seus contratos.

Esse vasto conjunto de atributos é o que define o domínio da fiscalização de contratos administrativos e desnuda seus riscos; e nisso reside a gênese da concepção deste projeto, cujo objetivo é definir parâmetros para a fiscalização de contratos de TI, que consolide modelos, rotinas e procedimentos e que seja capaz de (I) facilitar o desempenho da atividade de fiscalização de contratos; (II) aumentar a eficiência, o controle e o acompanhamento da execução dos contratos; (III) equilibrar a distribuição das tarefas inerentes à fiscalização entre os fiscais; (IV) viabilizar o compartilhamento de experiências, a reutilização de conhecimentos e incrementar o aprendizado organizacional; (V) aumentar a garantia da plena e fiel execução dos contratos; e (VI) implementar mecanismo efetivo para comunicar inovações, novos procedimentos, ajustes, experiências, recomendações ou determinações legais.

3. Metodologia de Execução

A execução do projeto deu-se, preliminarmente, mediante a identificação das necessidades e dos requisitos desejados pela Casa e pela escolha de uma metodologia para modelagem do processo.

Desse modo, elaborou-se um questionário contemplando dezesseis itens que foi, então, utilizado na condução de entrevistas com servidores do Centro de Informática que, à época da execução do projeto, atuavam como fiscais ou que já haviam sido responsáveis pela fiscalização e acompanhamento de contratos.

Foram realizadas entrevistas com vinte e cinco pessoas, de diversos níveis hierárquicos e com distinta experiência no tema. Os dados coletados foram então consolidados e formaram o conjunto inicial de requisitos que deveriam ser atendidos pelo processo de fiscalização.

O conjunto inicial de requisitos indicou ser a metodologia *SPEM* [11, 12] adequada à modelagem do processo. Em decorrência, a equipe de execução do projeto realizou a modelagem do processo com o uso da ferramenta *Eclipse Process Framework Composer* (epfc) [13], pois essa solução possui duas características fundamentais: (I) a ferramenta implementa a metodologia *SPEM* e (II) cria automaticamente um *site* a partir da modelagem de processo nela realizada.

Concomitantemente com a modelagem do processo, realizou-se pesquisa na literatura do domínio da fiscalização de contratos administrativos [9, 10, 14, 15, 16, 17], cujos insumos também determinaram a criação da primeira versão do processo.

A modelagem inicial do processo resultou na criação da primeira versão do *site* do processo de fiscalização. Esse primeiro produto passou, então, por quatro ciclos iterativos de apresentação, coleta de opiniões e recomendações, avaliação das sugestões apresentadas e conseqüente aplicação das melhorias consideradas pertinentes ao processo. Cada ciclo foi marcado pela apresentação do processo a um público distinto, a saber: (I) a Direção do Centro de Informática, (II) a equipe da Coordenação de Planejamento e Gestão de TIC, (III) a Secretaria de Controle Interno, Departamento de Material e Patrimônio e a Diretoria Administrativa da Câmara dos Deputados e, por fim, (IV) o grupo inicial de entrevistados.

Os quatro ciclos produziram sessenta e quatro sugestões de melhoria. As consideradas pertinentes foram aplicadas; as julgadas impertinentes, as que versavam sobre objeto alheio ao escopo do projeto ou que já haviam sido cotejadas na modelagem prévia, foram descartadas. A aplicação das melhorias resultou, então, na segunda versão do *site* do processo de fiscalização e no marco de encerramento da segunda etapa do projeto.

A terceira etapa iniciou-se com a publicação, na *intranet* do Centro de Informática, da segunda versão do *site* do processo, a cujo endereço deu-se ampla publicidade, e com sua apresentação, realizada no dia 20 de agosto de 2010 no Plenário 15 do Anexo II da Casa. A terceira etapa prolongou-se até 29 outubro de 2010, quando foram recebidas vinte e sete novas sugestões, que foram igualmente avaliadas e aquelas tidas como apropriadas foram aplicadas ao processo.

Em 11 de novembro de 2010 foi estabelecida a quarta versão do *site* do processo, também obtida mediante sua modelagem com a ferramenta *epfc*. Esta última versão marcou o encerramento da modelagem e seu resultado foi formalmente apresentado mediante os seguintes processos administrativos: o de n.135.399/2010, encaminhado à Assessoria Técnica da Diretoria-Geral da Casa; o de n. 135.408/2010, encaminhado à sua Secretaria de Controle Interno; e o de n. 135.405/2010, este último direcionado à Diretoria Administrativa.

A quinta e última etapa do projeto foi a de elaboração do material de treinamento a ser empregado na capacitação do corpo técnico do Centro de Informática. Assim, a partir da quarta versão do *site* do processo, foi criado um roteiro para a capacitação, além de documentos de apoio à realização do treinamento, formado por um conjunto de apresentações, uma lista de exercícios a serem praticados e as respectivas respostas.

O material elaborado foi submetido a uma sessão de simulação do treinamento, na qual a equipe do projeto ministrou o curso e a equipe da Coordenação de Planejamento e Gestão de TIC atuou como discente. A simulação permitiu dimensionar o tempo necessário à realização do treinamento padrão, além de terem sido colhidas sugestões e aperfeiçoamentos tanto para sua condução quanto para o material de apoio.

O dimensionamento do tempo necessário à capacitação no processo de fiscalização e a elaboração do material de apoio permitiram à equipe do projeto planejar a capacitação do corpo técnico do Centro de Informática, caracterizando, portanto, o alcance do segundo e último produto esperado do projeto “Elaboração de Processo de Fiscalização de Contratos do Domínio da Tecnologia da Informação”.

4. Resultados Obtidos

O Centro de Informática obteve quatro resultados significativos com a execução deste projeto. O primeiro, o próprio Processo de Fiscalização de Contratos de TI, formou um robusto conjunto de tarefas, orientações, referências, atividades, papéis, procedimentos e modelos, com forte amparo e embasada referência na legislação, facilmente manutenível e adaptável. O processo elaborado, portanto, é capaz de conferir maior segurança aos seus executores, os fiscais de contratos.

O segundo resultado foi o seu *site*, obtido como consequência da decisão tomada pela equipe do projeto pelo uso da ferramenta *epfc* para a realização da modelagem do processo. Esse produto permite que os fiscais façam rápida e precisa referência às atividades, normas, procedimentos e modelos previstos pelo processo, pois consolida em um único ponto a busca de informações sobre a fiscalização de contratos. Mais possibilita que as mudanças ocorridas na legislação ou na prática da fiscalização sejam comunicadas com larga amplitude e com grande agilidade, potencializando-se, deste modo, seu tempestivo emprego.

O terceiro resultado alcançado mediante a execução deste projeto foi a elaboração do material de treinamento, o que permitirá a capacitação de todo corpo técnico do Centro de Informática em seu processo de fiscalização de contratos. O mesmo material poderá, no futuro, ser utilizado pelos demais órgãos que compõem a estrutura administrativa da Câmara dos Deputados na capacitação de seu próprio pessoal, uma vez que o processo de fiscalização de contratos de TI possui instruções geais que são também aplicáveis à fiscalização de contratos administrativos de outros domínios.

Por fim, a execução deste projeto apresentou um resultado colateral significativo: a capacitação de sua equipe de execução da modelagem formal de processos e na práxis da fiscalização de contratos administrativos.

É notório que a capacitação de pessoas é um desafio perseguido pelas organizações, pois possuir quadros com robustez técnica é diferencial positivo. Assim, a execução deste projeto habilitou o Centro de Informática a melhor desempenhar suas atribuições, pois formou equipe de especialistas na fiscalização de contratos administrativos e, em consequência, viabilizou a instituição do Grupo de Apoio à Fiscalização de Contratos (GAFIC), necessidade verificada durante a identificação dos requisitos.

A previsão de ente no Centro de Informática que esteja apto a esclarecer dúvidas e a auxiliar os fiscais na execução de tarefas complexas certamente colabora para o sucesso do processo de fiscalização de contratos. Assim, encerrado o projeto, o Centro de Informática já conta com a equipe necessária à plena operacionalização do GAFIC.

4.1. O Processo de Fiscalização de Contratos

O Processo de Fiscalização de Contratos de TI (*fiscon*) aborda três temáticas distintas: (I) atribuir fiscalização do contrato a servidor efetivo, (II) fiscalizar devidamente o contrato e (III) acompanhar e apoiar as fiscalizações dos contratos em execução. Deste modo, modelou-se o processo de forma a contemplar todos esses domínios, o que resultou em formar o *fiscon* a partir de três subprocessos especializados: “Atribuir Fiscalização”, “Fiscalizar Contrato” e “Acompanhar Fiscalizações”.

O primeiro subprocessos, “Atribuir Fiscalização”, é responsável pela seleção dos servidores e sua respectiva designação. O subprocesso “Fiscalizar Contrato” contém as atividades e as tarefas que o fiscal deve realizar durante a fiscalização do contrato, nele incluindo-as etapas de planejamento, de execução e de encerramento da fiscalização. O subprocesso “Acompanhar Fiscalizações” é responsável por apoiar os fiscais na realização das tarefas inerentes à fiscalização, por manter o processo de fiscalização atualizado e por coletar dados e apresentar informações gerenciais sobre os contratos em execução.

Todos os três subprocessos são formados por atividades. Cada atividade, por seu turno, reúne determinado conjunto de tarefas que, quando executadas, permite o alcance dos objetivos pretendidos. A execução das atividades e tarefas é atribuída a papéis específicos e é apoiada por modelos para elaboração de artefatos, guias para sua execução, listas de verificação, além de outros insumos de apoio.

5. Aplicabilidade dos Resultados

Os dois produtos elaborados pelo projeto, a saber, o processo de fiscalização de contratos de TI e o respectivo material de treinamento, podem ser adaptados e reutilizados pelos órgãos da Administração Pública para o acompanhamento de seus próprios contratos com grande facilidade, pois o processo segue as determinações da legislação, as recomendações do Controle e as boas práticas já identificadas.

A facilidade na adaptação do processo também é garantida em decorrência de sua modelagem, que se deu mediante o uso de uma metodologia formal, de grande divulgação, amplo emprego e fácil domínio. Ademais, a utilização da ferramenta *epfc* permite que todos os elementos que compõem o processo sejam parametrizados

à exaustão e, assim, reutilizados em processos adaptados de fiscalização às necessidades da entidade adotante.

As entidades do segmento privado também podem aplicar os resultados deste projeto e, assim, parcialmente atender às recomendações de duas das normas técnicas basilares para a indústria da TI, discriminadas nas tarefas “Monitorar contrato” e “Aceitar o produto ou serviço” do “Processo de aquisição” [6] e “Acordo Contratual”, “Monitoração do acordo”, “Aceite do adquirente” e “Fechamento” do “Processo de aquisição” [5].

6. Características Inovadoras

As pesquisas conduzidas durante a execução do projeto não identificaram solução semelhante à ora elaborada.

A implementação do processo também apresenta inovações na dinâmica da fiscalização de contratos, especialmente por adotar a abordagem preconizada pelo domínio do gerenciamento de projetos uma vez que a fiscalização é tomada como se um projeto fosse, *i.e.*, o processo prevê que seja inicialmente planejada a fiscalização para somente então ser executada, com pleno acompanhamento do contrato. Após a execução, o “projeto de fiscalização” é formalmente encerrado.

A última etapa da fiscalização prevista pelo processo também incorpora elementos inovadores, pois determina que o projeto de fiscalização seja encerrado, com o conseqüente arquivamento dos registros produzidos, a descrição dos resultados obtidos, a listagem das ações administrativas necessárias ao encerramento do contrato e o relato da fiscalização. Estes elementos são utilizados para realimentar o processo de fiscalização, tornando-o mais eficaz mediante a incorporação de experiências decorrentes de sua própria execução.

A organização que o adota é também beneficiada na governança dos seus contratos, pois o processo prevê tarefa exclusivamente concebida para a coleta de indicadores, que poderão ser utilizados para a avaliação qualitativa e quantitativa dos contratos executados e acompanhados mediante o processo de fiscalização.

7. Conclusão e Perspectivas Futuras

O Projeto 2.03 teve como meta principal a elaboração de processo para facilitar e potencializar a fiscalização dos contratos do domínio da TI firmados pela Câmara dos Deputados e executados pelo seu Centro de Informática. Esta meta foi plenamente alcançada, visto que foi elaborado o Processo de Fiscalização de Contratos de TI.

Os resultados obtidos, entretanto, foram superiores aos inicialmente esperados, pois além de se obter todos os produtos exigidos do projeto, foi possível à sua equipe realizar trabalho sinérgico com as diversas pessoas envolvidas, especialmente as que foram designadas como fiscais. Criou-se, assim, vínculo de confiança favorável ao compartilhamento de informações, o que permite maiores ganhos na fiscalização e acompanhamento de contratos.

Não se esgotou, contudo, o domínio do problema, pois novas ações podem ser concebidas e iniciadas para dotar os fiscais de contratos de melhores instrumentos.

Neste sentido, é possível, por exemplo, a criação de sistema de informação próprio que automatize as tarefas previstas pelo processo de fiscalização. Para tal empreitada, o *fiscon* pode ser tomado como especificação inicial dos requisitos do sistema de informação a ser desenvolvido.

Também figura entre as perspectivas futuras a possibilidade de a Câmara dos Deputados publicar e compartilhar os produtos criados pelo projeto. Caso realizada, esta iniciativa permitirá, inclusive, a criação de uma comunidade de usuários em torno do processo de fiscalização de contratos, o que poderá resultar em seu próprio benefício, especialmente em conseqüência do compartilhamento de experiências e de melhorias.

Para realizá-la, a Casa poderá utilizar o Portal do Software Público Brasileiro, ação do Ministério do Planejamento, Orçamento e Gestão que visa ao compartilhamento de soluções e produtos do domínio da Tecnologia da Informação.

8. Referências Bibliográficas

[1] BRASIL. Congresso Nacional. Câmara dos Deputados. **Resolução da Câmara dos Deputados n. 16**, de 21 de janeiro de 1997. *Cria o Centro de Informática e dá outras providências*. Diário da Câmara dos Deputados, Brasília, DF, 22 jan. 1997. p. 2019

[2] _____. **Lei 8666, de 21 de junho de 1993**. *Regulamenta o art. 37, inciso XXI, da Constituição Federal, institui normas para licitações e contratos da Administração Pública e dá outras providências*. Diário Oficial da União, Seção 1, Brasília, DF, 22 jun. 1993. P. 8269

[3] _____. Congresso Nacional. Câmara dos Deputados. **Portaria n. 119, de 11 de novembro de 2006**. *Regulamenta a atividade de fiscal de contrato*. Boletim Administrativo da Câmara dos Deputados, Brasília, DF, 14 set. 2006

- [4] _____. Congresso Nacional. Câmara dos Deputados. **Portaria n. 179, de 22 de novembro de 2005.** *Disciplina a rotina e controle dos saldos de empenho.*
- [5] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR ISO/IEC 12207:** Engenharia de sistemas e de *software*: processos de ciclo de vida de *software*. Rio de Janeiro, 2009
- [6] _____. **ABNT NBR ISO/IEC 15288:** Engenharia de sistemas e de *software*: processos de ciclo de vida de sistema. Rio de Janeiro, 2009
- [7] BRASIL. **Lei 4.150, de 21 de novembro de 1962.** *Institui o regime obrigatório de preparo e observância das normas técnicas nos contratos de obras e compras do serviço público de execução direta, concedida, autárquica ou de economia mista, através da associação Brasileira de Normas Técnicas.* Diário Oficial da União, Seção 1, Brasília, DF, 11 jan. 1963. p. 345
- [8] BRASIL. Congresso Nacional. Câmara dos Deputados. **Ato da Mesa n. 80, de 7 de junho de 2001.** *Aprova o Regulamento dos Procedimentos Licitatórios da Câmara dos Deputados.* Diário da Câmara dos Deputados, Suplemento, Brasília, DF, 8 jun. 2001. p. 36
- [9] _____. Tribunal de Contas da União. **Licitações & Contratos: orientações e jurisprudência do TCU.** 4. ed. Brasília, DF: TCU, Secretaria-Geral da Presidência : Senado Federal, Secretaria Especial de Editoração e Publicações, 2010. 910p.
- [10] _____. **Licitações & Contratos Administrativos em Tecnologia da Informação:** jurisprudência do TCU. 1. ed. Brasília, DF: TCU, Secretaria de Material, Patrimônio e Comunicação Administrativa, 2007. 219p
- [11] IBM Training. **Basic Method Authoring with IBM Rational Method Composer v7.2.** San Jose, CA, USA: IBM Rational Software, Santa Teresa Lab.2008.
- [12] SOFTWARE & SYSTEMS PROCESS ENGINEERING METAMODEL SPECIFICATION. Disponível em <<http://www.omg.org/spec/SPEM/>>. Acesso em 3 maio 2010.
- [13] ECLIPSE PROCESS FRAMEWORK PROJECT. Disponível em <<http://www.eclipse.org/epf/>>. Acesso em 3 maio 2010.
- [14] ALVES, Léo da Silva. **Prática de gestão e fiscalização de contratos públicos.** 1. ed. Brasília: Brasília Jurídica, 2005. 259p.
- [15] BRASIL. Congresso Nacional. Senado Federal. **Gestão de Contratos:** orientações básicas ao Gestor. 1. ed. Brasília, DF: Senado Federal, Secretaria Especial de Editoração e Publicações, 2008. 126p.
- [16] _____. Congresso Nacional. Câmara dos Deputados. **Manual do fiscal de contratos.** Minuta. Brasília, DF: Câmara dos Deputados, Diretoria Administrativa, 2010.
- [17] _____. Ministério da Educação. Centro Federal de Educação Tecnológica de São Paulo. **Manual do fiscal de contratos.** São Paulo, SP, 2007.

Um Estudo Preliminar sobre Tipos de Personalidade Adequados a Inspectores de Software

Tayana Contea, Nayane Maiaa, Anna Beatriz Marquesa¹, Emília Mendesb¹, Guilherme Horta Travassosb¹

a Grupo de Usabilidade e Engenharia de Software
Universidade Federal do Amazonas (UFAM) - Manaus, AM - Brasil
b PESC - COPPE/UFRJ, Rio de Janeiro, RJ, Brasil

{tayana,, nayane.maia, anna.beatriz }@dcc.ufam.edu.br, {emilia, ght}@cos.ufrj.br

Abstract. *Usability is one of the most relevant factors for the acceptance of Web applications. Different inspection techniques have been proposed aiming at detecting usability defects and to increase the quality of Web applications. However, studies have indicated great variation in the results even when using inspectors with similar experience. It is possible that other non technical factors have influenced the results of these usability inspections. Therefore, this paper aims to investigate the influence that different personality profiles (based on Myers-Briggs Type Indicator) may have upon the performance of usability inspectors. These results, when compared to a previous study that investigated the personality profiles of code inspectors, suggest that the allocation of inspectors based on their personality profiles seems to increase the number of defects detected in usability inspections.*

Resumo. *A usabilidade é um dos fatores mais relevantes para a aceitação de aplicações web. Diferentes técnicas de inspeção têm sido propostas com o objetivo de detectar os defeitos de usabilidade e aumentar a qualidade das aplicações web. No entanto, estudos têm indicado uma grande variação nos resultados, mesmo quando usando inspetores com experiência semelhante. É possível que fatores não-técnicos influenciem os resultados dessas inspeções de usabilidade. Este trabalho tem como objetivo investigar a influência que os diferentes perfis de personalidade (com base no Indicador MBTI - Myers- Briggs Type Indicator) podem ter sobre o desempenho dos inspetores de usabilidade. Estes resultados, quando comparados a um estudo anterior que investigou o perfil de personalidade dos inspetores de código, indicam que a atribuição de inspetores com base em seus perfis de personalidade parece aumentar o número de defeitos detectados em inspeções de usabilidade.*

1. Introdução

A usabilidade pode ser apontada como um dos fatores que podem influenciar a melhoria na qualidade de um *software*, sendo definida como a capacidade que o *software* tem de ser entendido, usado e aprendido, e também sua capacidade de agradar ao usuário, quando utilizado sob condições específicas [1]¹. Neste contexto, a usabilidade é de grande importância para a utilização, entendimento, aprendizado e qualidade de um *software*.

No caso de aplicações *Web*, sua importância é ainda maior, pois tais aplicações são interativas, centradas no usuário e baseadas em hipermídia, onde a interface com o usuário desempenha um papel central [2], o que motivou várias pesquisas sobre técnicas de inspeção de usabilidade para aplicações *Web* como a técnica WDP proposta por [3], WDP-RT descrita em [4,5] e WE-QT proposta em [6]. Em estudos experimentais que buscavam avaliar a viabilidade dessas técnicas de inspeção de usabilidade, constataram-se diferentes níveis de resultados em termos de custo-eficiência (média de defeitos/hora) mesmo com inspetores com mesmos níveis de experiência (nenhuma ou baixa), o que nos leva a considerar que existem outros fatores que influenciam na habilidade de inspetores de usabilidade [5].

A influência do time de desenvolvimento é um fator tem sido cada vez mais estudado na Engenharia de *Software* [7]. Recentemente os enfoques essencialmente funcionais têm sido complementados com análises da influência do tipo de personalidade dos indivíduos no desempenho do time [7]. Esses novos estudos baseiam-se em obras vastamente difundidas por estudiosos de psicologia de acordo com os aspectos a serem estudados, tais como: teorias dos tipos de personalidade, teoria de papéis em time ou estilos cognitivos, além de indicadores de perfis de personalidade.

Segundo Boehm *apud* [7], tais pesquisas têm sugerido que as questões pessoais representam o maior potencial para melhorar a qualidade do *software* e do processo de desenvolvimento.

Em [8], é descrito um estudo que visa identificar perfis de personalidade que podem influenciar na habilidade inspetores de código. Faz-se necessário observar se estes fatores são os mesmos que influenciam na habilidade de inspetores de usabilidade de uma aplicação. Para verificar tal hipótese, realizou-se um estudo a fim de identificar quais perfis de personalidade são mais adequados para o papel de inspetor de usabilidade.

¹ 1 Pesquisadores não inicialmente cadastrados no projeto que colaboraram no andamento da pesquisa

Este estudo teve como objetivo analisar e aplicar teorias de perfis de personalidade a fim de obter a relação entre a característica psicológica de um inspetor de usabilidade e seu desempenho em inspeções de usabilidade.

O restante deste artigo está estruturado da seguinte forma. A seção 2 expõe a justificativa e os objetivos deste projeto. A seção 3 apresenta a metodologia utilizada para sua execução. A seção 4 discute os resultados obtidos. A seção 5 apresenta a aplicabilidade dos resultados. A seção 6 exibe as características inovadoras desta pesquisa. Por fim, a seção 7 conclui o artigo.

2. Objetivos e Justificativa

A variedade de papéis da área de desenvolvimento tais como analistas, projetistas, testadores, inspetores, dentre outros, demanda diferentes tipos de personalidade e comportamento. Estudos mostram que as habilidades requeridas por um determinado papel podem ter relação com o estilo de personalidade e/ou comportamento de cada indivíduo. Vários estudos têm sido desenvolvidos buscando identificar características pessoais que podem influenciar o desempenho no trabalho. Estas características podem ser identificadas através de vários testes de personalidade.

Um dos indicadores de tipo de personalidade mais utilizados em diversos estudos é o MBTI® (Myers-Briggs Type Indicator) [9]. O MBTI® é um instrumento de identificação de características pessoais, que possibilita identificar as características, pontos fortes e aspectos de desenvolvimento [9]. Os perfis de personalidade propostos por Myers-Briggs são definidos a partir de quatro grandes dimensões bipolares da personalidade: (1) *Extraversion/Introversion* (Extroversão/Introversão), (2) *Sensing/Intuition* (Sensação/Intuição), (3) *Thinking/Feeling* (Pensamento/Sentimento) e (4) *Judging/Perceiving* (Julgamento/Percepção).

Os tipos psicológicos citados por Myers-Briggs são classificados nestas quatro dimensões, cuja escala de preferência varia entre dois extremos. O MBTI® pressupõe que os indivíduos utilizam vários processos cognitivos, mas tendem a ter uma preferência entre as dicotomias citadas. O tipo de personalidade individual é identificado através de um código de quatro letras, cada uma representando o pólo dominante na dimensão (E ou I, S ou N, T ou F, J ou P). Em teoria, cada um dos dezesseis diferentes tipos de personalidade medidas pelo MBTI pode ser visto como um conjunto de padrões que aponta como o indivíduo se comporta.

Estudos mostram que as habilidades requeridas por um determinado papel podem ter relação com o estilo de personalidade e/ou comportamento de cada indivíduo.

Essa afirmativa pode ser verificada a partir de diversas publicações, algumas das quais estão apresentadas na Tabela 1.

Tabela 1 - Estudos sobre a Influência dos Tipos de Personalidade em Desenvolvimento de Software

<i>Publicação/ Autores</i>	<i>Foco</i>	<i>Indicador Utilizado</i>
FERREIRA & SILVA [14]	Influência dos perfis psicológicos na utilização de processo de software	MBTI e Teoria de Papéis em Times
GORLA & LAM [15]	Tipos de personalidade em diferentes papéis dos times de software	MBTI
MEIRA & SILVA [7]	Perfis psicológicos mais adequados aos profissionais responsáveis por Garantia da Qualidade	Teoria de Papéis em Times
CUNHA & GREATHEAD [8]	Perfil psicológico de revisores de código	MBTI

Esta pesquisa se propõe a averiguar se há relacionamento entre habilidades e perfis comportamentais dos profissionais que indique maior eficiência (desempenho) no papel de inspetor de usabilidade, a exemplo da pesquisa sobre a influência do perfil de personalidade para o papel de inspetor de código [8] utilizando o Indicador de Tipos de Personalidade de Myers-Briggs (MBTI®).

3. Metodologia de execução

Esta pesquisa foi realizada em três fases, sendo elas: Planejamento, Execução e Análise dos Resultados, conforme ilustrado na Figura 4.

A fase do Planejamento consistiu nas atividades de (1) caracterização e (2) planejamento do estudo. Na atividade (1) foi realizada uma revisão da literatura técnica com o objetivo de coletar dados sobre teorias e tipos de personalidade e características requeridas por um inspetor de usabilidade. Também foram revisados trabalhos que utilizaram essas teorias e modelos, a fim de verificar qual melhor se aplicaria ao problema em questão. Após a revisão, determinou-se que este estudo iria utilizar o Indicador MBTI®, em virtude de ser amplamente utilizado em estudos cujo intuito seja relacionar a característica do indivíduo com as habilidades requeridas em determinado papel.

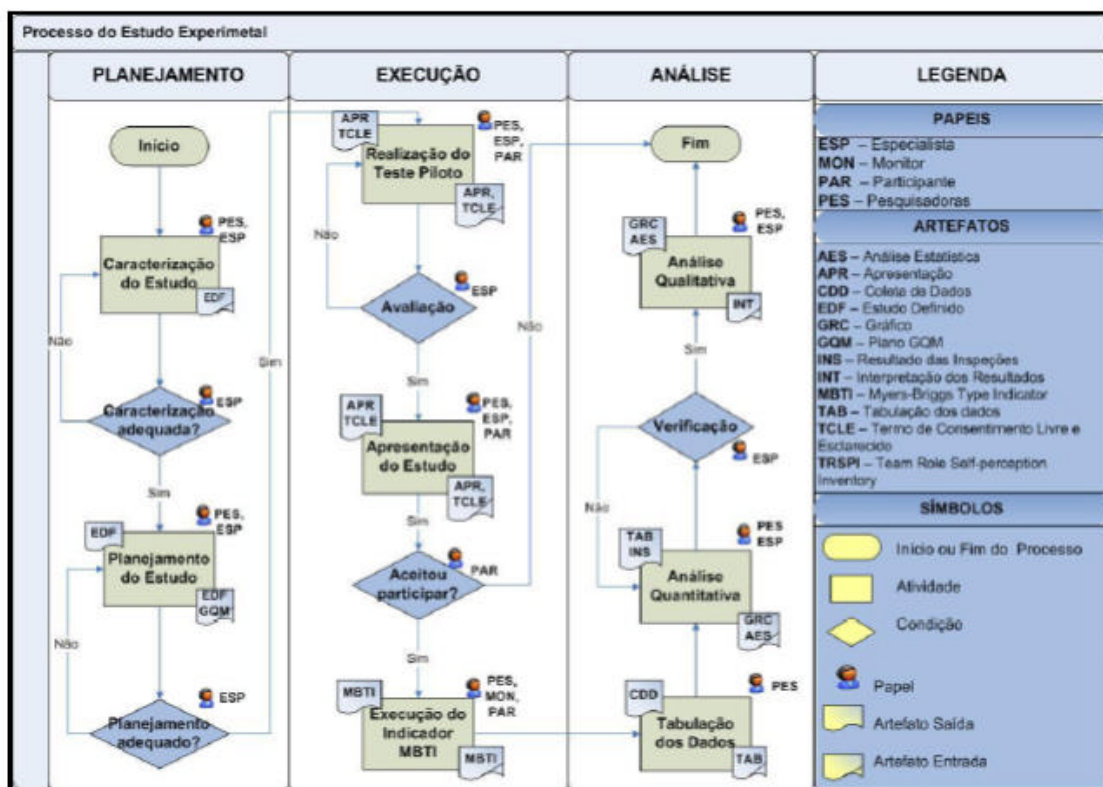


Figura 1 - Fases e Atividades da Pesquisa

A fase de Execução foi realizada em três etapas: (1) estudo piloto, (2) apresentação do estudo e (3) execução do indicador de personalidade MBTI®. A atividade (1) foi realizada com a presença de um especialista e de quatro participantes com experiência em inspeção de usabilidade. Nesta etapa foram apresentadas as teorias e descrição dos perfis psicológicos, tendo como artefatos o Termo de Consentimento Livre e esclarecido (TCLE) e as instruções para a realização do indicador de personalidade. A atividade (2) foi realizada após a verificação e validação dos artefatos e consistiu na apresentação das teorias e descrição de perfis de personalidade aos participantes do estudo. Assim, foi ministrado um seminário direcionado aos alunos de graduação e pós-graduação do Departamento de Ciência da Computação da Universidade Federal do Amazonas, matriculados nas disciplinas de Engenharia de Software e IHC. Ao término do seminário os participantes receberam o TCLE, onde eram convidados a participar da atividade (3) do estudo. Então, os participantes foram instruídos a realizar o indicador online de perfil de personalidade MBTI® e orientados a enviar o resultado via email às pesquisadoras. Este indicador de personalidade está disponível no endereço <http://www.inspiira.org>, e é baseado no Indicador de Tipos de Personalidade de Myers-Briggs (MBTI®). Após essa atividade deu-se início a terceira e última fase do estudo, a análise dos resultados.

Na fase de Análise foram selecionados os resultados dos alunos que haviam realizado as inspeções de usabilidade, descritas em [4], totalizando 27 (vinte e sete) participantes. Deste modo, a análise dos resultados abrange as atividades de (1) tabulação dos dados, (2) análise quantitativa e (3) análise qualitativa. A atividade (1) consistiu na coleta dos dados enviados pelos participantes com o resultado do perfil de personalidade. Nesta atividade foram geradas planilhas com a tabulação dos dados por participante, incluindo os resultados obtidos nas inspeções de usabilidade. Na atividade (2) realizou-se uma análise quantitativa dos indicadores de perfis de personalidade (MBTI®), verificando-se qual era o perfil de personalidade mais presente. Além disso, foram analisadas as dimensões de cada perfil, de forma individual e por temperamentos.

Esses valores quantitativos, após a verificação por um especialista, foram utilizados na atividade (3), que consistiu em uma análise para comparar os perfis de cada participante com seus resultados na atividade de detecção de defeitos em inspeções de usabilidade.

O objetivo foi averiguar a relação entre as características pessoais e o resultado da inspeção individual.

4. Resultados obtidos

Para a análise dos resultados da pesquisa foram considerados os dados referentes aos resultados das inspeções após a avaliação dos defeitos relatados por grupos de especialistas em usabilidade e no domínio do problema. Desta maneira, buscou-se garantir que apenas os defeitos reais apontados por cada inspetor seriam contabilizados.

Para avaliar o se o tipo de personalidade tinha alguma relação com o desempenho dos participantes na inspeção de usabilidade inicialmente foram calculadas as médias de defeitos encontrados em função do tempo para cada dimensão de personalidade. As médias de desempenho por dimensão do MBTI são apresentadas na Tabela 2.

Tabela 2 - Médias Defeito/hora dos inspetores por Dimensão MBTI

<i>Dimensão</i>	<i>I</i>	<i>E</i>	<i>S</i>	<i>N</i>	<i>T</i>	<i>F</i>	<i>J</i>	<i>P</i>
<i>Média Def/hora</i>	8,56	7,66	9,42	5,22	8,10	7,79	7,67	9,35
<i>Desvio Padrão</i>	4,20	3,86	3,75	2,79	3,46	5,06	4,13	4,60

Tabela 3 - Mediana (Def./Hora) por Dimensões MBTI

<i>Dimensão</i>	<i>I</i>	<i>E</i>	<i>S</i>	<i>N</i>	<i>T</i>	<i>F</i>	<i>J</i>	<i>P</i>
<i>Mediana</i>	7,74	7,7	9,37	4,16	8,4	7,26	5,60	8,52

Para observar os resultados apresentados utilizaremos o percentual obtido pela diferença entre as Médias Def/hora (Tabela 2) entre duas dimensões dividido pela média das medianas em Def/hora (Tabela 3) das mesmas dimensões. Esta medida tem a intenção de sugerir, numericamente, a diferença entre os pólos de cada dimensão, conforme a Tabela 3. Assim, percebe-se que existe uma pequena diferença nos resultados obtidos pelas dimensões Emotivos (F) x Pensadores (T) (3,9%) que a princípio não afetaria significativamente a organização das equipes de inspeção, ou seja, o desempenho seria equivalente. As dimensões Extrovertidos (E) x Introversos (I) (11,1%), apesar desta variação, ainda seria aceitável principalmente considerando a variância dos grupos. Por outro lado, as dimensões Julgadores (J) X Perceptivos (P) (19,74%) e Sensoriais (S) X Intuitivos (N) (57,38%) apresentam uma variação aumentada no desempenho, resultando em possíveis riscos para se organizar uma equipe de inspeção, ou seja, grupos organizados com estes perfis podem afetar a previsibilidade de comportamento dos times de inspeção. A partir disso, decidiu-se avaliar tais dimensões em virtude da variação observada. Como houve uma diferença na população das amostras não foi realizado um teste estatístico e sim uma análise a partir de boxplots, utilizando-se a ferramenta SPSS Data Editor.

A Figura 2 mostra os boxplots comparando o desempenho por Dimensões MBTI (dimensões Sensoriais x Intuitivos), em virtude da diferença apresentada na Tabela 5. Ao se analisar a Figura 2, nota-se que a mediana das amostras da dimensão dos Sensoriais é significativamente mais alta que a mediana da dimensão dos Intuitivos.

Assim, percebe-se que o grupo dos Sensoriais obteve melhores resultados em detecção de erros que o grupo dos Intuitivos ao inspecionar a usabilidade de uma aplicação Web.

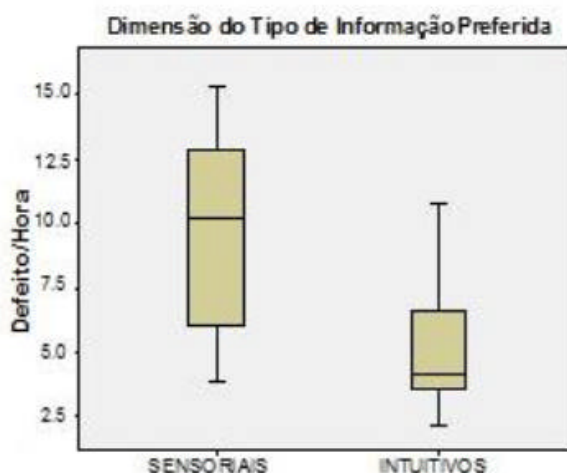


Figura 2 - Boxplots comparando o desempenho em detecção de erros de usabilidade por Dimensão: Tipo de Informação Preferida

De acordo com a definição de Jung *apud* [16], a dimensão Sensação-iNtuição relaciona-se com o processo de percepção, por meio do qual as informações são adquiridas, processadas e interpretadas. Suas principais características são apresentadas na Tabela 4.

Tabela 4 - Características de comportamento da dimensão S x N [16]

Sensação (S)	Caracterizado pela <i>atenção aos detalhes</i> . Os tipos sensoriais sentem-se confortáveis com <i>fatos concretos</i> que possam ser analisados em pequenas partes.
Intuição (N)	Caracterizado pela <i>atenção ao todo</i> e às informações de natureza abstrata. Os tipos intuitivos sentem-se à vontade com a visão de conjunto, procuram relações entre conceitos distintos e trabalham bem com hipóteses ou <i>idéias abstratas</i> .

Ao se comparar os resultados obtidos entre as dimensões S x N, onde os Sensoriais apresentaram o maior desempenho, pode-se verificar que uma característica que indica influência na diferença dos resultados dos Sensoriais é a atenção aos detalhes e fatos concretos, enquanto que os Intuitivos demonstram interesse pelo todo e por idéias abstratas. Fazendo uma analogia ao objeto de inspeção em cada estudo, é possível inferir que a interface é um fato concreto, enquanto o código é artefato abstrato. Nossa hipótese é de que isto seja o fator que explique a tendência de Sensoriais detectarem maior número de erros de usabilidade que os Intuitivos. A Figura 3 apresenta um esquema conceitual para essa hipótese.



Figura 3 - Hipótese do fator de influência para a inspeção de usabilidade

4.1. Artigo publicado

O artigo “Estudo sobre a Influência do Tipo de Personalidade do Inspetor no Desempenho de Inspeções de Usabilidade” (autores: Tayana Conte, Nayane Maia, Anna Beatriz Marques, Emília Mendes e Guilherme Horta Travassos) foi aceito para publicação e apresentação XIV Ibero-American Conference on Software Engineering (CIBSE 2011), a ser realizado no período de 27/04/2011 a 29/04/2011 no campus da PUC-Rio, Rio de Janeiro, Brasil.

5. Aplicabilidade dos resultados: Benefícios para a Indústria de Desenvolvimento de Software

Com os resultados desta pesquisa, pode-se perceber que existe um indício que inspeção de usabilidade requer habilidades e características diferentes que revisão de código, dado que os inspetores que apresentaram melhor desempenho em uma inspeção de usabilidade de uma aplicação foram os Sensoriais, enquanto que o estudo sobre perfis de revisores de código [8] indicou que os Intuitivos apresentaram melhor desempenho.

As inspeções melhoram a produtividade uma vez que os defeitos são encontrados quando são mais fáceis e mais baratos para corrigir. Esta pesquisa procurou verificar quais características de indivíduo o leva a se adequar melhor ao papel de inspetor de usabilidade, apresentando melhor rendimento, maximizando assim a produtividade das inspeções para avaliação de usabilidade.

6. Características inovadoras

Este trabalho é parte inicial de uma pesquisa que tem como intuito abordar os diversos fatores possam influenciar no desempenho de diferentes papéis em projetos de software.

Em função da subjetividade em relação à alocação adequada dos papéis e sua influência no desempenho do indivíduo, destaca-se o desafio, superado em função da importância de estudar cada vez mais aspectos humanos dentro da engenharia de software, em prol do sucesso e dos interesses da organização como um todo.

Esta pesquisa executou tanto uma análise quantitativa para avaliar o desempenho na detecção de erros de usabilidade, quanto uma análise qualitativa para caracterizar o comportamento que exercia influência nos resultados obtidos pelos inspetores.

7. Conclusões e trabalhos futuros

Esta pesquisa foi conduzida com o objetivo de caracterizar a influência do tipo de personalidade do inspetor na etapa de detecção de erros a partir de resultados de uma inspeção de usabilidade, onde foi levantada a questão se haviam outros fatores, além da experiência do inspetor, que pudessem explicar o porquê da diferença de resultados obtidos.

A pesquisa apontou um indício que inspeção de usabilidade requer habilidades e características diferentes que revisão de código, dado que os inspetores que apresentaram melhor desempenho em uma inspeção de usabilidade de uma aplicação foram os Sensoriais, enquanto que o estudo sobre perfis de revisores de código [8] indicou que os Intuitivos apresentaram melhor desempenho. Há ainda diferentes fatores que precisam ser investigados, que combinados com o artefato sob inspeção, podem representar fatores de influência relevantes. Investigações adicionais precisam ser realizadas visando observar mais diretamente tais comportamentos.

Uma limitação relevante deste estudo está relacionada ao número heterogêneo de participantes por amostras dos diferentes temperamentos e perfis de personalidade apresentados. Contudo, como o objetivo do estudo limitava-se a caracterizar a existência de fatores de influência do perfil de personalidade no desempenho na detecção de erros, fatores estes que podem levar ao aumento da qualidade e, conseqüentemente, da produtividade nas inspeções. Ainda assim, esta limitação apresenta-se como uma oportunidade para realização de uma pesquisa quantitativa a fim de verificar a validade das suposições levantadas neste artigo, comparando-se amostras de tamanhos homogêneos para os temperamentos apresentados.

Agradecimentos

Os autores agradecem aos participantes dos estudos e aos pesquisadores do USESUFAM pelas suas contribuições e agradecem o apoio da CAPES e CNPq

Referências Bibliográficas

- [1] ISO/IEC 9126-1, International Organization for Standardization. "Information Technology – Software Product Quality. Part 1: Quality Model". 1999.
- [2] Olsina, L., Covelha, G., Rossi, G., (2006), "Web Quality". In: MENDES, E., MOSLEY, N. (eds), Web Engineering, Springer Verlag.
- [3] Conte T., Massolar, J., Mendes, E., Travassos, G. (2007). Web Usability Inspection Technique Based on Design Perspectives. SBES 2007, João Pessoa, PB, Brasil.
- [4] Gomes, M. et al., "WDP-RT: Uma técnica de leitura para inspeção de usabilidade de aplicações Web". ESELAW 2009.
- [5] Gomes, M. et al., "Evolução de uma Técnica de Avaliação de Usabilidade através de Estudos In Vitro e In Vivo" SBQS 2010.
- [6] Fernandes, P. S., Conte, T. (2010). "Uma técnica automatizada de perguntas e respostas para inspeção de usabilidade em aplicações de Web". Projeto aprovado no Ciclo 2010 do Programa Brasileiro de Qualidade e Produtividade - PBQP 2010.
- [7] Meira, A. F., Silva, F. Q. (2009). "Hababilities and Behavioural Profiles of SQA Professionals Related to Process Maturity Levels". Quality Assurance, 136-144. doi: 10.1109/SBES.2009.15.
- [8] Cunha, A. D. and Greathead, D. (2007). "Does personality matter?: an analysis of codereview ability." Communications of the ACM. New York, USA: Association for Computing Machinery, v. 50, n. 5, p. 109-112, Maio.
- [9] Myers, I. and Briggs, K. Myers-Briggs Type Indicator. Disponível em: <<http://www.myersbriggs.org/>>. Acesso em: 28 mai. 2010
- [10] Feldt, R., Angelis, L., Torkar, R., Samuelsson, M. (2010) "Links between the personalities, views and attitudes of software engineers". Information and Software Technology.
- [11] Jung, C. G. (1991). Tipos psicológicos. Obras Completas, vol. VI. Petrópolis: Vozes
- [12] Morales, S. A. (2004) "Relação entre competências e tipos psicológicos Junguianos nos Empreendedores" Tese (Doutorado) - Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Produção, 2004.
- [13] Keirsey, David (1998) "*Please Understand Me II*". Prometheus Nemesis Book Company.
- [14] Ferreira, A., Silva, F. Q. (2008). "Fatores humanos que influenciam a utilização de processos de software". SBQS 2008.

- [15] Gorla, N. and Lam, Y.W. (2004) "Who should work with whom? Building effective software project teams". Communication of the ACM, June, v. 47, n. 6.
- [16] Maximiano, A. C. A. (2008). "Administração de Projetos: como transformar ideias em resultados". 3ª. Ed. – São Paulo: Atlas.

Fatores Críticos de Sucesso em Programas de Melhoria de Processos de Software - Um Estudo Qualitativo

Davi Viana, Olavo Matos, Vinícius Secatti, Tayana Conte

Grupo de Usabilidade e Engenharia de Software , Departamento de Ciência da Computação
Universidade Federal do Amazonas (UFAM) – Manaus, AM - Brasil

{davi.viana,olavomatos,tayana}@dcc.ufam.edu.br

Abstract. *Several aspects can influence the Software Process Improvement (SPI) program. It is important to analyze the aspects to obtain a better understanding of its influence on the SPI program. This article discusses the main findings of a qualitative study conducted during the implementation of SPI. The aim is to deepening understanding about how various factors can affect an SPI program from view-point of collaborators.*

Resumo. *Diversos aspectos podem influenciar um Programa de Melhoria de Processo de Software (MPS). É importante analisar os aspectos buscando obter uma melhor compreensão sobre sua influência em um programa de MPS. Este artigo discute os principais resultados de um estudo qualitativo conduzido durante a implementação de um programa de MPS. O objetivo é aprofundar a compreensão sobre como os mais diversos aspectos podem influenciar um programa de MPS do ponto de vista dos colaboradores.*

1. Introdução

Organizações possuem processo de software para guiar o desenvolvimento dos produtos. Estes processos são compostos por uma sequência de passos ou etapas, cada uma das quais executando um conjunto de atividades bem definidas que conduzem a conclusão do projeto (JALOTE, 1997).

Para aumentar a capacidade das organizações no desenvolvimento de software, uma das abordagens mais amplamente utilizadas é a implementação de Melhoria de Processo de Software (MPS), porém há diversos fatores que podem influenciar o sucesso destas iniciativas (MONTONI e ROCHA, 2007).

Estudos que apresentam a influência de fatores sobre as iniciativas de implementação de melhoria de processos de software vêm sendo objeto de estudos nos últimos anos (MONTONI e ROCHA, 2010). O propósito desses estudos é obter um melhor entendimento sobre as questões que influenciam iniciativas de melhoria de processo de software, bem como suas interações, causas, efeitos e formas de tratamento. Essas questões são comumente denominadas de Fatores Críticos de Sucesso (FCS).

Embora haja um expressivo número de pesquisas para a identificação dos Fatores Críticos de Sucesso em MPS, segundo MATOS *et al.*, 2010, são menores os números de pesquisas que tem como foco entender como o colaborador da organização envolvida percebe o próprio programa MPS. Com a finalidade de aumentar o comprometimento dos envolvidos, é necessário entender quais os fatores críticos de sucesso são percebidos pelos colaboradores, pois sua percepção é uma valiosa fonte de informação, uma vez que o mesmo esteve presente na atividade analisada. As informações obtidas auxiliam na análise dos fatores relacionados a iniciativas de MPS.

O restante deste artigo está estruturado da seguinte forma. A seção 2 expõe os objetivos e a justificativa deste projeto. A seção 3 apresenta a metodologia utilizada para sua execução. A seção 4 discorre sobre os resultados alcançados. A seção 5 apresenta a aplicabilidade dos resultados. A seção 6 exibe as características inovadoras desta pesquisa. Por fim, a seção 7 conclui o artigo.

2. Objetivos e Justificativa

Diversos estudos têm sido conduzidos para verificar os diferentes tipos de fatores críticos de sucesso em programas de melhoria de processo. Por exemplo, Em BADDON e HALL (2003) relatam que experiências negativas anteriores ao programa de MPS e resistência organizacional relacionada às mudanças na forma de trabalhar já estabelecida, tem como consequência a baixa motivação dos colaboradores com o programa de melhoria que está sendo executado.

Segundo DYBA (2000), fatores como treinamento dos colaboradores da equipe e monitoramento das atividades são questões chave para o sucesso de um programa de MPS. Posteriormente, em outro estudo, DYBA (2003) descreve que o tamanho da organização é um fator que pode influenciar no sucesso da implantação de programa de MPS. Apesar dos diversos fatores encontrados na literatura, é necessária uma análise dos fatores que influenciaram o primeiro programa de MPS do Estado do Amazonas a fim de contribuir para futuras implementações de MPS no Estado.

O objetivo principal deste artigo é apresentar um estudo qualitativo a respeito dos fatores críticos de sucesso de um programa de MPS sob a perspectiva do colaborador. No presente trabalho, apresenta-se os resultados um estudo em três pequenas empresas de desenvolvimento de software pioneiras na implementação do nível G do modelo MPS.BR no Estado do Amazonas.

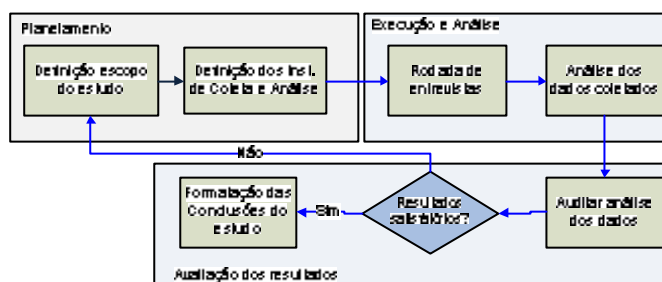
O propósito final é que os resultados deste estudo possam ser utilizados nas próximas implementações de programas de MPS a fim de contribuir para o sucesso das atividades do programa de MPS.

3. Metodologia de execução

Através de um estudo qualitativo, os fatores críticos de sucesso deste tipo de iniciativa foram identificados com a finalidade de se ter uma maior compreensão das influências desses fatores para um programa de melhoria. Este tipo de análise se faz necessária visto que, segundo SANTANA (2007), a implementação de iniciativas de melhoria de processo de software representa um grande desafio devido aos diversos fatores que influenciam no processo.

A metodologia seguida em cada fase deste estudo é apresentada na Figura 1 e consiste de três etapas: (1) Planejamento, (2) Execução e Análise e (3) Avaliação dos Resultados.

Na etapa de Planejamento defini-se o que será pesquisado e o que se espera de resultados. Em seguida, são preparados os instrumentos de coleta e análise de dados. Foram utilizados questionários semi-estruturados com questões abertas e para a análise de dados utilizou-se procedimentos de *Grounded Theory* - GT (STRAUSS e CORBIN, 1998), mais especificamente, as etapas de codificação aberta (envolve a quebra, a análise, a comparação, a conceituação e a categorização dos dados) e codificação axial (explicitam-se as relações entre os códigos, mostrando relações de causas e efeitos, condições intervenientes e estratégias de ação).



	responsável pelo programa de MPS	pelo programa de MPS dentro da organização. E necessário deixar claro o responsável pelo MPS para que possa esclarecer alguma dúvida
	Receio da implantação do programa de MPS	Era uma experiência nova nas organizações e para os colaboradores, deste modo houve um receio inicial
	Desconhecimento dos resultados reais do programa de MPS	Os colaboradores não tinham visão dos reais benefícios do programa de MPS, este conhecimento foi percebido após a implantação das melhorias no primeiro projeto
Fatores técnicos que influenciaram o MPS	Processos e resultados esperados do modelo MPS.BR	Os colaboradores julgaram ser de difícil compreensão, por isso foram realizados treinamentos nos processos que foram avaliados
	Atividades do MPS.BR	Algumas atividades eram vistas como obstáculo ao “trabalho real”. Isto reflete numa falta de priorização das atividades do programa de melhoria pelos colaboradores
	Documentação dos projetos	Foi necessário um treinamento básico a respeito de documentação de projetos. Isto acabou gerando atrasos nos artefatos que precisavam ser entregues
Impactos do programa de melhoria	Distribuição do conhecimento	Houve uma maior interação e troca de informações na equipe. Isto colaborou com os colaboradores, pois ficaram mais cientes do andamento das atividades.
	Otimização do trabalho	Atividades foram bem definidas e houve uma organização dos projetos, deixando-os mais entendíveis. Houve uma padronização nos documentos dos projetos
	Colaboradores mais detalhistas	Com os treinamentos e a importância do programa de MPS, os colaboradores passaram a realizar as atividades com mais atenção

Com a finalidade de complementar com mais fatores críticos de sucesso o questionário foi reformulado com base nos resultados obtidos nesta primeira fase.

4.2. Segunda fase: Após a avaliação do programa de Melhoria

A segunda fase deste estudo qualitativo está descrito detalhadamente em SANTOS *et al.* (2010) e ocorreu após a avaliação oficial do modelo de melhoria. Três colaboradores (organização C) que participaram ativamente do programa de MPS foram entrevistados. Desta forma, foi possível obter informações a respeito de dois momentos importantes do programa de melhoria.

Alguns resultados encontrados foram similares ao da primeira fase, destacando-se: Comprometimento dos colaboradores e o tipo de dedicação dos colaboradores. Um resumo dos demais resultados desta segunda fase é apresentado na Tabela 2.

Tabela 2. Resumo dos resultados da segunda fase

Tipos de Fatores	Fatores	Resultados Encontrados
Fatores que influenciaram a participação dos colaboradores	Resistência ao programa de MPS	Os colaboradores julgavam que as atividades do programa de MPS eram trabalhosas e complicadas criando uma resistência ao programa de MPS
	Conhecimento anterior ao programa de MPS	Alguns colaboradores julgaram ter conhecimentos em gerência de projetos e isto auxiliou na definição e execução das atividades do programa de MPS
	Decisões a respeito do programa de MPS	Algumas decisões da diretoria desagradavam alguns colaboradores, como a definição estratégica dos projetos/clientes utilizados na avaliação
Fatores técnicos que influenciaram o programa de melhoria	Padronização do processo	O processo foi padronizado através da documentação bem definida e das atividades necessárias. Esta padronização do processo possibilitou a visualização mais completa dos objetivos a serem alcançados nos projetos de software
	Organização dos artefatos	Houve melhoria na comunicação interna da equipe e na formalização da interação com o cliente. Colaboradores reportaram também que houve uma melhoria na qualidade do serviço prestado

	Falta de priorização para as atividades do programa de MPS	Os colaboradores também tinham que executar as atividades do dia-a-dia da empresa. Desta forma, o programa de melhoria possuía atividades que eram feitas posteriormente às atividades do cotidiano da organização
--	--	--

Dado a quantidade de aspectos técnicos, buscou-se na terceira fase analisar aspectos humanos que influenciaram o programa de MPS a fim de gerar um conjunto mais diversificado de Fatores Críticos de Sucesso em programas de MPS.

4.3. Terceira fase: Aprofundamento da Pesquisa sobre Fatores Críticos de Sucesso

Para aumentar os resultados sobre Fatores Críticos de Sucesso em Programas de MPS foram levantados aspectos humanos advindos do Comportamento Organizacional descritos por ROBBINS (2005). Desta forma foi possível analisar os aspectos humanos que influenciaram o programa de MPS.

O questionário foi adaptado novamente visando coletar informações a respeito dos aspectos humanos. Em seguida foi realizada uma entrevista-piloto com um colaborador da organização C para avaliar o questionário proposto.

As entrevistas foram realizadas quatro meses após a avaliação das organizações no programa de melhoria. Inicialmente, foram conduzidas três entrevistas na organização A, sendo os resultados deste trabalho apresentados em detalhes por Santos *et al.* (2011). Em seguida foram realizadas duas entrevistas na organização B.

Após analisar os resultados encontrados nas duas organizações, criou-se um conjunto de sugestões embasadas nos resultados do estudo qualitativo, conforme apresentados na Tabela 3.

Tabela 3. Sugestões embasadas nos achados da pesquisa

Aprendizagem	É fundamental verificar o conhecimento anterior dos colaboradores nos conceitos dos processos que serão implementados (no caso desta pesquisa Gerência de Projetos e Gerência de Requisitos) para criar direcionamentos adequados a fim de que os mesmos aprendam as atividades do programa de MPS de forma eficaz. Sugere-se fazer com que todos os colaboradores pratiquem o que foi aprendido em projetos-piloto ou pelo menos em sala de aula. Com isto é possível institucionalizar os processos dentro da organização.
Treinamento	Com os treinamentos é necessário verificar se os colaboradores realmente sabem o que fazer e como fazer. O treinamento também deve auxiliar na definição dos papéis dentro da organização.
Motivação	Aconselha-se criar meios para motivar os colaboradores em relação às atividades do programa de melhoria, como: criar uma política de reconhecimento dos colaboradores e um sistema de recompensas. É indicado também apresentar a real importância do programa de melhoria e explicar sobre a importância dos prazos, tentando evitar que isso se torne uma pressão nos colaboradores.
Personalidade	É relevante identificar colaboradores-chaves, ou seja, colaboradores que sejam pró-ativos e que consigam motivar outros colaboradores a fim de que eles possam auxiliar na disseminação do programa de MPS.
Percepção	É importante que tanto os colaboradores quanto a alta gerência tenha percepção do programa de MPS. Deste modo, ela pode estar informada sobre o que está acontecendo em relação às atividades do programa de melhoria na organização.
Eficácia de Liderança	É necessário verificar a capacidade de liderança do responsável pelo programa de MPS na organização sob pena de não se obter êxito na implementação do MPS. Notou-se também que a dedicação exclusiva do responsável pelo MPS foi importante para a execução das atividades.
Satisfação com o trabalho	A satisfação com as atividades do programa de MPS é obtida através de benefícios ao colaborador, como flexibilidade no horário de trabalho.
Tomada de decisão individual	As decisões devem tomadas de forma consciente, os motivos precisam ser analisados e discutidos com os responsáveis pelo programa de melhoria, antes de ser repassados para os demais colaboradores.
Avaliação de desempenho	É sugerido fazer uma avaliação entre os colaboradores a fim de identificar os pontos de melhoria, pois diferentes níveis de desempenho podem comprometer o programa de melhoria.
Estresse profissional	Deve-se ter cautela com eventos que ocorrem na organização como, por exemplo, na cobrança da finalização das atividades e no retrabalho de determinadas atividades do programa de melhoria.

4.4. Artigos publicados

- ? O artigo "Aplicando Grounded Theory para Compreender os Fatores Críticos de Sucesso em Iniciativas de Melhoria de Processo de Software" (MATOS *et al.*, 2010) foi apresentado no VI Workshop "Um olhar socio-técnico sobre a Engenharia de Software" (WOSES 2010), realizado em Belém, PA.

- ? O artigo “Resultados de um Estudo Qualitativo sobre a implementação do modelo MPS em empresas do programa AmazonSoft” (SANTOS *et al.*, 2010) foi apresentado no VI Workshop Anual do MPS (WAMPS 2010), realizado em Campinas, SP.
- ? O artigo “Aspectos humanos que afetam um programa de melhoria de processo de software - Uma análise qualitativa” (SANTOS *et al.*, 2011) foi aceito para apresentação no XIV Congresso Ibero-Americano em Engenharia de Software (CibSE 2011), a ser realizado no Rio de Janeiro, RJ.

4.5. Recursos Humanos Capacitados

Projeto Final de Conclusão de Curso de Vinícius Secatti: Bacharel em Ciência da Computação. Universidade Federal do Amazonas - UFAM

Projeto Final de Conclusão de Curso de Olavo Matos: Bacharel em Ciência da Computação. Universidade Federal do Amazonas - UFAM

4.6. Outros resultados

Resultados deste projeto fazem parte da pesquisa de dissertação de Davi Viana dos Santos: Mestrando em Informática. Universidade Federal do Amazonas – UFAM

5. Aplicabilidade dos resultados: Benefícios para a Indústria e Academia

Sendo o primeiro programa de Melhoria de Processo de Software realizado no Estado do Amazonas é necessário replicar esta iniciativa em outras empresas sediadas em nosso Estado.

Os resultados a respeito de fatores críticos de sucesso apresentados neste projeto podem contribuir para outras iniciativas de MPS, de forma a usar os fatores identificados como parte da estratégia de implementação e auxiliar no sucesso da execução das atividades do programa de Melhoria de Processo de Software, contribuindo assim para aumentar a competitividade das empresas locais no cenário Nacional.

Além dos benefícios para a indústria, vale ressaltar os benefícios acadêmicos deste estudo. Este projeto contribuiu para o “corpo de conhecimento” em fatores críticos de sucesso em programas de MPS, devido à diversidade dos fatores encontrados e a integração universidade-indústria, que contou com membros da universidade para executar o estudo e membros das organizações como parte do objeto de estudo.

6. Características inovadoras

Como característica inovadora deste projeto, temos o desenvolvimento de um estudo qualitativo nas primeiras empresas avaliadas no nível G do MPS.BR do Estado do Amazonas. Este tipo de estudo proporciona dados relevantes sobre a situação das organizações regionais em relação a programas de MPS do ponto de vista dos colaboradores membros das organizações. Estes colaboradores estavam diretamente ligados a programas de MPS. Desta forma, os dados obtidos estavam mais diretamente relacionados aos acontecimentos do programa de MPS.

7. Conclusões e trabalhos futuros

Este trabalho apresentou um estudo qualitativo a respeito dos fatores críticos de sucesso em programas de MPS desenvolvido neste ciclo do PBQP. Para a realização deste trabalho foram selecionadas três organizações pioneiras na implementação os processos do nível G do MPS.BR no estado do Amazonas.

Segundo CONTE *et al.* (2009), a contribuição dos métodos qualitativos no trato das questões intangíveis e cruciais para o profundo entendimento da problemática inerente à produção de software é notória. A escolha de um tipo de método apropriado para realizar é muito importante para a qualidade dos resultados que se deseja obter. Deste modo, se faz necessário o uso de métodos específicos para obter uma melhor compreensão da influencia dos fatores críticos de sucesso em um programa de MPS do ponto de vista dos colaboradores. Neste trabalho, foram aplicados procedimentos do método *Grounded Theory* para análise qualitativa, auxiliando na identificação dos fatores críticos de sucesso.

Neste estudo qualitativo foi possível verificar diversos fatores que influenciaram na implementação do programa de MPS, como: fatores que influenciaram na participação dos colaboradores no programa de melhoria, fatores que influenciaram diretamente o programa de melhoria e fatores de impacto do MPS na organização. Além disso foram verificados alguns aspectos humanos que influenciaram o programa de melhoria como: aprendizagem, motivação, satisfação com o trabalho e estresse profissional.

Como trabalhos futuros, tem-se a aplicação dos resultados apresentados de forma a apoiar outras estratégias de implementação de MPS e a replicação deste estudo qualitativo nos mais diversos programas de MPS do país com a finalidade de criar uma base de conhecimento que torne possível a extração de similaridade dos fatores críticos de sucesso que influenciam nestas iniciativas.

Agradecimentos

Os autores agradecem a todos os colaboradores das organizações entrevistadas e ao apoio financeiro da FAPEAM e do CNPq. Os autores agradecem também a colaboração do Dr. Cleidson de Souza e Dr. Dalton Vilela no auxílio dos procedimentos qualitativos durante grande parte do desenvolvimento deste projeto.

Referências Bibliográficas

- BADDOO, N., HALL, T., (2003), "De-motivators for software process improvement: an analysis of practitioners' views", *Journal of Systems and Software*, v. 66, n. 1, pp. 23-33.
- CONTE, T., CABRAL, R., TRAVASSOS, G.H., 2009, "Aplicando Grounded Theory na Análise Qualitativa de um Estudo de Observação em Engenharia de Software – Um Relato de Experiência". In: V WOSSES 2009 v. 1, pp. 26 - 37, Ouro Preto, MG, Brasil.
- DYBA, T., (2000), "An Instrument for Measuring the Key Factors of Success in Software Process Improvement", *Journal of Empirical Software Engineering*, v. 4, pp. 357-390.
- DYBA, T., (2003), "Factors of software process improvement success in small and large organizations: an empirical study in the scandinavian context", *Proceedings of the 9th European software engineering conference*, Helsinki, Finland.
- JALOTE, P. (1997). "An integrated approach to software engineering". Springer, 2a.edição.
- MATOS, O., SECATTI, V., SANTOS, D., OLIVEIRA, H., CONTE, T. (2010). "Aplicando Grounded Theory para Compreender os Fatores Críticos de Sucesso em Iniciativas de Melhoria de Processo de Software". WOSSES 2010.
- MONTONI, M. ROCHA, A. R. (2007). "A Methodology for Identifying Critical Success Factors that Influence Software Process Improvement Initiatives: An Application in the Brazilian Software Industry", *Em EuroSPI 2007*, LNCS 4764, pg. 175-186.
- MONTONI, M., ROCHA, A. R. (2010). "Aplicação de Grounded Theory para Investigar Iniciativas de Implementação de Melhorias em Processos de Software". In: *Anais do IX SBQS*, pp 167-181.
- ROBBINS, S.P., 2005, "Comportamento Organizacional", Pearson Education do Brasil Ltda.
- SANTANA, A. (2007). "Problemas em Iniciativas de Melhoria de Processos de Software sob a Ótica de uma Teoria de Intervenção". *Dissertação de Mestrado*. UFPE. Março 2007.
- SANTOS, D.V., RABELO, J., MAR, C., et al., 2010, "Resultados de um Estudo Qualitativo sobre a implementação do modelo MPS em empresas do programa AmazonSoft ". In: *VI Workshop Anual do MPS (WAMPS 2010)*, Campinas, SP.
- SANTOS, D.V., VILELA, D.C.J., SOUZA, C.D., CONTE, T., 2011, "Aspectos humanos que afetam um programa de melhoria de processo de software - Uma análise qualitativa". In: *XIV Congresso Ibero-Americano em Engenharia de Software*, Rio de Janeiro, RJ - Brasil (Artigo aceito para publicação).
- STRAUSS, A., CORBIN, J., 1998, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 2 ed. ed. London, SAGE Publications.

GPO - Gestão de Processos Organizacionais uma Metodologia para Promoção da Qualidade

Renato Luiz Della Volpe¹, Ana Cecília Peixoto Zabeu¹, Sergio Massao Jomori¹, Edvar Pera Junior²

¹ ASR Consultoria e Assessoria em Qualidade Ltda.

Rua Doutor Deodato Wertheimer, 1352 - sala 11 – Mogi das Cruzes – SP – Brasil

² Associação pela Excelência do Software de Campinas - Núcleo Softex Campinas
Estrada da Telebrás, km 0,97 – UNICAMP – Campinas – SP – Brasil

(renatovolpe@asrconsultoria.com.br; anazabeu@asrconsultoria.com.br;
sergiojomori@asrconsultoria.com.br; edvar@cps.softex.br)

1. Introdução

É notória a evolução e busca de melhorias na gestão organizacional das empresas de tecnologia da informação [MCT-2009]. No âmbito destas melhorias, entre outros fatores como a competitividade no mercado, melhores resultados financeiros e operacionais, estão a preocupação e o entendimento de que resultados somente serão obtidos com o estabelecimento e a melhoria dos processos organizacionais. Contudo, apesar dos esforços que estão sendo realizados, ainda existem diversas dificuldades para micro e pequenas empresas em iniciar sua atuação no estabelecimento e melhoria de processos. Entre estas dificuldades estão a baixa capacidade de alocar recursos financeiros para a realização de projetos de melhoria de processos e o entendimento de qual modelo ou norma de gestão é o mais adequado às necessidades da empresa para garantir uma atuação mais eficiente e eficaz. Este trabalho apresenta experiências realizadas na implementação em diversas empresas de TIC de uma metodologia de gestão da qualidade para minimizar estas dificuldades e os riscos a elas associadas.

2. Objetivos e Justificativa:

Este trabalho apresenta a implantação, a implementação e os resultados obtidos até o momento, de metodologia desenvolvida para a melhoria de processos e gestão organizacional em empresas de TIC – Tecnologia da Informação e Comunicação. O foco principal são as micro e pequenas empresas de TIC que, por seu porte, dispõem de limitados recursos financeiros para implantação de metodologias tradicionais, mas já se conscientizaram da necessidade de uma melhor estruturação para seu crescimento. Com isto, os objetivos e motivações do trabalho são:

2.1 Objetivos principais

- a) Incentivar a mudança de cultura das empresas, incitando o uso de metodologias e promovendo a gestão da qualidade nas empresas, dando condições para o conhecimento básico e inicial de gestão;
- b) Fomentar a necessidade de implementar processos que irão contribuir para o crescimento sustentado das organizações e
- c) Conscientizar os empresários sobre a necessidade estratégica de adequação a modelos e normas da qualidade.

2.2 Motivações

As motivações para a estruturação da metodologia GPO – Gestão de Processos Organizacionais partiram da experiência da ASR em empresas de TIC, observação de diversas dificuldades levantadas e por vezes declaradas em análises críticas de atuação, feitas em parceria com clientes, quando da implementação de modelos de qualidade CMMI e MPS.BR ou normas de gestão NBR/ISO 9001. Independente da atuação ser isolada ou em grupos de empresas, as barreiras e dificuldades levantadas demonstravam ser comuns. A correlação entre estas barreiras e dificuldades e as respectivas motivações para a criação da metodologia desenvolvida estão descritas na Tabela 1.

Tabela 1 - Dificuldades e barreiras x Motivações

Principais dificuldades/ barreiras	Motivações para atuação
Pouco envolvimento da Alta Direção	Estabelecer mecanismos para maior envolvimento.
Falta de alinhamento entre estratégia e atuação	Aplicação de conceitos de liderança de SGQ – Sistema de Gestão da Qualidade
Dificuldade em “escrever” processos	Capacitação para estruturar e documentar processos
Dificuldade em entender os problemas da organização	Estabelecer mecanismos de entendimento dos problemas e priorização de atuação
Alinhar os processos recomendados por modelos ou normas às necessidades	Estabelecer mecanismos de priorização de ações
Dificuldade em sistematizar processos	Estabelecer a aplicação de conceitos básicos de sistematização
Falta de recursos para definição de processos	Estabelecer uma “velocidade” de aplicação conforme recursos e limitações
Dificuldade na escolha do “modelo” a aplicar	Estabelecer uma integração entre modelos para nivelar conhecimentos e auxiliar na decisão

A correlação entre as motivações e como a metodologia GPO foi desenvolvida será apresentada no detalhamento deste trabalho e em sua conclusão, permitindo o rastreamento das ações realizadas frente às principais dificuldades/ barreiras acima citadas.

3. Metodologia de Execução

A idealização da metodologia se iniciou em 2007 com a estruturação de um grupo de trabalho no interior paulista para melhoria do processo de gestão de projetos. Foi realizado um levantamento de necessidades das 16 empresas participantes e verificou-se que muitos dos problemas e dificuldades das empresas estavam relacionados com os relatados na Tabela 1. No mesmo período a ASR e o Núcleo Softex Campinas fizeram uma análise dos trabalhos até então realizados na implementação em grupos de trabalho do CMMI e MPS.BR e concluíram que havia necessidade de estruturar uma metodologia focada nas micro e pequenas empresas de TIC que permitisse a gestão da qualidade nestas empresas e que apresentasse algumas características importantes em seu contexto:

- ? implementação rápida, focada e adequada às necessidades de cada empresa;
- ? auxílio às empresas para estabelecerem estratégias de escolha do modelo, norma ou método mais adequado às suas necessidades;
- ? baixo custo de implementação e
- ? resultados rápidos

Todas estas necessidades levaram a estabelecer a metodologia GPO de maneira a atender não somente às necessidades iniciais dos grupos, como estruturar uma forma de difundi-la a outros grupos e localidades.

3.1 Principais referências e enfoques

A metodologia GPO foi estruturada considerando as boas práticas de gestão e pode ser evoluída, conforme novas necessidades e práticas forem sendo observadas ou requeridas. Na estruturação da metodologia foram adotados modelos ou normas como principais referências, que provêm melhores resultados atualmente, como segue:

- ? MPS.BR– Melhoria do Processo do Software Brasileiro [SOFTTEX 2009] – com enfoque principal nos níveis G (Gerência de Requisitos e Gerência de Projetos) e F (Gerência de Configuração). Quanto aos demais processos do nível F (Garantia da Qualidade, Medição e Aquisição) também existe a atuação no GPO, mas foram estruturados com base na NBR/ISO 9001:2008 conforme abaixo.
- ? CMMI-DEV – *Capability Maturity Model Integration for Development* [CMMI 2005] – com enfoque no nível 2 respeitando as considerações acima feitas para o MPS.BR.
- ? NBR/ISO 9001:2008 – Sistema de Gestão da Qualidade [NBR ISO 9000 2005] – com enfoque principal na estrutura de gestão e liderança, garantia da qualidade, medição e análise e aquisição.
- ? ITIL – Information Technology Infrastructure Library [ITIL 2007] – enfoque principal de atuação em processos de pós-venda e suporte.

- ? PPQG – Prêmio Paulista de Qualidade da Gestão [PPQG. IPEG 2010] – baseado no PNQ- Prêmio Nacional da Qualidade, foi aplicado para a estruturação do método de auto-avaliação que possui como premissas promover o maior envolvimento da alta direção, proporcionar o entendimento da real situação da empresa e das melhores estratégias e permitir uma evolução gradativa e alinhada às estratégias de cada organização.
- ? Metodologias Ágeis (ex: SCRUM / XP) [SCRUM. InofQ 2009] – aplicação de conceitos ágeis de gestão de projetos que se mostraram eficazes para empresas de desenvolvimento e manutenção de software e de pequeno e médio porte.

Com base nestas boas práticas, os principais enfoques do GPO de atuação são:

3.1.1 Liderança

Enfoque para o desenvolvimento da Missão, Visão, Valores e Políticas de cada empresa, bem como a análise crítica para sua manutenção e aplicação interna. Inclui também o envolvimento da alta direção no monitoramento da evolução da gestão da organização, conscientizando os empresários sobre a necessidade de inserir na cultura da empresa a busca da excelência através do uso de metodologias e melhores práticas em controle de processos.

3.1.2 Processos - estruturação

Enfoque para o desenvolvimento dos processos, incluindo a Documentação do Processo; Fluxo de Processo; Integração entre processos; Definição de papéis e responsabilidades; Descrição, quando aplicável, de Instruções Operacionais; Definição de recursos e materiais de suporte (*templates*, *check-list*, planilhas, *scripts*, ferramentas, etc.). São consideradas também ferramentas para a documentação dos processos e sua forma de registro e arquivamento.

São levados em conta os processos principais e de apoio [PNQ. FNQ 2010] sendo:

- ? Processos principais: Processos que agregam valor diretamente para os clientes. Estão envolvidos na geração do produto e na sua venda e transferência para o comprador, bem como na assistência após a venda e disposição final.
- ? Processos de apoio: Processos que sustentam os processos principais do negócio e a si mesmos, fornecendo produtos e insumos adquiridos, equipamentos, tecnologia, softwares, recursos humanos e informações.

3.1.3 Processos – institucionalização e gestão

Enfoque para apoiar a institucionalização/ sistematização dos processos desenhados e documentados, para o início de sua aplicação na organização. São estabelecidas formas de capacitação das pessoas, avaliação da aplicação dos processos, análise crítica da aplicação dos processos, registros necessários para a evolução e melhoria dos processos.

3.2 Metodologia GPO

A metodologia GPO foi inicialmente estruturada em 5 (cinco) blocos de atuação implementados de forma evolutiva. Para cada bloco era realizado um Workshop e ocorria a atuação específica de consultoria de aplicação conforme seu enfoque .

Os enfoques iniciais de cada bloco eram:

1. Gestão de Processos Organizacionais em empresas - autoavaliação
2. Estruturação de Processos Organizacionais
3. MPS.BR e ISO 9001
4. ITIL
5. Conclusão dos Processos – apresentação das empresas e encerramento

A partir dos bons resultados alcançados em 03 (três) grupos de empresas e das análises críticas realizadas com todos os envolvidos, observou-se a necessidade de modificações e melhorias da estruturação, considerando principalmente:

1. Melhor adequação às necessidades de cada grupo de empresas
2. Estruturação que considerasse a maturidade já existente de algumas empresas
3. Estruturação que propiciasse a continuidade de atuação e evolução constante.

Com base nestas análises críticas, a metodologia evoluiu para a estrutura conforme Figura 1 com programas modulares para diversos focos (serviços, infraestrutura, software, produto) e com níveis graduais de atuação:

- ? Nível básico: programas para introdução de conceitos fundamentais de processos
- ? Nível intermediário: programas para continuidade de ações de melhoria
- ? Nível avançado: programas visando “certificação”

O GPO prevê como primeira atividade a realização de um Workshop de Avaliação Prévia com as empresas interessadas, onde se define qual será a forma de atuação por grupo de empresas e a estruturação dos módulos customizados. Cada programa de atuação é composto por módulos específicos, conforme Figura 2.

No início do programa realiza-se uma autoavaliação em cada empresa participante, acompanhada por módulos de consultoria. Na autoavaliação são utilizados critérios objetivos de avaliação baseados no PPGQ com uma pontuação máxima de 250 ou 500 pontos [PPQG. IPEG 2010, classes Bronze e Prata], dependendo do foco de atuação. A pontuação é distribuída nos diversos itens de avaliação (ver exemplos nas Tabelas 2 e 3). Cada item de avaliação possui o descritivo dos requisitos a serem atendidos e é pontuado considerando-se os percentuais de três dimensões [PPQG. IPEG 2010 – PNQ. FNQ 2010]:

- ? Enfoque: refere-se ao grau em que as práticas de gestão da organização apresentam adequação ao que é requerido e sua proatividade.
- ? Aplicação: refere-se ao grau em que as práticas de gestão da organização apresentam disseminação (aplicação ou não em toda organização) e continuidade (uso sistemático e não esporádico das práticas)
- ? Melhoria Contínua: refere-se ao grau em que a empresa aplica as lições aprendidas para a melhoria dos processos e da gestão.



Figura 1: Estrutura atual da metodologia GPO

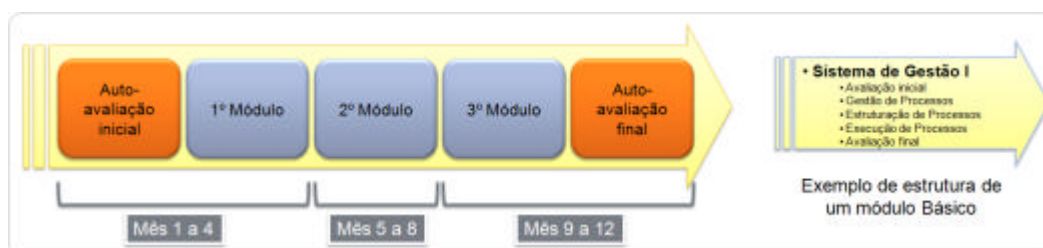


Figura 2: Composição de um programa e exemplo de um módulo

Com os resultados obtidos, os itens de avaliação são alinhados às estratégias e respeitando as restrições das empresas, estes são priorizados, resultando em um plano de ações para o período de atuação.

O período de atuação típico para a implementação das diversas atividades é de 8 a 12 meses, mas pode variar dependendo das necessidades de cada grupo de empresas. O propósito, neste período, é que a empresa consiga adquirir um entendimento claro “do que fazer”, “o que usar” (qual ou quais metodologias são as mais adequadas à realidade e às necessidades da empresa) e “como fazer” para gradativamente consolidar e aprimorar seus principais processos.

Nem todas as empresas do grupo necessitam ter o mesmo ritmo, que é ditado pelos seus recursos alocados. Contudo, metas a serem atingidas no período são estabelecidas no início do projeto (após avaliação inicial).

Tabelas 2 e 3 - Critérios de avaliação e pontuação – resumido.

0%		20%		40%		60%		80%		100%	
Enfoque											
As práticas de gestão são inadequadas aos requisitos, ou não estão relacionadas/ praticadas	As práticas de gestão existentes são adequadas para alguns requisitos do Item. Reativo.										
Aplicação											
O conjunto de práticas de gestão não abrange as áreas, processos, produtos ou partes interessadas pertinentes ou	O conjunto de práticas de gestão abrange algumas áreas, processos, produtos ou partes interessadas pertinentes.										
Continua											
Não há melhorias sendo implantadas.	Há melhorias sendo implantadas.										

		Pontos possíveis	Percentual Obtido	Pontuação Obtida
Liderança				
LID1	Estabelecimento da Gestão Básica da Organização	20		0
LID2	Exercício da Liderança	20		0
TOTAL DE LIDERANÇA		40		0
Processos				
PRO1	Processos - estruturação	20		0
PRO2	Processos - Gestão	20		0
		40		0
PRP1				0
PRP2				0
PRP3				0
PRP4				0
PRP13	Pós-venda			0
PRA1				
PRA2				
PRA3				
PRA4				
PRA5				
PRA6				
PRA10	Aquisição de MP, peças e serviços de terceiros			
TOTAL GERAL		500		

4. Resultados Obtidos e sua aplicabilidade

Os resultados da aplicação da metodologia GPO podem ser apresentados com os enfoques de:

- a. Resultados da evolução das empresas: O gráfico 1 representa a evolução de pontuação das empresas de um dos grupos durante o período de atuação. O gráfico 2 representa a faixa de pontuação das diversas empresas no início e no final dos trabalhos. Esta classificação de faixas de pontuação caracteriza a empresas em cinco estágios de gestão da qualidade que são estabelecidos pelo PPQG conforme Tabela 3.
- b. Resultados da continuidade na gestão da qualidade: Dos grupos em atuação já existe a necessidade de empresas continuarem na evolução dos processos em um percentual que varia entre 40% a 60%, as quais estabeleceram como meta de melhoria em um dos enfoques abaixo:
 - ? Continuidade na evolução dos processos e da pontuação/ faixa de classificação da empresa.
 - ? Buscar a certificação NBR/ISO 9001.
 - ? Buscar a adequação e avaliação formal do MPS.BR. Neste tópico já existem casos reais, sendo uma empresa Nível F e quatro empresas Nível G e outras atuando em grupos subsidiados pelo Programa MPS.BR.
 - ? Estruturação e aplicação de ITIL aos serviços
- c. Resultados na formação de novos grupos: Os resultados obtidos e analisados criticamente no início e término das atividades de cada grupo, estão resultando na criação de novos grupos.

5. Características Inovadoras

O projeto agrupa de forma estruturada e analisada criticamente as melhores práticas de gestão em aplicação nos dias atuais. Contudo, sua aplicação é inovadora no sentido de dar condições às empresas de se autoavaliarem e definirem apropriadamente as necessidades de implementação suportadas por orientação de consultorias de aplicação. Além disso, embasam as empresas para trilhar os caminhos mais adequados de escolha do método de gestão da qualidade mais adequado às suas necessidades.

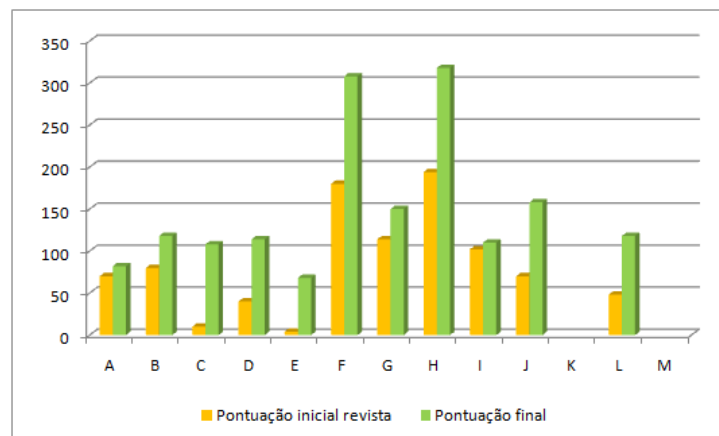


Gráfico 1 - evolução da pontuação das empresas – Inicial e Final

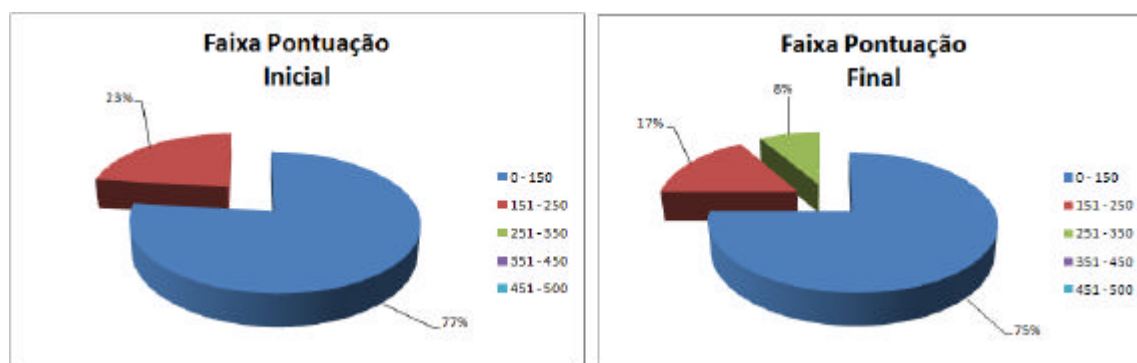


Gráfico 2 - Evolução das faixas de avaliação

6. Conclusão e Perspectivas futuras

Frente as dificuldades e barreiras que geraram as motivações da constituição da metodologia GPO (ver Tabela 1), as diversas ações apresentadas neste artigo e retratadas no mapeamento da Tabela 4, contribuíram para a evolução da maturidade das empresas e ao atendimento do objetivo principal em promover a consciência e importância da atuação na gestão da qualidade.

Tabela 3 - Faixas de classificação das empresas (*) 500 pontos (Prata)

(*) detalhes da planilha foram suprimidos devido ao espaço. A tabela completa pode ser encontrada em [PPQG. IPEG 2010 - PNQ. FNQ 2010].

Faixa	Pontuação	Estágio da Organização
5	451 - 500	
4	351 - 450	
3	251 - 350	
2	151 - 250	Já é possível detectar a existência de práticas de gestão estruturadas voltadas para responder aos requisitos dos critérios de avaliação, ainda com lacunas importantes nas questões de disseminação, proatividade, aprendizado e integração. A tomada de decisão é feita parcialmente com base nas informações estruturadas e o processo de comparação com referenciais externos ainda carece de melhor estruturação.
1	0 - 150	Estágio muito preliminar de desenvolvimento do sistema de gestão organizacional, o que recomenda a adoção de critérios de avaliação de gestão melhor adaptados ao seu estágio de desenvolvimento.

Tabela 4 - Motivações x Ações de melhoria

Motivações para atuação (conforme Tabela 1)	Ações e enfoques da metodologia que contribuem para os resultados					
	Auto avaliação	Estabelecimento da Missão/ Visão e Valores	Capacitação sobre processos	Melhores práticas respeitando limitações	Consultoria de orientação e suporte	Planejamento e priorização respeitando limitações e restrições das empresas
Estabelecer mecanismos para maior envolvimento da alta direção.	✓	✓				
Aplicação de conceitos de liderança de SGQ – Sistema de Gestão da Qualidade		✓				
Capacitação para estruturar e documentar processos			✓	✓	✓	
Estabelecer mecanismos de entendimento dos problemas e priorização de atuação	✓		✓		✓	✓
Estabelecer a aplicação de conceitos básicos de sistematização		✓		✓	✓	
Estabelecer uma “velocidade” de aplicação conforme recursos e limitações				✓	✓	✓
Estabelecer uma integração entre modelos para nivelar conhecimentos e auxiliar na decisão				✓	✓	

7. Referências Bibliográficas

- CMMI. Capability Maturity Model® Integration (CMMISM), Version 1.2 CMMI-DEV for Development (CMMI-DEV, V1.2), 2005
- ITIL. Official ITIL® Website – ITILV3 - <<http://www.itil-officialsite.com/>>, 2007
- MCT. Ministério da Ciência e Tecnologia. Secretaria de Política de Informática e Automação. Qualidade e Produtividade no Setor de Software Brasileiro. Disponível em: <<http://www.mct.gov.br/sepin/>>. Acesso em: 2009
- NBR ISO 9000. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS NBR ISO 9000 - Sistemas de gestão da qualidade - Fundamentos e vocabulário Rio de Janeiro: ABNT, dezembro 2005, 35
- PPQG. IPEG - Instituto Paulista de Excelência da Gestão, PPQG – Prêmio Paulista de Qualidade da Gestão de 2010 < <http://www.ppqg.org.br/ppqgoque.asp>>
- PNQ. FNQ – Fundação Nacional da Qualidade – Prêmio Nacional da Qualidade 2010 < <http://www.fnq.org.br/site/397/default.aspx>>
- SCRUM. InfoQ - Agile Community Content on InfoQ - Scrum and XP from the Trenches - <<http://www.infoq.com/agile>> 2009
- SOFTEX. Guia Geral:2009 (Maio de 2009, atualizado em setembro de 2009 – Melhoria de Processo de Software Brasileiro <http://www.softex.br/portal/mpsbr/_guias/default.asp>

Melhoria do Processo de Aquisição de Software de 2008 a 2010

Kival Chaves Weber¹, Danilo Scalet²

1Associação para Promoção da Excelência do Software Brasileiro (SOFTEX)

2Companhia de Informática do Paraná (CELEPAR)

kival.weber@nac.softex.br, danilo@celepar.pr.gov.br

Abstract. *This paper describes actions to improve the software and related services Acquisition process in Brazil, based in the MPS Model, from 2008 to 2010 with a focus on the 2010 cycle. These actions are being implemented from 2004 on and, recently, were improved by means both of MPS.BR Program new actions here described, coordinated by SOFTEX, and the MPS Acquisition process customization to comprehend Brazilian legal requirements aiming its use in Government IT acquisitions.*

Resumo. *Este artigo descreve ações de melhoria do processo de Aquisição de software e serviços correlatos (S&SC) no Brasil, baseadas no Modelo MPS, de 2008 a 2010 com foco no ciclo 2010. Estas ações vêm sendo implementadas no Brasil desde 2004 e, recentemente, foram aprimoradas tanto por meio de novas ações do Programa MPS.BR aqui descritas, sob coordenação da SOFTEX, quanto pela personalização do processo de Aquisição MPS para incluir requisitos legais Brasileiros visando seu uso nas aquisições governamentais de TI.*

Introdução

A Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) coordena as ações do Programa MPS.BR – Melhoria de Processo do Software Brasileiro desde a sua criação em dezembro de 2003. Este programa mobilizador tem duas metas [WEBER et al, 2006]: i) uma meta técnica visando à criação e ao aprimoramento do Modelo MPS, que é composto pelo modelo de referência MR-MPS, método de avaliação MA-MPS e modelo de negócio MN-MPS; ii) uma meta de mercado visando à disseminação e adoção do Modelo MPS em organizações, públicas e privadas, em todas as regiões do país. O Modelo MPS está documentado nos seguintes Guias MPS: Guia Geral, Guia de Aquisição, Guia de Implementação e Guia de Avaliação. Para mais informações sobre o Programa MPS.BR e o Modelo MPS, consulte www.softex.br/mpsbr

O Guia de Aquisição descreve um processo de Aquisição de software e serviços correlatos (S&SC) baseado na Norma Internacional ISO/IEC 12207, complementado pela norma IEEE STD 1062:1998. Este documento visa servir como um guia para empresas que adquirem S&SC, detalhando as atividades e tarefas envolvidas, descrevendo os produtos de trabalho e fornecendo exemplos de preenchimento dos principais documentos. Não é objetivo deste guia servir como Guia de Implementação para um processo de aquisição que venha a ser avaliado utilizando-se o MA-MPS, pois este é o propósito de outros guias do Modelo MPS. A versão vigente é o Guia de Aquisição:2009, que sucedeu as versões 1.0 (2005), 1.1 (2006) e 1.2 (2007).

A estrutura deste artigo final sobre o projeto 2.07 do PBQP Software - ciclo 2010 segue as orientações publicadas pelo MCT/SEPIN em www.mct.gov.br. Após a contextualização do projeto nesta seção 1 - Introdução, a seção 2 apresenta seu Objetivo e Justificativa. A seção 3 - Metodologia de Execução explica como o projeto foi desenvolvido. A seção 4 - Resultados Obtidos cita os produtos e resultados relevantes obtidos. A seção 5 - Aplicabilidade dos Resultados explicita tanto a relevância, abrangência e impacto dos resultados obtidos quanto as características inovadoras do projeto. A seção 6 - Conclusão e Perspectivas Futuras encerra este trabalho.

Objetivo e justificativa

O objetivo geral deste projeto submetido ao PBQP Software pela SOFTEX/MPS.BR, na área de Aquisição da ETM – Equipe Técnica do Modelo MPS, foi contribuir para a melhoria do processo de Aquisição de software no Brasil, por meio do processo de Aquisição do Modelo MPS. Embora este projeto seja plurianual, pois foi realizado no período 2008 a 2010, este artigo tem seu foco no ciclo 2010.

A justificativa deste projeto visava ao atendimento de necessidades específicas tanto de aprimorar o processo de Aquisição MPS quanto de aumentar sua disseminação e adoção em todas as regiões do Brasil.

Metodologia de execução

Para alcance do objetivo geral, no âmbito da governabilidade do Programa MPS.BR, foram propostos cinco resultados:

- ? Aprimoramento do processo de Aquisição MPS, por meio de novas versões dos Guias MPS que dispõem sobre o processo de Aquisição;
- ? Capacitação de pessoal, por meio tanto de cursos e provas oficiais de Aquisição MPS (C4 e P4) quanto de cursos especiais de Aquisição MPS, além da habilitação de Consultores de Aquisição (CA) pela

SOFTEX/MPS.BR;

- ? Atualização dos COMUNICADOS SOFTEX/MPS.BR 12/2006 e 03/2008 relativos ao processo de Aquisição MPS;
- ? Ações visando à disseminação e adoção dos guias que tratam do processo de Aquisição MPS, tanto no Brasil quanto no exterior;
- ? Submissão de livros, artigos técnicos e relatos de experiência publicados em renomadas chamadas nacionais e internacionais.

Resultados obtidos

Observando que se trata de um projeto plurianual (2008-2010), podem ser destacados os seguintes resultados no ciclo 2010:

Aprimoramento do processo de Aquisição MPS

Em 2010, os Guias MPS não foram atualizados. Isto foi feito em 2009, com destaque para a publicação em Português e Espanhol do Guia de Aquisição:2009 e a publicação em Português do Guia de Implementação:2009, parte 8 - Implementação do MR-MPS (níveis G até A) em organizações que adquirem software.

Capacitação de pessoas no processo de Aquisição MPS

Na área de Aquisição MPS, participaram de cursos oficiais C4 (16h): 320 pessoas de 2008-2010, das quais 24 em 2010; foram aprovadas em provas oficiais P4 (4h, sem consulta): 52 pessoas de 2008-2010, das quais 5 em 2010.

Em 2010, foram criados cinco cursos especiais na área de Aquisição MPS:

- Requisitos para Aquisição de Software (8h)
- Seleção de Fornecedores de Software (8h)
- Definição de Critérios para Aquisição de Software (16h)
- IN 04/2010 SLTI/MP e PDTI (16h)
- Contratação de Serviços de TI para Órgãos Públicos (16h e 40h)

Atualização de COMUNICADOS SOFTEX/MPS.BR na área de Aquisição

O COMUNICADO 12/2006 – ICA e CA foi atualizado em 12JUL2010:

- apresentando as condições para o credenciamento de Instituições de Consultoria de Aquisição (ICA), uma boa novidade nesta área
- aperfeiçoando os requisitos para habilitação dos Consultores de Aquisição (CA)

O COMUNICADO 03/2008 – Disseminação do Processo de Aquisição de Software também foi atualizado em 12JUL2010:

- descrevendo oportunidades que podem ser aproveitadas para disseminar o Processo de Aquisição de software, tanto pela SOFTEX quanto por Agentes SOFTEX, ICA e CA

Em 31DEZ2010, havia:

- oito CA – Consultores de Aquisição habilitados
 - Ana Guerra, Campinas-SP
 - Danilo Scalet, Curitiba-PR
 - Edméia Andrade, Brasília-DF
 - João Condack, Rio de Janeiro-RJ
 - Lúcia Nigro Pinheiro, Niterói-RJ
 - Pedro da Silva Junior, Goiânia-GO
 - Regina Colombo, Campinas-SP
 - Rodrigo Quites Reis, Belém-PA
- duas ICA – Instituições de Consultoria de Aquisição autorizadas
 - PRIME UP (validade até 20 de outubro de 2012)
 - Q4E (validade até 16 de agosto de 2012)

Disseminação e adoção do processo de Aquisição MPS

Em 2010, foram realizadas ações visando à disseminação e adoção do processo de Aquisição MPS, tais como:

- ações diretas da SOFTEX (p.ex.: divulgação do processo de Aquisição MPS junto à ABEP – Associação Brasileira de Entidades Estaduais de Tecnologia da Informação, a cargo de Danilo Scalet junto com José Antonio Antonioni)
- ações regionais de Agentes SOFTEX (p.ex.: tanto divulgação regional do processo de Aquisição MPS junto a potenciais interessados, públicos e privados, quanto identificação de oportunidades para parcerias com ICA e CA)
- negócios tanto de ICA quanto de CA

Em 2010, foram incluídas no Plano Operativo (POA ABR2010-DEZ2011) do Projeto BID RELAIS – Rede Latino Americana da Indústria de Software ações visando à disseminação e adoção do processo de Aquisição MPS nos países que participam deste projeto: Brasil, Colômbia, México e Peru. Submissão de livros, artigos técnicos e relatos de experiências sobre o processo de Aquisição MPS

Dentre os artigos técnicos e relatos de experiências sobre o processo de Aquisição MPS submetidos tanto no país (p. ex.: PBQP Software, SBQS e WAMPS) quanto no exterior (p. ex. QUATIC) destacam-se [WEBER et al, 2007; GILLIS e ANDRADE, 2009; SCALET e ANDRADE, 2009].

Neste período merece destaque especial o desenvolvimento do “Processo de contratação de software e serviços correlatos para entes governamentais (PCSSC-EG)”, no âmbito de um projeto submetido ao PBQP Software – ciclo 2009, buscando compatibilizar o processo de aquisição de software e serviços correlatos com a legislação pública brasileira. Em 2010, esse projeto foi laureado pelo MCT/SEPIN com o Prêmio Dorgival Brandão Júnior da Qualidade e Produtividade em Software - ciclo 2009 [CRUZ et al, 2010].

Aplicabilidade dos Resultados

Relevância

O Plano de Ação 2007-2010: Ciência, Tecnologia e Inovação para o Desenvolvimento Nacional, lançado pelo Governo Federal em 20NOV2007, a Prioridade Estratégica III – Pesquisa, Desenvolvimento e Inovação em Áreas Estratégicas (Tecnologias da Informação e Comunicação), contempla o seguinte na seção 8.2 - Programa de Estímulo ao Setor de Software e Serviços: Promover melhores práticas – Estimular a adoção das melhores práticas de engenharia de software pelas empresas do setor em seu processo de desenvolvimento bem como nos processos de aquisição de software e serviços correlatos, em conformidade com o estado da arte e as normas internacionais de qualidade aplicáveis (o grifo é dos autores do artigo).

Em 2010, a Secretaria de Logística e Tecnologia de Informação (SLTI)/Ministério do Planejamento, Orçamento e Gestão (MP) revisou a Instrução Normativa IN-04/2008 SLTI/MP, republicada como a nova IN-04/2010 SLTI/MP que dispõe sobre o processo de contratação de soluções de TI pelos órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática (SISP) do Poder Executivo Federal.

Impacto

MPS.BR é um programa mobilizador, de longo prazo, que visa à disseminação de normas, modelos e melhores práticas tanto para a melhoria dos processos de desenvolvimento e manutenção de software quanto para o processo de Aquisição de software e serviços correlatos.

Segundo o Tribunal de Contas da União (Acórdão 2.308/2010-TCU-Plenário), o orçamento de gastos em TI para 2010 somente na Administração Pública Federal (APF) era de pelo menos R\$ 12,5 bilhões (US\$ 7.5 billion), sendo parte significativa desse orçamento dirigido para a contratação de serviços relacionados a software.

Abrangência

Melhoria crescente do processo de Aquisição de software tanto em organizações públicas federais, estaduais e municipais, quanto em empresas privadas principalmente nas de grande porte.

Inovação

A melhoria do processo de Aquisição de software é uma ação nova, com características inovadoras, tanto no Brasil quanto no exterior.

Merece destaque a atualização do COMUNICADO 12/2006 – ICA e CA (atualizado em 12JUL2010) tanto apresentando as condições para o credenciamento de Instituições de Consultoria de Aquisição (ICA), uma boa novidade nesta área, quanto aperfeiçoando os requisitos para habilitação dos Consultores de Aquisição (CA)

Conclusão e perspectivas futuras

Este artigo descreve ações de melhoria do processo de Aquisição MPS, de 2008 a 2010, com foco no ciclo 2010. Estas ações, que vêm sendo implementadas no Brasil desde 2004 [ALVES e GUERRA, 2004], recentemente foram aprimoradas tanto por meio das novas ações da SOFTEX/MPS.BR descritas neste trabalho quanto pela personalização do processo de Aquisição MPS para uso nas contratações governamentais descritas em [CRUZ et al, 2010].

Um fato a destacar neste período foi o salto quantitativo e qualitativo na publicação de livros, artigos técnicos e relatos de experiências Brasileiros sobre o processo de Aquisição de software e serviços correlatos (S&SC). Neste sentido, também poderá ser bastante útil o resultado do projeto do MCT/SEPIN visando selecionar o melhor livro da Série de Livros do PBQP Software dentre os submetidos para o concurso em 2011, que tem como tema a “Melhoria do processo de Aquisição de software e serviços correlatos para órgãos públicos”, para lançamento da publicação em junho de 2011, em Curitiba-PR, no âmbito do Simpósio Brasileiro de Qualidade de Software (SBQS).

No médio e longo prazos são muito promissoras as perspectivas de uma maior disseminação e adoção destas ações no Brasil face às lições aprendidas até agora e às oportunidades decorrentes das mudanças em 2011 tanto no governo federal quanto nos novos governos estaduais.

A partir de 2011, também se apresentam como bastante promissoras as ações decorrentes do uso do processo de Aquisição MPS nos quatro países participantes do novo Projeto BID RELAIS – Rede Latino Americana da Indústria de Software: Brasil, Colômbia, México e Peru.

Referências bibliográficas

ALVES, Angela Maria; GUERRA, Ana Cervigni. **Aquisição de Produtos e Serviços de Software**. Rio de Janeiro, Elsevier, 2004.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - ABNT. **ABNT NBR ISO/IEC 12207:2009 - Engenharia de sistemas e software - Processos de ciclo de vida de software**.

CRUZ, Cláudio Silva; ANDRADE, Edméia Leonor Pereira; FIGUEIREDO, Rejane Maria da Costa. **Processo de contratação de software e serviços correlatos para entes governamentais (PCSSCEG)**. IN: Workshop Anual do MPS (WAMPS 2010), 6., Campinas, SP, SOFTEX, 2010.

GILLIS, Eric Robert; ANDRADE, Edméia Leonor Pereira. **Comparação entre a Instrução Normativa SLTI/MP nº 04 e o Guia de Aquisição do MPS.BR**. IN: Workshop Anual do MPS, 5. WAMPS 2009. Campinas: SOFTEX, 2009.

IEEE STD 1062:1998. IEEE Software Engineering Standards Collection. **IEEE Recommended Practice for Software Acquisition**, IEEE STD 1062 Ed. NY (1998).

Ministério da Ciência e Tecnologia (MCT). Secretaria de Política de Informática (SEPIN). **Programa Brasileiro da Qualidade e Produtividade em Software. Estratégias e ações**. Brasília: MCT, 2009.

Ministério do Planejamento, Orçamento e Gestão (MP). Secretaria de Logística e Tecnologia da Informação (SLTI). **Instrução Normativa SLTI nº 4, de 19 de maio de 2008**. Dispõe sobre o processo de contratação de serviços de Tecnologia da Informação pela Administração Pública Federal direta, autárquica e fundacional. Brasília: SLTI/MP, 2008.

Ministério do Planejamento, Orçamento e Gestão (MP). Secretaria de Logística e Tecnologia da Informação (SLTI). **Instrução Normativa SLTI nº 4, de 12 de novembro de 2010**. Dispõe sobre o processo de contratação de Soluções de Tecnologia de Informação pelos órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática (SISP) do Poder Executivo Federal. Brasília: SLTI/MP, 2010.

SCALET, Danilo; ANDRADE, Edméia Leonor Pereira. **O Guia de Aquisição do MPS.BR e o Pregão Eletrônico**. IN: Workshop Anual do MPS (WAMPS 2009), 5. Campinas-SP, SOFTEX, 2009.

SOFTEX, Sociedade para Promoção da Excelência do Software Brasileiro. **MPS.BR - Melhoria de Processo do Software Brasileiro – Guia de Aquisição:2009**. Campinas: SOFTEX, Junho 2009.

WEBER, Kival Chaves; ARAÚJO, Eratóstenes Edson Ramalho, ROCHA, Ana Regina Cavalcanti; OLIVEIRA, Kathia Marçal; ROUILLER, Ana Cristina; VON WANGENHEIM, Christiane Gresse; ARAÚJO, Renata; SALVIANO, Clenio; MACHADO, Cristina Filipak; SCALET, Danilo; GALARRAGA, Odisnei; AMARAL, Márcio Pecegueiro; YOSHIDA, David. **Melhoria de Processo do Software Brasileiro (MPS.BR): um Programa Mobilizador**. IN: Anais do Congresso Latino Americano de Informática (CLEI), Santiago, Chile, 2006.

WEBER, Kival Chaves; ARAÚJO, Eratóstenes Edson Ramalho; SCALET, Danilo; ANDRADE, Edméia Leonor Pereira; ROCHA, Ana Regina Cavalcanti; MONTONI, Mariano Angel. **MPS Model-Based Software Acquisition Process Improvement in Brazil**. IN: Proceedings of the Sixth International Conference on the

Quality of Information and Communications Technology (QUATIC), Lisbon New University, Lisbon, Portugal, 2007.

Qualidade em Software como Desenvolvimento Sustentável para o Pólo de Software AmazonSoft

Paulino W. P. Viana^{1,2}, Tayana U. Conte³

1Fundação Centro de Análise, Pesquisa e Inovação Tecnológica - FUCAPI
Faculdade FUCAPI – Manaus – AM – Brasil

2SOLTIN – Soluções Integrada em Tecnologia de Informação Ltda.
Manaus – AM - Brasil

3Departamento de Ciência da Computação – Universidade Federal de Amazonas
(UFAM) - Manaus – AM - Brasil

wagner.palheta@fucapi.br, tayana@dcc.ufam.edu.br

Resumo. Este artigo apresenta os resultados e lições aprendidas na implementação do Nível G do MPS.BR em um grupo de empresas de Software participantes do AmazonSoft – Pólo de Software do Amazonas. Esse projeto descreve as principais dificuldades e limitações. Caracterização das empresas, descreve as etapas do projeto e o resultado maior com as três primeiras empresas avaliadas MPS.BR no estado do Amazonas.

1. Introdução

Desde 2003, o PBQP - Programa Brasileiro da Qualidade e Produtividade em Software tem sido difundido na região Norte, tendo como patamar inicial a participação no ciclo de projetos de 2004 com dez projetos da região [4]. Ao longo dos anos, diversas iniciativas foram realizadas, com o objetivo de difundir o PBQP e o modelo MPS.BR [3]. Em alguns anos, Manaus superou diversos municípios em números de projetos submetidos ao PBQP. Isso só denota a preocupação das empresas da região em qualidade de software. Este trabalho apresenta a experiência de um projeto pioneiro de implementação de boas práticas para atender as expectativas do nível G de maturidade do MPS.BR em um grupo de empresas de software incubadas no CIDE – Centro de Incubação e Desenvolvimento Empresarial, que coordena o Pólo de Software do Amazonas denominado AmazonSoft. A próxima seção apresenta o processo de seleção e os critérios adotados para inserir as empresas no grupo. Na seção 2, apresenta os objetivos e justificativos para realizar o projeto. A seção 3 apresenta a metodologia utilizada para selecionar e conduzir o projeto. Na seção 4 são apresentados os resultados do projeto. A seção 5 apresenta a aplicabilidade dos resultados obtidos. Na seção 6 são apresentadas as características inovadoras. E por último são apresentadas as conclusões e perspectivas futuras.

2. Objetivos e Justificativa

O principal objetivo desse projeto foi aumentar a qualidade de software das empresas participantes do Pólo de Software AmazonSoft. As empresas de software incubadas no CIDE - Centro de Incubação e Desenvolvimento Empresarial desempenham um importante papel na busca pela consolidação do Pólo de Software AmazonSoft. A implantação de um programa para qualificar as empresas através de uma implementação de processo e avaliação MPS.BR trará maior visibilidade e credibilidade da qualidade dos produtos desenvolvidos pelas empresas do Pólo de Software. Outro grande desafio, era justamente a ausência de empresas avaliadas em MPS.BR no estado do Amazonas, pois isso é um resultado perseguido desde de 2004 [4].

O planejamento desse projeto para atender as principais metas de implementar e avaliar cinco empresas participante do Pólo de Software, contou com uma metodologia de trabalho bilateral entre o grupo de implementadores e empresas.

3. Metodologia de Execução

Com base nos dados das empresas administrados pelo CIDE, foram identificadas dez empresas de software interessadas em participar do projeto. Entretanto, somente cinco empresas participaram do projeto por questão de recursos financeiros. O CIDE possui importantes informações das empresas e isso possibilitou selecionar as empresas sob o ponto de vista de recursos necessários como contrapartida, condições técnicas, recursos humanos e potenciais chances de sucesso das empresas. O objetivo dessa seleção foi criar um grupo de empresas que tivessem necessidades de melhoria em seus processos e com condições de atender aos requisitos do nível G, para garantir sucesso nas avaliações oficiais. O CIDE definiu dois critérios para selecionar as empresas. O primeiro critério foi a capacidade financeira das empresas para garantir a contrapartida. O segundo critério possuía um peso maior, pois tratava da disponibilidade de alocação de recursos humanos e tecnológicos pelo tempo de duração do projeto por parte das empresas, e também foi observado o comprometimento de apoio da alta direção, nesse caso firmado formalmente.

Essa iniciativa contou com apoio financeiro do CIDE, SEBRAE/AM e com contrapartida das empresas participantes. Os consultores MPS.BR que se encontravam em Manaus pertenciam a duas Instituições Implementadoras diferentes e foram autorizados a trabalharem juntos e de forma colaborativa. A empresa SOLTIN Ltda. ficou responsável pela gestão dos recursos técnicos e financeiros do projeto.

Foram selecionadas cinco empresas para o primeiro projeto de implementação MPS.BR nível G no Amazonas. As atividades de implementação do projeto iniciaram em maio de 2009, com a realização de uma reunião de *kick-off*, com participação dos coordenadores e dos patrocinadores das cinco empresas. O CIDE e a SOLTIN atuaram em conjunto no papel de IOGE (Instituição Organizadora de Grupo de Empresas), a SOLTIN atuou como Instituição Implementadora, o SEBRAE/AM como o maior patrocinador do projeto.

Algumas empresas possuíam experiências em processos de software e metodologias ágeis, mas outras empresas possuíam um processo *ad hoc* de desenvolvimento de software. As empresas desenvolviam softwares para ambiente Web, softwares para plataformas móveis e desenvolvimentos de projetos que envolviam hardware e software. Foi definida a seguinte estratégia para o projeto:

- ? Diagnóstico inicial do processo de desenvolvimento de software da organização;
- ? Planejamento do projeto de implementação nas empresas;
- ? Capacitação e treinamento nos Processos Gerência de Projetos e Gerência de Requisitos;
- ? Consultoria técnica para adaptação/definição dos processos de software das empresas;
- ? Preparação para Avaliação oficial MPS.BR.

O projeto seguiu conforme o planejado, mas por questões de agendamentos das Avaliações oficiais, as empresas tiveram um pouco mais de tempo para institucionalizar seus processos. As avaliações oficiais foram contratadas seguindo as determinações do MA-MPS.

4. Resultados Relevantes

O projeto de implementação do nível G do MPS.BR nas empresas do AmazonSoft proporcionou grandes resultados. Não somente pela disseminação do modelo MPS.BR no Amazonas mas principalmente na construção de uma nova cultura, pois empresas de software passaram a acreditar na importância em ter qualidade em seus processos.

Durante o projeto foram capacitadas 51 pessoas através dos treinamentos dos Processos Gerência de Requisitos e Gerência de Projetos. Esses treinamentos ocorreram por duas vezes, pois algumas empresas tiveram problemas de rotatividade de seus colaboradores. Como esse movimento do MPS.BR foi tão intenso no estado, na ocasião da realização do Curso oficial de Introdução do MPS.BR, item obrigatório para as empresas que seriam avaliadas, houve a participação de 17 pessoas, sendo que 10 pessoas possuíam suas inscrições garantidas pelo projeto. Portanto houve 70% de inscrições extras devido a divulgação e motivação pela busca de conhecimento sobre MPS.BR. Dando continuidade na formação de conhecimento do modelo MPS.BR, na ocasião da realização da Prova Oficial P1, foram aprovadas cinco pessoas no estado.

O projeto possibilitou a realização de dois trabalhos acadêmicos que resultaram em publicações. O primeiro tratava do diagnóstico inicial para avaliação em uma das empresas [2]. O segundo trabalho aplicou *Grounded Theory* em duas empresas para compreender os fatores críticos de sucesso em iniciativas de melhoria de processo de software [1]. Além disso, esse projeto foi submetido ao ciclo de 2009 e com continuidade para o ciclo de 2010 no PBQP - Programa Brasileiro de Qualidade de Software.

As avaliações oficiais foram contratadas, mas somente três empresas conseguiram ter seus projetos avaliados. As outras duas empresas tiveram problemas para concluir seus projetos e não foram avaliadas. Portanto como resultado mais importante da iniciativa de criar um projeto para implementar do nível G do MPS.BR nas empresas do AmazonSoft foi a avaliação bem sucedida das três primeiras empresas do Amazonas no modelo MPS.BR.

5. Aplicabilidade dos Resultados

Os resultados obtidos pelas empresas geraram muitas satisfações para os empresários, pois todos os colaboradores ficaram mais envolvidos e comprometidos com as ações estratégicas das empresas. Os clientes perceberam o impacto das mudanças, pois tiveram que renegociar requisitos e prazos, e com isso observaram o grau de melhoria das negociações. As pesquisas realizadas foram publicadas e isso aumenta a disseminação dos conhecimentos gerados dentro do projeto, incentivando a novos pesquisadores se aprofundarem nos temas específicos.

6. Características Inovadoras

O impacto desse projeto foi muito bom, pois não existia nenhuma empresa avaliada em MPS.BR no Amazonas. Com o principal resultado do projeto alcançado, as empresas poderão prospectar novos negócios e aumentar seus ganhos e continuar investindo em melhoria de processo de software. Com a articulação do CIDE em busca de parceiros para viabilizar esse projeto, a participação do SEBRAE/AM foi fundamental, pois a instituição portou 70% dos recursos necessários. Essa foi a primeira parceria entre as instituições com objetivos comuns no âmbito de MPS.BR no Amazonas. Apesar de que somente três empresas foram avaliadas no Nível G do MPS.BR, todos os parceiros ficaram satisfeitos com o resultado final.

7. Conclusões e Perspectivas Futuras

Os desafios foram enormes nesse projeto, podemos citar e assumir como erro inicial de planejamento, não foi exigido o comprometimento formal dos principais patrocinadores do projeto. Iniciamos o projeto contando somente com a contrapartida das empresas para operacionalizar as primeiras ações do projeto. Isso gerou certos desconfortos, pois os maiores aportes só foram disponibilizados posteriormente e com isso tivemos que replanejar as principais ações, como treinamento oficial do MPS.BR e contratação das avaliações.

Apesar de ter sido maciçamente divulgado durante o projeto que uns dos principais objetivos de grupo de empresas são compartilhar recursos e promover a troca de experiências entre as empresa, sempre foi visto com certa desconfiança essa troca de experiência, pois em alguns casos as empresas são concorrentes e isso poderia prejudicar contratos futuros.

Conforme visto em [1], um dos maiores problema por parte dos colaboradores das empresas foi o desconhecimento da principal importância de implementar melhoria de processo de software pela melhoria contínua e não puramente por uma certificação.

Outro ponto importante detectado na pesquisa foi que, a falta de comprometimentos e motivação colabora para realizar atividade de maneira *ad hoc*.

Com o objetivo de viabilizar o projeto e foi proporcionado o valor muito baixo de contrapartida para as empresas e isso gerou um enorme problema para a gestão do projeto. Dessa forma não conseguimos aplicar mais horas de consultoria durante o projeto. Isso foi analisado e identificado como um fator muito importante, pois não se pode esperar que as empresas possuam conhecimento prévio sobre boas práticas de melhoria de processo de software. Tal conhecimento faz parte da *expertise* dos consultores e deve ser passado nas horas de consultoria. Todos esses problemas detectados foram registrados e com essa experiência nos facilitará no planejamento das estratégias, buscando maior eficiência nos futuros projetos de implementação no Amazonas.

Este artigo apresentou a experiência de implementação do MPS.BR nível G em um projeto com cinco empresas que obteve como resultado máximo três empresas avaliadas no Amazonas. As dificuldades encontradas durante a implementação do projeto de implementação nas empresas do grupo, serviram como pontos de melhorias para nossa metodologia de trabalho. Esses pontos de melhorias identificados serão aperfeiçoados para buscarmos uma maior maturidade em nossos futuros projetos MPS.BR.

A parceria com o CIDE e SEBRAE-AM foram fundamentais para viabilizar o projeto e acreditamos que tal projeto só justificou os compromissos firmados dessas Instituições, no que diz respeito em apoiar o desenvolvimento regional. Caracterizado dessa forma como o principal retorno de investimento, as empresas foram avaliadas e trouxe maior visibilidade nacional para as empresas de software do Amazonas.

Identificamos a necessidade de criação de uma Instituição Implementadora credenciada pela SOFTEX para viabilizar novos projetos. A carência de recursos humanos também é uma ação estratégica, pois devemos garantir essa formação contínua de novos consultores na região.

8. Referências

- [1] MATOS JR., O.; SECATTI, V.; SANTOS, D. V.; OLIVEIRA, H.; CONTE, T. U.; *Aplicando Grounded Theory para Compreender os Fatores Críticos de Sucesso em Iniciativas de Melhoria de Processo de Software*. In: VI Workshop "Um Olhar Sociotécnico sobre a Engenharia de Software" (WOSES 2010), 2010, Belém. Anais do VI WOSES – IX Simpósio Brasileiro de Qualidade de Software.
- [2] SANTOS, D. V.; CONTE, T. U.; VIANA, P. W. P.; XAVIER, C.; *Diagnóstico inicial para Avaliação MPS.BR Nível G em Empresa do Grupo AmazonSoft*. III Mostra Técnico-Científica. FIAM – Feira Internacional da Amazônia, 2008.
- [3] SOFTEX, *MPS.BR-Melhoria de Processo do Software Brasileiro – Guia Geral*, Versão 1.2, 2007. Disponível em: <http://www.softex.br/mpsbr>.
- [4] VASCONCELLOS, F. J. S., *Qualidade de Software na Região Norte*. IV SBQS – PBQP Software Ciclo 2004 – Projetos Premiados, 2005.

Curso de Residência em Arquitetura de Software: Uma Parceria UFPA, UNAMA e CESUPA com AmazonCorp, CTIC-UFPA e PRODEPA

Carla Alessandra Lima Reis¹, Claudio Roberto de Lima Martins², Marcos Venícios Araújo³, Rodrigo Quites Reis¹

¹Laboratório de Engenharia de Software - Universidade Federal do Pará (UFPA)

²Instituto Federal do Pará (IFPA)

³Centro Universitário do Estado do Pará (CESUPA)

clima@ufpa.br, cmartins@ifpa.edu.br, maraujo@cesupa.br, quites@ufpa.br

Abstract. *This paper presents the experience on planning, enactment and effect of the Residency on Software Architecture accomplished at the region of Belém, state of Pará, Brazil. This course resulted in the improvement of 13 professionals allocated to 3 different companies in projects mainly related to software reuse, reengineering, and interoperability.*

Resumo. *Este artigo descreve a experiência no planejamento, condução e repercussão do Curso de Residência em Arquitetura de Software ocorrido entre 2008 e 2010 na região metropolitana de Belém, estado do Pará. O curso tratou da formação de 13 profissionais em nível de especialização que atuaram em projetos de 3 empresas da região que trataram de temas como reuso, reengenharia e interoperabilidade de software.*

1. Introdução

Atualmente mais e mais áreas de conhecimento e setores produtivos da humanidade são influenciadas decisivamente por software especializado. Com o aumento do poder de processamento dos computadores e a crescente disponibilidade de equipamentos, cresce a demanda por software especializado para uma grande variedade de contextos. Além disso, um percentual significativo dos esforços é voltado para as atividades-fim das organizações-usuárias, o que contribui para um aumento da complexidade dos produtos e elevação dos requisitos como segurança, concorrência, distribuição e escalabilidade.

A **Engenharia de Software** é a área da Ciência da Computação voltada ao estabelecimento de técnicas, métodos, processos e ferramentas voltadas tanto ao aumento da qualidade do software produzido quanto à melhoria da produtividade da indústria. Entre as inúmeras iniciativas da área, destaca-se a crescente preocupação com aspectos como segurança, disponibilidade e escala dos sistemas construídos. Mesmo sistemas desenvolvidos por grandes fabricantes internacionais possuem falhas deste tipo. Tais questões são decorrentes com frequência de negligências com a elaboração da Arquitetura, na fase costumeiramente chamada de **Projeto (ou Design) de Software**.

A elaboração e evolução da arquitetura de sistemas de software constituem elementos diferenciadores na qualidade dos produtos da atualidade. Isto ocorre porque no projeto da arquitetura são respondidas as questões relacionadas com, por exemplo, a segurança, escalabilidade, usabilidade, portabilidade, e tempo de resposta das aplicações. Entretanto, há uma notória deficiência na formação de profissionais com este perfil no país, o que contribui com a tímida participação do país na construção nos sistemas atuais da Web 2.0. Em paralelo a isto, há enorme demanda para a evolução e interoperabilidade de sistemas legados para atender novos requisitos, o que também exige a participação de profissionais com perfil diferenciado.

A experiência pioneira do CIn-UFPE (Centro de Informática da Universidade Federal de Pernambuco) na criação dos cursos de especialização na modalidade de Residência em Software se mostrou uma alternativa interessante no aperfeiçoamento de profissionais seguindo a prática comum na Medicina. Vencedor do Prêmio Dorgival Brandão Junior de 2005 do Programa Brasileiro de Qualidade e Produtividade em Software do MCT, a parceria do CIn-UFPE com a Motorola pavimentou o caminho para investimentos do Governo Federal para disseminar esta prática em diferentes locais do país. Em sequência a experiência da UFPE, o Edital MCT/CNPq/CT-Info nº 001/2008 - Residência em Desenvolvimento de Software 2008 (CNPq, 2008) permitiu a submissão de propostas para apoio financeiro a cursos na modalidade de Residência em Software por instituições de todo o território nacional.

Este artigo descreve a experiência no planejamento, condução e repercussão do Curso de Residência em Arquitetura de Software ocorrido entre 2008 e 2010 na região metropolitana de Belém, estado do Pará. O curso tratou da formação de 14 profissionais em nível de especialização na área de Arquitetura de Software que atuaram em projetos de três empresas da região que trataram de temas como reuso, reengenharia e interoperabilidade de software.

2. Objetivos e Justificativa

O Pará, em função das suas dimensões geográficas e de ser o estado mais populoso de toda região norte, impõe desafios de dimensões amazônicas para o desenvolvimento socioeconômico da sua população, além de propiciar a expansão sustentável do seu parque industrial que é fortemente baseado na exploração dos

recursos naturais. Ao redor deste cenário, o software utilizado tem a responsabilidade em lidar com problemas de instabilidade em conexões de rede e precariedade no fornecimento de energia comuns no interior do estado. Do ponto de vista do setor público, as soluções de TI no Pará tem a obrigação de reduzir as distâncias regionais e reforçar a presença do Estado.

O objetivo geral deste projeto era o de contribuir com o aperfeiçoamento da formação de profissionais que possam atuar na Elaboração, Avaliação e Melhoria de Arquiteturas para Sistemas de Software. O tema “Arquitetura de Software” foi rapidamente identificado como prioritário nas reuniões preliminares de planejamento do curso – detalhadas na seção 3, a seguir – em função da escassez de pessoal qualificado na região e da grande demanda em projetos das empresas do Pará.

Constituíram objetivos específicos deste projeto:

- ? **Contribuir com a formação de profissionais na área.** Atualmente, há notória carência de profissionais que tenham formação teórica e prática na área de Arquitetura de Software em todo país. Isto decorre da tradicional baixa carga horária disponível nos cursos de Graduação para o estudo da Arquitetura de sistemas de alta complexidade, além da costumeira indisponibilidade de carga horária para a concepção de sistemas com grande exigência de requisitos não funcionais e de arquitetura “em sala de aula”.
- ? **Formar nova geração de especialistas em Engenharia de Software.** O setor de software é uma área de interesse estratégico na Política Industrial do país, com enorme demanda de profissionais capacitados. Além disso, o projeto busca contribuir com a formação de profissionais em uma das áreas críticas na atualidade, com carência em todo o mundo (Downey, 2008).
- ? **Promover a transferência dos resultados do projeto para a indústria e sociedade.** O projeto prevê o estágio formal dos residentes em projetos reais desenvolvidos nas empresas envolvidas.
- ? **Apoiar no Incremento de Maturidade das Empresas Envolvidas.** As práticas de Engenharia de Software e, mais especificamente, as atividades relacionadas com a definição e evolução de Estilos Arquiteturais constituem elementos requeridos na avaliação do modelo Melhoria do Processo de Software Brasileiro (MPS.BR) (Softex, 2011) para o **Nível D - Largamente Definido**. Especificamente, o processo “**PCP - Projeto e Construção do Produto**” lida com os detalhes da definição da arquitetura e seleção de componentes de software. No momento em que o projeto foi iniciado, a PRODEPA possuía uma avaliação vigente do nível F, enquanto que o Centro de Tecnologia da Informação e Comunicação da UFPA (CTIC-UFPA) encontrava-se na implantação do nível G (avaliado formalmente em 15/12/2008). Portanto, este era um alinhamento entre a proposta do curso e as iniciativas das empresas em perseguir um aumento contínuo da sua maturidade.

No ano de 2010 houve a concentração das defesas dos trabalhos de conclusão de curso. Especificamente, os objetivos em 2010 foram voltados a realização das defesas dos trabalhos concluídos pelos residentes, além da avaliação dos resultados obtidos.

3. Metodologia de Execução

Este projeto surgiu a partir de um levantamento informal realizado na Secretaria de Desenvolvimento, Ciência e Tecnologia (SEDECT) do estado do Pará que indicou a demanda pelo tema por parte da Indústria e, simultaneamente, a capacidade instalada de Mestres, Doutores e Grupos de Pesquisa nas Instituições de Ensino Superior do Pará. Reuniões foram realizadas na SEDECT – como ilustrado pela fotografia da figura 1 - com o objetivo de se discutir o modelo ideal a ser seguido.

Desde o início (2008) se identificou que para atingir os objetivos de se maximizar o alcance deste projeto seria necessária uma articulação envolvendo instituições de ensino superior e empresas interessadas. Assim, uniram-se a UFPA, CESUPA e UNAMA como instituições acadêmicas responsáveis por executar o curso – isto é, oferecer docentes, orientadores e salas de aula, ficando o curso formalmente sediado na primeira instituição, sob coordenação do Professor Rodrigo Quites Reis.

O projeto foi submetido ao CNPq considerando a participação da PRODEPA, CTIC-UFPA e AmazonCorp, empresas onde os residentes atuaram. O orçamento do projeto submetido ao CNPq foi composto da seguinte forma:

- ? Apoio do CNPq no pagamento de bolsas DTI-3 a 14 alunos (valor de R\$ 1.045,89) por 12 meses
- ? Contrapartida financeira de todas as empresas envolvidas com valor total de R\$ 137.523,84. Esta contrapartida foi usada para aquisição de um notebook por aluno selecionado, material bibliográfico, consumíveis de impressão entre outras despesas com o curso. A distribuição dos bolsistas era proporcional à contrapartida das empresas, cabendo inicialmente a PRODEPA treze residentes, ao CTIC-UFPA dois residentes, e à AmazonCorp dois residentes.



Figura 1 Reunião na SEDECT com representantes do setor de TIC do estado do Pará para início do planejamento do curso

Com a aprovação do projeto por parte do CNPq, houve início da divulgação do curso através da imprensa local, mídia impressa e um evento realizado no Hangar – Centro de Convenções da Amazônia - patrocinado pela PRODEPA (em maio de 2008).

O processo seletivo foi realizado em três etapas¹²: 1) prova escrita (sobre temas de Engenharia de Software e Programação de Computadores) – etapa eliminatória e classificatória; 2) análise de currículo – classificatória e eliminatória; e 3) entrevista (contando com a participação de representantes das empresas – etapa classificatória. Ao final do processo, uma demanda qualificada de 17 alunos foi selecionada.

Como a demanda qualificada excedia a quantidade de bolsas disponíveis, a Coordenação do curso buscou uma parceria com a FAPESPA - Fundação de Amparo a Pesquisa do Estado do Pará que custeou duas bolsas adicionais para alunos selecionados para realizar a residência na PRODEPA. O curso foi iniciado em setembro de 2008 e contou com as disciplinas relacionadas nas tabela 1.



Figura 2 Entrega dos notebooks para os alunos do curso 1 Conforme edital disponível em <http://www.labes.ufpa.br/residencia>

² 1 Conforme edital disponível em <http://www.labes.ufpa.br/residencia>

Tabela 1 Disciplinas do Curso

Nome da disciplina	Carga Horária	Assuntos tratados	Ministrantes
Introdução a Engenharia de Software	60	Visão geral do processo e qualidade de software	Carla A. Lima Reis (UFPA)
Engenharia de Requisitos para Arquitetura de Software	30	Especificação e levantamento de requisitos funcionais, não funcionais, arquiteturais e de qualidade	Marcos Venícios Araújo (CESUPA)
Paradigmas de Programação de Computadores	60	Teoria e Prática dos Paradigmas de Objetos, Desenvolvimento baseado em Componentes, e Orientação a Aspectos	Rodrigo Quites Reis (UFPA)
Projeto de Software com UML	60	UML: histórico, diagramas, e extensões. Projeto de Banco de Dados.	Fábio Bezerra (UNAMA/CESUPA)
Manutenção e Reengenharia de Software	30	Refatoração. Manutenção de sistemas legados.	Claudio Martins (UNAMA)
Gerência de Projetos e Arquitetura de Software	30	Gestão de Projetos. Projetos Iterativos. RUP: Elaboração e Fluxo de Análise e Projeto.	Carla A. Lima Reis (UFPA)
Ferramentas e Arquitetura de Software	30	Classificação de ferramentas. IDEs e CASE.	Ernani Sales (UFPA)
Reutilização de Software	30	Padrões de Projeto e Implementação. Frameworks: construção e uso.	Fábio Bezerra (CESUPA)
Projeto de Software de Alta Complexidade	30	Melhores práticas. Avaliação Arquitetural pelas técnicas SAAM e ATAM	Rodrigo Quites Reis (UFPA)
Tópicos Avançados em Arquitetura de Software	30	Linguagens de Descrição Arquitetural, Arquitetura Centrada em Modelos, e Linhas de Produto de Software	Rodrigo Quites Reis (UFPA)
Inglês Instrumental para o Setor de Software	60	Leitura e prática de textos científicos e técnicos na área de desenvolvimento de software	Breno França (UFPA)
Metodologia Científica	30	Produção de conhecimento. Redação de textos científicos	Larissa Sato (CESUPA)
	480		

4. Resultados Obtidos

Os resultados obtidos com o curso são apresentados nas subseções a seguir. Primeiro, na subseção 4.1 serão discutidos os Trabalhos de Conclusão de Curso produzidos pelos residentes. Em seguida, é apresentado um panorama sobre o destino dos egressos do curso. Dos 17 alunos que iniciaram o curso, 13 concluíram e defenderam suas monografias de final de curso no período entre junho a dezembro de 2010.

4.1 PRODEPA

Os nove residentes da PRODEPA desenvolveram os trabalhos relacionados na Tabela 2. Em virtude das características da empresa, grande parte dos trabalhos foram relacionados com o tema Reutilização de Software: há propostas de bibliotecas e *frameworks* de propósito geral, um processo padrão para gestão de reuso – alinhado aos itens requeridos do MPS.BR, e uma biblioteca para geoprocessamento. Também foram tratados problemas na área de segurança e interoperabilidade de software.

Tabela 2 Trabalhos de Conclusão de Curso desenvolvidos na PRODEPA

Residente	Título	Orientador	Tema
Alison Ricardo	Proposta de Arquitetura para Sistemas de Segurança de Aplicações WEB Corporativas.	Prof. Rodrigo Quites Reis	Segurança

		(UFPA)	
Euclides de Mello Lopes Netto	Uma Proposta de Programa de Reutilização e Métricas de Software para a PRODEPA	Prof. Marcus Venícius Araújo (CESUPA)	Reutilização de Software
Leonardo Possamai Mezzomo	Integração da ferramenta JASPERSERVER com aplicações da PRODEPA.	Prof. Marcos Venicius (CESUPA)	Interoperabilidade / Reengenharia de Software
Marcos Filipe Alves Salame	Proposta para implantação de SOA com ESB auxiliada por uma metodologia para facilitar a interoperabilidade dos sistemas da PRODEPA.	Prof. Rodrigo Quites Reis (UFPA)	Interoperabilidade
Patrícia Matias Lopes	Proposta de um processo de gerência de reutilização para a PRODEPA.	Prof. Carla Lima Reis (UFPA)	Reutilização de Software
Paulo André Pereira de Miranda	Especificação para o Desenvolvimento de Componentes Orientados a Objetos no Âmbito da PRODEPA.	Prof. Sandro Bezerra (UFPA)	Reutilização de Software
Simon Corrêa	Arquitetura portátil para desenvolvimento de software Orientado a Objeto.	Prof. Sandro Bezerra (UFPA)	Reutilização de Software
Victor Rafael Coutinho Pereira	PROJETO MOISÉS: Uma proposta de divisão da biblioteca MUIRAQUITÃ.	Prof. Cláudio Martins (UNAMA)	Reutilização de Software
Ygor Oliveira	Elaboração de proposta de arquitetura para sistemas de Geoprocessamento	Prof. Fábio Bezerra (CESUPA)	Reutilização de Software

4.2 CTIC-UFPA

Os trabalhos desenvolvidos no CTIC-UFPA são apresentados na Tabela 3. Aqui, um trabalho foi orientado aos esforços de propor um processo para gerência de reutilização de software, enquanto que o segundo apresentou um barramento lógico (*service bus*) para prover interoperabilidade entre sistemas da organização.

Tabela 3 Trabalhos de Conclusão de Curso desenvolvidos no CTIC-UFPA

Residente	Título	Orientador	Tema
Daniel Dias de Carvalho	Proposta de um Processo de Gerência de Reutilização para o CTIC-UFPA: Uma Abordagem Arquitetural	Profa. Carla Alessandra Lima Reis (UFPA)	Reutilização de Software
Liken Iu Matsumoto de Lima	CTIC-ESB: Proposta de uma Infraestrutura Para Facilitar a Integração entre as Aplicações Web do CTIC	Prof. Rodrigo Quites Reis (UFPA)	Interoperabilidade de Software

4.3 Amazon Corp

Os trabalhos desenvolvidos na AmazonCorp são apresentados na Tabela 4.

Tabela 4 Trabalhos de Conclusão de Curso desenvolvidos na AmazonCorp

Residente	Título	Orientador	Tema
Murilo Farias Sales	Modelo Arquitetural para Interoperabilidade de Informações de Projetos de Desenvolvimento de Software	Prof. Rodrigo Quites Reis (UFPA)	Interoperabilidade de Software
Silvio Bruno Paiva Costa	Uma Experiência de Adaptação Arquitetural no contexto de um Projeto de Software da Amazon Corporation	Prof. Sandro Ronaldo Bezerra Oliveira (UFPA)	Reengenharia de Software

5. Aplicabilidade dos Resultados

A aplicação dos resultados desse projeto é analisada segundo dois pontos de vista complementares: 1) a contribuição efetiva dos trabalhos desenvolvidos pelos residentes nas empresas envolvidas no projeto; 2) a contribuição do curso na formação dos egressos.

É notória a importância da interação dos residentes com os projetos em andamento das empresas envolvidas. As empresas foram unânimes em descrever que os projetos desenvolvidos contribuíram para um aumento da densidade. A Figura 3 apresenta o destino dos egressos. Todos os egressos atuam em desenvolvimento de software. Dos treze alunos que concluíram, sete cursam o Mestrado em Ciência da Computação da UFPA (quatro em Dedicação Exclusiva). A maioria dos egressos que estão no Mercado atuam em empresas públicas, entre as quais: PRODEPA, Banpará – Banco do Estado do Pará, e Cobra Tecnologia S.A.

6. Características Inovadoras

Três aspectos podem ser destacados como inovadores neste projeto.

A Gestão do Curso ocorreu de forma compartilhada entre as instituições de ensino superior e Empresas. Workshops foram realizados de forma bimestral de forma itinerante entre as três empresas para dar visibilidade dos trabalhos que estavam sendo desenvolvidos entre todos os envolvidos. Do ponto de vista acadêmico, as três instituições ficaram responsáveis pelos docentes, instalações para ocorrerem as aulas, além de fornecer os orientadores para o Curso.

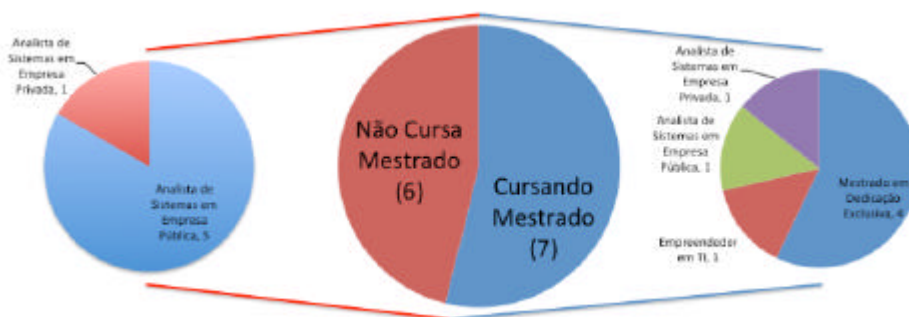


Figura 3 Destino dos Egressos do Curso de Residência (dados de fevereiro de 2011)

O segundo aspecto a destacar diz respeito à grade curricular. Foram apresentados assuntos específicos de Arquitetura de Software – tal como Projeto de Software de Alta Complexidade, Métodos de Avaliação Arquitetural (SAAM e ATAM), e assuntos que tratam da questão transdisciplinar da Arquitetura de Software com outros temas de Engenharia de Software – tais como as disciplinas em “Gerência de Projetos e Arquitetura de Software”, sem conhecimento dos autores acerca de alguma iniciativa semelhante no país em nível de pós-graduação *latu sensu*.

O terceiro aspecto inovador refere-se à natureza do curso de residência em si. Sendo Arquitetura de Software reconhecida por Downey (2008) por ser um assunto que demanda um conjunto específico de habilidades muito difícil de ser obtido apenas com qualificações acadêmicas e/ou experiência profissional, o curso de Residência em Arquitetura de Software buscou unir os dois “mundos”.

7. Conclusão e Perspectivas Futuras

Há uma clara expectativa da reedição do curso em uma segunda turma. Além disso, o modelo de formação em Residência vem sendo adotado em um projeto da UFPA para formação de especialistas na área de desenvolvimento de software para processamento de alto desempenho de forma que se consiga obter massa crítica para desenvolvimento de soluções que explorem adequadamente o Centro Paraense de Computação Distribuída e de Alto Desempenho (em implantação).

8. Referências

CNPq. (2008) Edital MCT/CNPq/CT-Info nº 001/2008 - Residência em Desenvolvimento de Software 2008. Disponível na Internet por [www em http://www.cnpq.br/editais/ct/2008/001.htm](http://www.cnpq.br/editais/ct/2008/001.htm) (acesso em 2008).

Downey, J.; Babar, M.A. (2008) On identifying the skills needed for software architects. Proceedings of the first International Workshop on Leadership and Management in Software Architecture. ACM Press, 2008.

Softex. (2011) Melhoria do Processo de Software Brasileiro – Guia Geral. Disponível na Internet por [www em http://www.softex.br/mpsbr](http://www.softex.br/mpsbr) (acesso em 2011).

O Uso de PBL no Ensino de Engenharia de Software

Simone C. dos Santos, Jones O. Albuquerque, Silvio R. L. Meira

Centro de Estudos e Sistemas Avançados do Recife (C.E.S.A.R)

Caixa Postal 15.064 - 91.501-970 - Pernambuco - PE - Brazil

simone@cesar.org.br, jones.albuquerque@gmail.com, silvio@cesar.org.br

Abstract. *The objective of this article is to describe an innovative pedagogical methodology based on PBL (Problem Based Learning) to improve the learning effectiveness in software engineering, through the implementation of software factories, where students can work together to solve real problems. This methodology is running in a graduate course in Software Engineering managed by C.E.S.A.R (<http://www.cesar.org.br>), with key contributions to market and society.*

1. Introdução

O crescimento do mercado de Tecnologia da Informação (TI) e seu papel cada vez mais importante na criação de vantagem competitiva das organizações têm gerado uma forte demanda por novas e diferentes maneiras de se produzir software com alta qualidade, flexibilidade e ganhos de produtividade e, conseqüentemente, uma forte demanda por profissionais qualificados para conduzir este processo. Com esta motivação, o C.E.S.A.R (Centro de Estudos e Sistemas Avançados do Recife) criou o Mestrado Profissional em Engenharia de Software – MPES com o objetivo de capacitar líderes em suas áreas de atuação com ênfase em práticas e métodos inovadores voltados ao desempenho na área de Engenharia de Software. A proposta do MPES é desenvolver uma visão ampla do processo de desenvolvimento de software e de todos os recursos envolvidos (Pessoas, Processos e Infraestrutura), além do entendimento de como aplicar este conhecimento para resolver problemas reais com significativa complexidade de maneira inovadora.

O MPES é organizado em duas etapas. A primeira etapa, *vivência de fábricas de software*, inicia com a seleção de projetos reais do mercado, apresentados aos alunos como potenciais aplicações a serem desenvolvidas em suas fábricas. Cada fábrica escolhe um projeto e define seu negócio com base neste projeto. Novos papéis além de professores/tutores são necessários, garantindo o acompanhamento do uso sistemático de processos tanto para a criação das fábricas, quanto para o desenvolvimento de software conduzido em seus projetos. Uma metodologia de ensino baseada em PBL (*Problem-Based Learning*) [1] criada no MPES permite trabalhar os elementos necessários para a realização desta etapa, como descreve a Seção 3. Esta metodologia é executada por professores e monitores que atuam como profissionais seniores em suas áreas de conhecimento.

A segunda etapa (*Pesquisa e Desenvolvimento*) é conduzida por um processo de acompanhamento específico, em sincronia com a disciplina Metodologia Científica. O trabalho final do curso pode ainda adotar os seguintes formatos: *dissertação* (problema real resolvido por soluções fundamentas em pesquisa), *produto* (foco em desenvolvimento de software ou processo) e *coletânea de artigos* (artigos aceitos em congressos de relevância de mercado). Vale ressaltar que o MPES foi criado com ênfase em três linhas de pesquisa (Fábrica de Software, Reuso e Qualidade), entretanto, já na execução da 2ª. turma, a linha de pesquisa “Educação em TIC” foi inserida no programa, tendo como principal motivação a evolução contínua da qualidade da metodologia de ensino adotada.

Finalmente, o MPES iniciou em agosto de 2007, com uma turma de 18 alunos.

Passados 3 anos de existência, hoje o MPES contabiliza 72 alunos em 5 turmas, 32 mestres (com uma previsão de mais 18 mestres até o final deste ano, totalizando 50 mestres), 14 fábricas de software gerando ao menos 1 aplicação de software por fábrica, 4 produtos, produções científicas publicadas em conferências e periódicos nacionais e internacionais, parcerias com uma dezena de empresas de mercado e da academia, e atração de alunos de vários estados brasileiros como Alagoas, Paraíba, Rio Grande do Norte, Sergipe, Maranhão, Tocantins, Goiás, Mato Grosso e Paraná, além do estado de Pernambuco, na qual reside a sede do C.E.S.A.R.

2. Objetivos e Justificativa

O objetivo deste projeto é a formação de capital humano em Engenharia de Software alinhada a demandas reais de Mercado, tendo como premissa a qualidade e produtividade de software. Como objetivos específicos para o Ciclo 2010 destacam-se:

(1) Formação de capital humano de forma a atender às atuais exigências de qualificação do mercado de trabalho da indústria de software; (2) Articulação integrada da formação profissional com entidades demandantes do mercado, visando melhorar a eficácia e eficiência de organizações por meio da solução de problemas reais e melhorias de processos com uso da Tecnologia da Informação; (3) Aplicação de conceitos, métodos e técnicas da qualidade e produtividade de software em ambientes de desenvolvimento de software

usados como instrumento de ensino, bem como na avaliação de resultados proporcionados por estes instrumentos; (4) Formação de mestres em Engenharia de Software, com visão ampla sobre o processo de desenvolvimento sistematizado e eficaz, comprovadas por produção científica relevante.

3. Metodologia de Execução

Para a implementação do método PBL, o MPES provê um ambiente baseado em Fábricas de Software, no qual os estudantes estão imersos em projetos práticos de desenvolvimento de software de forma sistemática e controlada por processos de qualidade, tendo como referência o modelo de maturidade CMMI (*Capability Maturity Model Integration*).

A metodologia identifica quatro grupos de capital humano envolvidos no processo e criação das fábricas de software: o *time* que compõe uma fábrica, responsável pela definição do negócio da fábrica e desenvolvimento de software; os *tutores/professores*, profissionais seniores de mercado em sua área de conhecimento com formação e experiência acadêmica; os *monitores de fábrica*, responsáveis pelo acompanhamento de todas as fábricas, sob os pontos de vista de processos e produtos; os *clientes*, responsáveis pela demanda do software, validação de requisitos e dos resultados acordados com as fábricas de software.

Quanto ao conteúdo, tendo como base as disciplinas do SWEBOK (Software Engineering Body of Knowledge – <http://www.swebok.org/>), uma iniciativa do IEEE Computer Society, o MPES trabalha com sete disciplinas durante a etapa da vivência das fábricas de software: Fábrica de Software, Requisitos e Interface com o Usuário, Gestão de Projetos, Desenvolvimento Orientado à Arquitetura, Engenharia de Reuso, Interoperabilidade e Verificação e Validação.

Quanto aos processos criados, a metodologia contempla um processo para a construção das Fábricas de Software descrito em [2], envolvendo desde a definição de equipes, modelo de ciclo de vida e de negócios, a detalhes relacionados a sistema e infraestrutura necessários ao seu funcionamento. Além disso, um processo para o controle e acompanhamento dos projetos de software foi criado, por meio de avaliações de planejamento, artefatos gerados, evolução e satisfação do cliente [3]. A área de processo *PPQA - Process and Product Quality Assurances* do CMMI é usada como referência, garantindo que a execução dos processos e a construção dos produtos nas fábricas estejam sendo realizadas de acordo com o planejado. Visando um maior alinhamento entre os processos de software e de negócios, na edição da turma 2011.1 foi inserido como melhoria uma avaliação referente ao *Plano de Negócio*, contendo indicadores sobre os objetivos da fábrica, entendimento do problema, avaliação do produto, concorrência, precificação e retorno financeiro. Vale salientar que avaliações individuais são realizadas em cada disciplina ministrada. Vale ressaltar que a avaliação do aprendizado é conduzida de forma que o trabalho coletivo e as habilidades individuais sejam complementares e reflitam diretamente no desempenho do estudante.

Portanto, o desempenho final de cada aluno é calculado pela composição de atividades coletivas (60%) e atividades individuais (40%).

Um pool de projetos práticos de mercado é criado para cada turma que se inicia.

Visando uma maior proximidade e cooperação com o mercado local de software durante a execução da turma 2010.1, o MPES publicou um edital de chamada de projetos de software a empresas locais, conseguindo seis submissões:

- 1) Projeto Gestão Integrada de Serviços de Próteses Odontológicas (Reabilitação Oral). Empresa BIOPAR Participações Ltda, empresa holding do Grupo OT;
- 2) Projeto ODIN.MED – Engenharia Clínica On line, empresa Engebio Serviços Técnicos de Engenharia LTDA.
- 3) NoMaze – Rede Social de Apoio à Decisão Estratégica, Valença & Associados – Aprendizagem Organizacional (V&A)
- 4) Sistema ERP DGVGR - Visão Gerencial de Resultados, Datagold Assessoria e Informática Ltda.
- 5) Framework para jogos 3D em iPhone, empresa MeanTime;
- 6) Framework para jogos 3D em Android, empresa MeanTime.

Os projetos foram submetidos usando um modelo padrão de RFP (*Request For Proposal*). Dos seis projetos apresentados à turma, três deles (itens 3, 5 e 6) foram selecionados e executados pela fábricas de 2010.1, gerando dois produtos para o mercado de desenvolvimento de aplicações para dispositivos móveis. Vale ressaltar que um projeto, para participar do pool, deve cumprir alguns requisitos: deve ter características de inovação; relevância para a indústria; modelo de negócios aplicado e; ter um processo de desenvolvimento de software que permeie todo o ciclo de desenvolvimento. Os projetos precisam ainda ter um cliente real envolvido. Os clientes motivam os estudantes a se envolverem no projeto para a solução dos problemas.

4. Resultados Obtidos

4.1. Recursos humanos capacitados

Até o momento o MPES capacitou 72 estudantes, com alto nível de satisfação do corpo discente, medida por meio de avaliações em cada disciplina e de avaliações institucionais. Desde a 1ª. turma, 100% dos alunos do MPES que concluíram a etapa de disciplinas responderam SIM para as perguntas *"Você aconselharia outros colegas a fazerem este Curso?"* e *"O curso favorece melhorias para o seu desempenho profissional?"*. Além disso, as avaliações institucionais registraram para os docentes uma média de satisfação de 75% dentro dos conceitos Muito Bom ou Excelente. Alguns depoimentos ressaltam a relevância do curso para a vida profissional dos estudantes, já na 1ª. etapa das disciplinas:

"O MPES é um mestrado realmente profissional. Não é um Mestrado Acadêmico disfarçado. É um legítimo Mestrado Profissional, com seu corpo docente formado por profissionais, experientes e especialistas em suas áreas. O foco do mestrado é solucionar problemas reais do mercado". Gustavo Alexandre, aluno da 1ª turma do MPES do C.E.S.A.R.

"A grande competência dos professores, que são profissionais com larga experiência no mercado, e o ambiente de experimentação real, que os alunos vivem dentro do Mestrado com as fábricas de software, são grandes diferenciais do MPES do C.E.S.A.R.edu. O próprio nome C.E.S.A.R é respeitado no mercado de software e com um Mestrado reconhecido pela CAPES que estimula os alunos a viverem situações reais com os seus clientes, ótima infra-estrutura e corpo docente experiente na área, isso torna este curso de Mestrado muito empolgante e interessante". Rodolfo Wagner, mestrando da turma MPES 2008.1.

"...2. Mestrado Profissional: Por ser mais focado no mercado, considero mais bem aproveitado o aprendizado do mestrado nas minhas atividades profissionais. 3. Confiança na empresa (cesar.edu): ...Dentro da empresa, eu era visto como um excelente profissional, mas existiam críticas em relação a minha capacidade de gestão e o Mestrado vem me ajudando a mudar essa visão justamente por deixar de ser ad-hoc." Jean Carlos Medeiros da Silva, Líder de equipe e mestrando da turma MPES 2008.1

Tendo como principal motivação a avaliação do Mestrado Profissional sob os critérios de transferência de conhecimento para a sociedade, capacitação de profissionais qualificados para o exercício de uma prática profissional transformadora e melhoria da eficácia e eficiência das empresas, uma Pesquisa foi realizada com os alunos das turmas 2007-2, 2008-1, 2008-2 e 2009-1 com o objetivo de verificar a evolução profissional relacionada ao MPES, como demonstram os relatos a seguir:

Pesquisa realizada: abordagem em três questões

1. Qual a sua função/cargo durante o período de entrada no mestrado?
2. Qual a sua função/cargo atual?
3. Você obteve alguma evolução em rendimentos, cargos, empresa ou tipo de contratação devido ao mestrado? Se houve evolução em rendimentos, qual foi o percentual de aumento?

Resultados da pesquisa: resultados positivos

"Clóvis Holanda – conclusão prevista para 2º. Semestre de 2010 1. Analista de Sistemas Senior. 2. Analista de Sistemas Senior. 3. Obtive 10% de aumento no salário, e a empresa está pagando o valor integral do mestrado."	"Fernando Scalia – conclusão prevista para 2º. Semestre de 2010 1. Analista de Sistemas. 2. Arquiteto de Software. 3. Houve uma evolução em cargo e rendimento, de aproximadamente 30%."
"Wesley Silva – conclusão prevista para 2º. Semestre de 2010 1. Desenvolver Júnior. 2. Atualmente ainda desenvolvedor júnior, mas em abril serei analista de sistemas. 3. Houve um acréscimo de 9% no momento da contratação no emprego atual, porém no cargo que irei assumir assim que finalizar o mestrado obterei o acréscimo de mais 10%."	"Nelson Carnaíba – conclusão prevista para 2º. Semestre de 2009 1. Engenheiro de Sistemas junior no C.E.S.A.R 2. Analista de Sistemas e de Processos de Software na Politec 3. Enquanto estava no C.E.S.A.R, recebi uma promoção, passando a ser engenheiro de sistemas pleno, havendo um aumento em torno de 35% de meu salário. Porém, recebi uma proposta de trabalhar como analista de sistemas e processos na Secretaria da Fazenda de Alagoas pela Politec. O aumento houve em termos de responsabilidades profissionais e no prestígio da qualidade de meu trabalho. Estar no mestrado foi um fator determinante para assumir a responsabilidade que hoje tenho. (...) "
"Jean Medeiros – conclusão prevista para 2º. Semestre de 2009 1. Analista de Sistema e Especialista de Negócio 2. Coordenador de Equipe 3. Não houve mudança salarial"	"Ademar Izu Gomes – conclusão prevista para 1º. Semestre de 2009 1. Eu era desenvolvedor pleno no período que ingressei no mestrado. 2. Sou desenvolvedor sênior. 3. O fato de estar cursando um mestrado foi um fator determinante para a minha alocação em um cliente em Londres. Tive um percentual de aumento de 30%."
"Eric Cavalcanti – conclusão prevista para 1º. Semestre de 2009 1. Engenheiro de Sistemas Pleno 2. Engenheiro de Sistemas Sênior 3. Não. Considerando que eu não concluí o mestrado. Espero que ao concluir o item 3 possa ser sim."	"Rodrigo Frutuoso Lopes – mestrado concluído 1. Engenheiro de Teste - Junior 2. Engenheiro de Teste – Pleno 3. Fui contratado como professor pela Faculdade de Filosofia, Ciências e Letras de Caruaru (FAFICA), como também fui re-enquadrado dentro da empresa C.E.S.A.R, o qual deixei de ser Eng de Teste Junior e agora sou Eng de Teste Pleno. No total o aumento foi na ordem de 35%."
"Gustavo Alexandre – mestrado concluído 1. Engenheiro de Teste / Engenheiro de Junior 2. Engenheiro de Teste / Engenheiro de Pleno 3. Por enquanto, não."	"Vanessa Freitas Cândido – conclusão prevista para 1º. Semestre de 2009 1. Engenheira de Testes 2. Líder de Equipe 3. Não."

4.2. Produtos de software gerados

O MPES possibilitou a criação de 14 fábricas de software, gerando quatro produtos de mercado. Um deles open source, o Firescrum (www.firescrum.net), disponível para download em <http://sourceforge.net/projects/firescrum/>. O FireScrum obteve o 2o. lugar no Prêmio Dorgival Brandão Júnior da Qualidade e Produtividade em Software do Ciclo 2009 do Programa Brasileiro da Qualidade e Produtividade em Software - PBQP Software do MCT. A relação de fábricas de software e produtos pode ser encontrada em <http://www.cesar.edu.br/pesquisa/produto.php>.

4.3. Processos

- ? Metodologia PBL para o ensino de Engenharia de Software descrita na Seção 3 deste documento;
- ? Processo de Coleta de Métricas para Predição de Defeitos em Projetos de Desenvolvimento de Software, trabalho de pesquisa da aluna Melissa Barbosa Pontes;
- ? Processo de Triagem de Defeitos para Equipes de Testes de Software terceirizadas, trabalho de pesquisa do aluno Thiago Farias Costa.

4.4. Dissertações

O MPES atualmente contabiliza a defesa de 32 dissertações em diversas áreas de Engenharia de Software, as quais podem ser acessadas em <http://www.cesar.edu.br/pesquisa/dissertacoes.php>. Há uma previsão de mais 18 trabalhos concluídos até o final de 2011, totalizando 50 dissertações.

4.5. Artigos publicados

A produção científica do corpo docente e discente do MPES contabiliza no período de 2007 a 2010, 8 artigos publicados em periódicos, 124 artigos completos em anais de conferências e 40 produções técnicas. Uma relação desta produção pode ser encontrada em: <http://www.cesar.org.br/pesquisa/publicacoes/>.

4.6. Parcerias

Empresas de mercado:

- ? MV Sistemas - MV Sistemas é hoje a 6ª maior empresa brasileira de software e líder absoluta do mercado de Sistemas de Gestão de Saúde. Projeto executado pela fábrica de software H3S.
- ? BIOPAR Participações Ltda, empresa holding do Grupo OT: projeto submetido ao pool de projetos.
- ? ODIN.MED – Engenharia Clínica On line, empresa Engebio Serviços Técnicos de Engenharia LTDA. Projeto submetido ao pool de projetos.
- ? Valença & Associados – Aprendizagem Organizacional (V&A). Projeto executado pela fábrica de software F.A.R.M.
- ? Datagold Assessoria e Informática Ltda. Projeto submetido ao pool de projetos.
- ? MeanTime. Projetos executados pela fábricas de software Monjave e GameBrick.
- ? Criativa, atuando há mais de 6 anos na região norte do Brasil, a Criativa realiza projetos de desenvolvimento conectados ao futuro, com qualidade e agilidade. Dissertação.

Instituições de Ensino:

- ? UFRPE - Universidade Federal Rural de Pernambuco: colaboração docente; orientação de dissertações; projeto Xiscanoé (<http://www.xiscano.org>)
- ? UFPE - Centro de Informática: colaboração docente; orientação de dissertações; Projeto LabPS no INES (www.ines.org.br).
- ? FBV - Faculdade Boa Viagem: Projeto Smart Education.

5. Aplicabilidade dos Resultados

Quanto à relevância, o MPES tem como essência as áreas de qualidade e produtividade de software, e contribuiu com o aprimoramento e ampliação de mecanismos de multiplicação de conceitos, metodologias e técnicas de qualidade e produtividade de software, tendo em vista a Metodologia de Ensino baseada em PBL desenvolvida no âmbito do curso, publicada e apresentada em importantes eventos tais como [3], [4] e [5].

Quanto ao impacto da proposta do curso, destacam-se: a formação de mestres em Engenharia de software no Brasil, preparados para atividades profissionais na área de conhecimento em questão; promoção e aperfeiçoamento dos mecanismos de integração instituição de ensino-empresa por meio de execução de projetos reais como instrumentos de ensino, com vistas à melhoria da qualidade e produtividade de software.

Quanto à abrangência, destacam-se o desenvolvimento e implementação de experiências em pós-graduação em Engenharia de Software, que podem ser replicadas nacional e internacionalmente; desenvolvimento de modelos de ensino compatíveis com a proposta de mestrado profissional, cujo objetivo é centrado na aptidão do profissional para o mercado de trabalho; e recrutamento e desenvolvimento de projetos reais, gerando oportunidade de desenvolvimento de software a empresas sem recursos.

6. Características Inovadoras

- ? Metodologia de ensino e aprendizagem baseada em PBL, suportada por processos definidos com referência no modelo qualidade de software CMMI: “princípios de qualidade para ensinar desenvolvimento de software com qualidade”;
- ? Relevância do curso para a vida profissional dos estudantes: alto nível de satisfação do corpo discente, medida por meio de avaliações em cada disciplina e de avaliações institucionais;
- ? Articulação com empresas da indústria de software, ao trazer seus projetos para as fábricas de software criadas no MPES;
- ? Atração de estudantes de diferentes estados do Brasil, permitindo a transferência de conhecimento e tecnologia entre diferentes ecossistemas de software;
- ? “Aprenda com quem Faz”: experiência profissional prática do corpo docente, todos com o título mínimo de mestre e exercendo cargos de destaque em sua área de atuação.

7. Conclusões e Perspectivas Futuras

As demandas de mercado por profissionais com qualificação voltadas para a prática profissional e desenvolvimento de competências multidisciplinares têm estimulado o uso de metodologias de ensino diferenciadas. Embora, o método PBL tenha construído sua reputação na área médica, seus princípios mostram-se completamente alinhados às necessidades do ensino de Engenharia de Software, em particular, quando trata-se do desenvolvimento de software, uma vez que esta é uma atividade que exige trabalho em equipe, alta interatividade entre todos envolvidos e vivência prática de projetos em condições reais de mercado. A metodologia é inovadora quando define a estratégia de criação e acompanhamento destas fábricas, estabelecendo processos, ferramentas e papéis específicos, permitindo que a prática aconteça de forma sistematizada e estruturada. Os resultados positivos provenientes da execução do curso em 5 turmas comprovam sua aplicabilidade, impacto e abrangência.

Finalmente, a melhoria contínua de sua metodologia de ensino é uma premissa para o MPES. Em cada edição de uma nova turma, mudanças são implementadas, visando correção de deficiências e melhorias de resultados.

Referências

- [1] Savery, JR and Duffy, TM. Problem based learning: An instructional model and its constructivist framework. *Educ Technology*,;35(5):31-7, 1995.
- [2] Moraes, A. K.O.; and Meira, S. R. L. and Albuquerque, J. O. Open Source Software Factory – Step by Step: A Case Report. In: First International Conference on Open Source Collaborative Development Platforms (Héphaístos), 2006.
- [3] SANTOS, S. C.; CAVALCANTI, A. P. C. ; BATISTA M. C. M. ; ALBUQUERQUE, J. ; MEIRA, S. R. L. . A metodologia de ensino segundo a abordagem PBL (Problem-Based Learning): uma aplicação na área de computação. In: CQ - XI Curso de Qualidade, 2009, Bento Gonçalves. XXIX Congresso da Sociedade Brasileira de Computação, 2009.
- [4] SANTOS, S. C.; Maria da Conceição Moraes Batista ; CAVALCANTI, A. P. C. ; ALBUQUERQUE, J. ; MEIRA, S. R. L. . Appling PBL in Software Engineering Education. In: IEEE 22nd Conference on Software Engineering Education and Training, 2008, Hyderabad, Índia. v. 1. p. 182-189.
- [5] SANTOS, S. C.; BATISTA, M. C. M. ; CAVALCANTI, A. P. C. ; ALBUQUERQUE, J. ; MEIRA, S. R. L. . Usando PBL na Qualificação de Profissionais em Engenharia de Software. In: SBES, 2008, Campinas. Fórum de Educação do XXII Simpósio Brasileiro de Engenharia de Software, 13-17/10/2008. Campinas, São Paulo, 2008.

Desenvolvimento de um Método e uma Ferramenta para a Reutilização de Processos de Desenvolvimento de Software

André Luiz Peron Martins Lanna^{1;2}, Carlos Alberto Marques Pietrobon^{1;3}

¹Pontifícia Universidade Católica de Minas Gerais - Belo Horizonte - MG

²Universidade de Brasília - Gama - DF

³Universidade Federal de Ouro Preto - Ouro Preto - MG

andre.lanna@gmail.com, capietro@pucminas.br

Abstract: *Software reuse can be considered an impact factor on software quality and does not cover only work products, but also software process representation and its inherent knowledge. This paper presents the adoption of Component-Based Development (a reuse method typically applied on software artifacts) on software processes as a mean to improve software quality. Reusing known process that were previously executed tends to improve process quality as it become more stable.*

Resumo: O reuso de processos de software pode ser considerado como um fator de impacto para a qualidade de software por utilizar maneiras de desenvolvimento já testadas e consagradas na organização desenvolvedora. Este trabalho apresenta uma proposta de reuso de processos inspirada no Desenvolvimento Baseado em Componentes de software e que faz uso de conhecimento para sua realização. Reutilizar processos previamente executados (e portanto conhecidos) tende a aumentar a qualidade do novo processo à medida em que este se torna mais estável.

1. Introdução:

Atualmente a qualidade do produto de software tem sido um assunto de grande interesse tanto para a área acadêmica quanto para as organizações desenvolvedoras de software.

Diferentes abordagens têm sido propostas de modo a permitir que tais produtos possam atingir níveis mais elevados de qualidade. Algumas delas buscam garantir a qualidade ao focar o produto que está sendo desenvolvido, i.e., o software. O reuso de software é uma técnica na qual (geralmente) artefatos previamente desenvolvidos, testados e integrados em outros software são utilizados como base para o desenvolvimento do novo software.

Sua utilização oferece vantagens tais como diminuição dos custos de produção e a rápida entrega dos produtos [Sommerville 2007, Li et al. 2007]. Pelo reuso de software o aumento da qualidade do software ocorre por ele ser construído com base em elementos que já foram criados, avaliados e possivelmente melhorados no passado. Dentre as diversas formas de reuso de software destaca-se o Desenvolvimento Baseado em Componentes (DBC). Nesta abordagem as estruturas reutilizáveis são chamadas “componentes” que possuem seus próprios requisitos e interfaces através das quais são agrupadas (compostas) para formarem estruturas maiores e mais complexas (o software propriamente dito) [Szyperski 2002, Wang and Sun 2009].

Outras abordagens buscam aumentar a qualidade do software ao realizar melhorias em seu processo de desenvolvimento. Nestas abordagens acredita-se que processos que possuem qualidade geram produtos com qualidade, o que indica uma clara relação entre produto e processo. Entretanto definir um processo de software é uma tarefa difícil de ser realizada, uma vez que não devem ser considerados apenas os fatores técnicos de um processo mas também os organizacionais e humanos [Lanna and Pietrobon 2008].

Tais aspectos devem ser dispostos de modo harmônico de modo a atender às necessidades da organização. Por este motivo esta tarefa geralmente é realizada por pessoas mais experientes, com conhecimentos profundos dos processos organizacionais, na tentativa de definir processos mais estáveis e de qualidade [Barreto et al. 2008].

As duas abordagens para qualidade de software apresentadas (foco no produto e no processo) apresentam resultados positivos quando efetivamente implantadas em uma organização. Entretanto atualmente elas ocorrem de modo separado e seus resultados podem ser potencializados se forem consideradas em conjunto [Lucrédio et al. 2008]. Osterweil em [Osterweil 1987, Osterweil 1997] indica similaridades entre software e processos ao afirmar que é possível definir requisitos, projetar, implementar e evoluir processos de software, tal como ocorre no desenvolvimento de software. Tais semelhanças permitem que técnicas geralmente aplicadas em artefatos de software sejam adaptadas e aplicadas em processos de software, incluindo aquelas relacionadas ao reuso de software.

O projeto “Desenvolvimento de um método e uma ferramenta para a reutilização de processos de desenvolvimento de software” foi submetido ao Programa Brasileiro de Qualidade e Produtividade em Software (PBQP-SW) do Ministério de Ciência e Tecnologia (MCT), e desenvolvido no Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) da PUC-Minas como tema de uma dissertação de mestrado. Este projeto visa

aliar as duas propostas vistas anteriormente em uma abordagem para reutilização de processos de software, baseado nas similaridades apontadas em [Osterweil 1987, Osterweil 1997].

Deste modo espera-se que as vantagens de ambas abordagens continuem válidas nesta proposta e que apresentem melhorias se comparadas aos resultados de cada uma isoladamente.

Este texto está estruturado como segue. Na seção 2 os objetivos do projeto e sua justificativa. A seção 3 apresenta a metodologia adotada para a execução do projeto e como cada um de seus elementos foi definido. Na seção 4 encontram-se os resultados obtidos durante o ciclo de 2010. Por fim, na seção 5 são apresentadas conclusões e perspectivas futuras para o projeto.

2. Objetivos e justificativa

[Xu 2005] e [Borges and Falbo 2002] apresentam a importância e os benefícios em utilizar o conhecimento adquirido em processos anteriores durante a definição e adaptação de novos processos de software. De acordo com [Montoni 2003] esse conhecimento possui grande valor para a organização pois pode ajudar a solucionar dúvidas ou problemas ocorridos durante a execução do processo de software. Deste modo a qualidade do software pode ser aumentada à medida em que os membros de uma organização aumentam seu conhecimento e entendimento sobre o processo com o qual estão envolvidos.

O objetivo principal deste trabalho é reutilizar processos de software junto com seus conhecimentos adquiridos em execuções passadas como uma alternativa para a

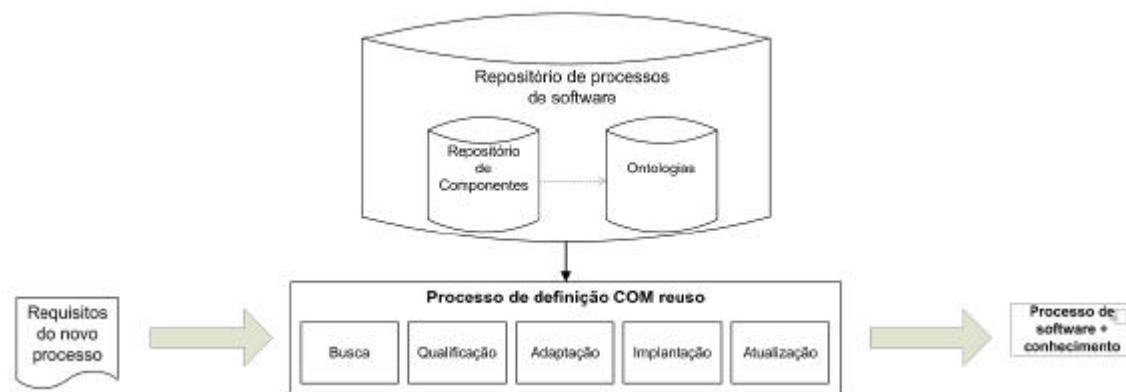


Figura 1. Abordagem proposta para a reutilização de processos de software.

definição de novos processos mais maduros e estáveis em menores tempo e custos. A abordagem foi estabelecida a partir da técnica tradicional de reuso DBC. Os seguintes objetivos específicos foram definidos: (a) a definição de uma estrutura reutilizável (componente de processo) capaz de representar processos de software em seus diferentes aspectos e armazenar o conhecimento adquirido em suas execuções passadas, (b) a descrição do processo de software e seus conhecimentos de modo a facilitar o seu entendimento e futuras reutilizações.

Baseado nas características das abordagens apresentadas anteriormente o projeto se justifica por ser mais uma proposta para a melhoria de processos. Pode-se afirmar que a melhoria ocorre como consequência da reutilização de processos previamente definidos, executados e possivelmente melhorados, e do uso do conhecimento adquirido em execuções passadas nos novos processos, o que minimiza as chances de ocorrerem falhas em suas execuções. Além isto traz as vantagens de diminuir o tempo de definição de novos processos (uma vez que não serão criados a partir do zero), permitindo que esta tarefa possa ser executada por profissionais menos experientes que utilizarão o conhecimento disponível como suporte para definição de novos processos.

3. Metodologia de execução

Os principais elementos constituintes desta proposta são o componente reutilizável de processo de software, o repositório de componentes de processo e a definição “com” reuso. O modo como eles se relacionam está apresentado na Figura 1. Em linhas gerais, inicialmente deve-se estabelecer os requisitos do novo processo. A definição do processo ocorre através de cinco etapas bem definidas (busca, qualificação, adaptação, implantação e atualização), executadas a partir de um repositório de componentes de processos. O resultado destas atividades é a definição de um novo processo com o conhecimento adquirido em execuções passadas.

O desenvolvimento desta abordagem foi realizado em quatro etapas iniciando-se pela definição de um modelo de processo de software a ser considerado no restante do trabalho. Posteriormente foi definida a estrutura do componente capaz de representar processos de software aderentes ao modelo pré-estabelecido, seguida pela descrição do componente de processo de software a fim de facilitar seu entendimento e futura reutilização.

Por fim foi estabelecido um processo a ser executado para guiar o modo como os componentes de processo são extraídos do repositório e reutilizados na definição do novo processo. Um ferramenta de apoio foi criada para facilitar a busca de componentes e definição de novos processos. Cada uma destas etapas está descrita a seguir.

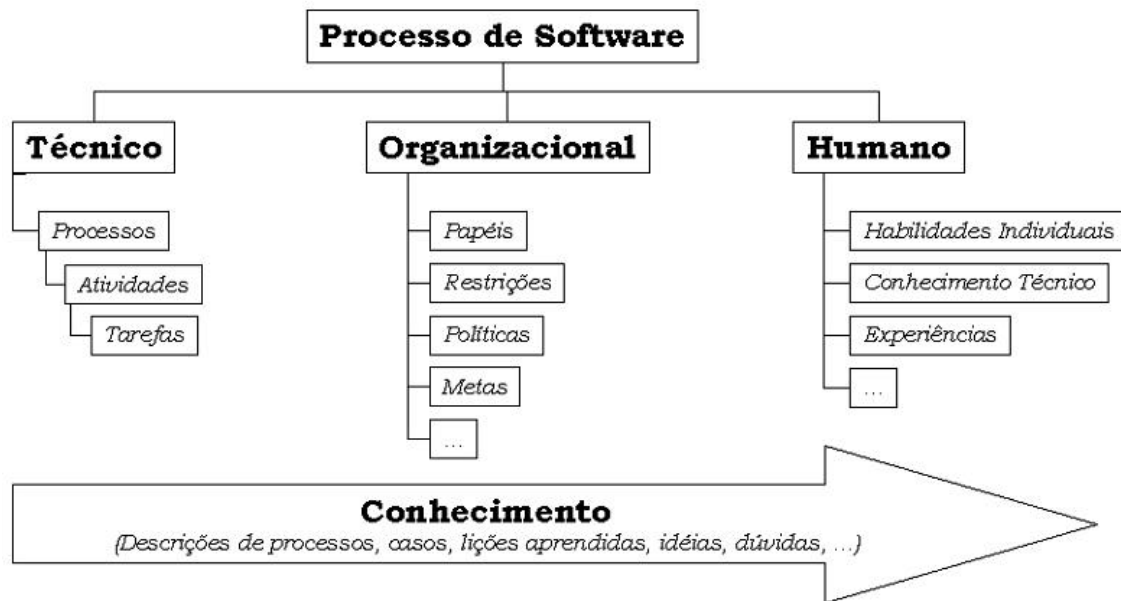


Figura 2. Elementos constituintes do modelo de processo de software proposto pelo projeto.

3.1. Modelo de processo de software

Um processo de software deve ser definido através dos aspectos técnico, organizacional e humano sendo eles dependentes entre si, pois decisões em um dado aspecto podem influenciar os outros dois. Portanto durante a definição de um processo de software eles devem ser trabalhados em conjunto de modo a harmonizá-los. Esta proposta considerou o aspecto técnico como sendo o nível de detalhamento (granularidade) para a representação de um processo, pois ele pode representar todo o processo ou partes dele (atividades e tarefas).

Os conhecimentos de um processo de software servem para descrevê-lo ou registrar situações ocorridas em alguma execução passada. Cada registro de conhecimento em um processo de software é chamado de item de conhecimento e pode ser de diversos tipos. Este projeto considerou os tipos de conhecimento definidos por [Montoni 2003], mas permite que as organizações definam seus próprios tipos de conhecimento. Apesar de fazer parte do processo de software, o conhecimento é considerado independente da representação do processo pelo fato do conhecimento explicar o porque do processo ser definido de uma determinada forma.

Deste modo, estabeleceu-se como modelo de processos de software o conjunto de aspectos técnicos, organizacionais e humanos, além do conjunto de itens de conhecimento.

A figura 2 representa o modelo de processos de software.

3.2. Componente de processo de software

Baseado nas características de componente de software tradicional foi definida uma estrutura reutilizável capaz de representar um processo de software conforme o modelo de processos estabelecido para este trabalho. Tal estrutura denominada "componente de processo de software" é definida como sendo: (a) unidades de composição com interfaces e dependências de contexto, (b) capaz de representar um processo de software em seus três aspectos seja qual for seu nível de detalhe (processo, atividade ou tarefa) e (c) capaz de armazenar os itens de conhecimento relacionados ao processo. O componente de processo de software tem tamanho variável conforme o nível de detalhe da representação do seu processo. Como o componente de processo representa o modelo estabelecido na figura 2, sua granularidade (tamanho) é estabelecida pelo aspecto técnico do processo de software.

```

<?xml version="1.0" encoding="UTF-8"?>

<ProcessComponent date="" author="" collaborator="">
  <TechnicalAspect granularity="">
    <TechnicalItem id="" name="" type="">Description of item</TechnicalItem>
    <TechnicalItem id="" name="" type="">
      <spem:Process>
        <spem:documentation>Descrição de um item no SPEM
      </spem:documentation>
      </spem:Process>
    </TechnicalItem>
  </TechnicalAspect>

  <OrganizationalAspect>
    <OrganizationalItem id="" name="" type="">Description of
item</OrganizationalItem>
  </OrganizationalAspect>

  <HumanAspect>
    <HumamItem id="" name="" type="">Description of item</HumamItem>
  </HumanAspect>

  <KnowledgeSet>
    <KnowledgeItem type="">Description of item</KnowledgeItem>
  </KnowledgeSet>

  <interface id="" name="" providedRequired="">
    <service id="" name="" numberInputArtifacts="" numberOutputArtifacts="">
      <artifact id="" name="" inputOutput="">Description of
artifact</artifact>
    </service>
  </interface>
</ProcessComponent>

```

Figura 3. Representação do componente em XML.

A realização (ou implementação) do componente de processo de software consiste em representar processos de software aderentes ao modelo da Figura 2 através de uma linguagem estruturada. Tal implementação foi realizada através da linguagem XML, para a qual foi estabelecida uma Data-type definition (DTD) aderente ao modelo de processo.

A estrutura genérica para representação de um processo em XML é apresentada pela figura 3.

Note que os quatro grandes conjuntos constituintes do modelo de processo estão representados pelas marcações *TechnicalAspect*, *OrganizationalAspect*, *HumanAspect* e *KnowledgeSet*, sendo cada um destes conjuntos formados por marcações que representam os elementos que, de fato, vão formar o processo de software. Como o componente de processo representa o modelo estabelecido na figura 2, sua granularidade (tamanho) é estabelecida pelo aspecto técnico do processo de software. A implementação do componente através do arquivo XML considera ainda a compatibilidade com o modelo SPEM, devido a sua larga utilização em ferramentas de modelagem e definição de processos. A figura 3 representa a implementação de um componente de processo e seus principais elementos.

3.3. Descrição do componente de processo

Para facilitar a definição de processos por meio de reuso, este projeto preocupou-se em descrever os componentes de processos. A descrição é realizada de duas maneiras: a descrição do componente de processo e a descrição dos elementos que constituem o processo de software.

Descrever um componente de processo de software consiste em explicar as características do arquivo XML que representa o processo de software. Foram definidos vinte e três metadados que caracterizam um componente processo de software, sendo eles agrupados em cinco categorias: identificação, utilização, maturidade, documentação e alteração. Esses metadados são representados nesta proposta através da linguagem RDF (Resource Description Framework).

Descrever o processo de software consiste em descrever seus aspectos e conhecimentos utilizando termos e conceitos definidos por alguma ontologia. A utilização da ontologia foi considerada como forma de aumentar o nível de expressão e entendimento acerca do processo. Como cada processo de software é executado em ambientes ou organizações diferentes, a descrição dos aspectos e conhecimento não deve se restringir a uma ontologia estabelecida apenas para descrever contextos de processos de software, mas sim absorver os termos definidos pela ontologia adotada pelo processo e considerá-los durante sua descrição.

3.4. Definição de processos de software com reuso de processos

O projeto propôs um método denominado “definição com reuso”, capaz de guiar a reutilização de componentes de processo de software considerando a descrição de componentes por metadados e dos aspectos e conhecimento do processo por termos de ontologia.

A definição com reuso é realizada através das cinco etapas apresentadas na Figura 2: busca, qualificação, adaptação, composição e atualização.

A busca de componentes tem como objetivo pesquisar no repositório os componentes de processo que possivelmente irão atender aos requisitos do processo em definição. Ela estabelece um nível de equivalência entre a especificação de requisitos do novo processo e as especificações dos componentes vindos do repositório, através de operações denominadas matching. O nível de equivalência destas operações é variável, contemplando a equivalência total ou parcial dos requisitos.

A etapa de qualificação de componentes tem como objetivo determinar quais componentes recuperados serão reutilizados na definição do novo processo. Essa avaliação é realizada com base nas implementações dos processos de software (através dos itens de processo que o constituem) e principalmente nos itens de conhecimento que eles possuem. Para a etapa de qualificação são estabelecidos os critérios de inclusão (determina características que obrigatoriamente o componente de processo deve apresentar para que não seja descartado) e de classificação (determinam os componentes mais aptos à reutilização). A definição destes critérios fica a cargo do responsável pelo novo processo.

A etapa de adaptação de componentes é responsável por identificar e resolver inconsistências existentes entre os componentes de processo qualificados, sendo que as inconsistências ocorrem por divergências entre as interfaces dos componentes que serão interligados.

É na etapa de implantação que os componentes de processo serão de fato reutilizados.

Esta abordagem considera que reusar componentes de processos consiste em importar para a ferramenta de execução de processos adotada pela organização todos os elementos que representam o processo de software, incluindo as ontologias utilizadas pelo novo processo, os itens constituintes dos aspectos técnico, organizacional e humano além dos itens de conhecimento.

Após todos elementos terem sido importados, o processo está definido e implantado na organização.

Por fim, a etapa de atualização de componentes é responsável pela troca de um componente de processo por um outro de qualidade mais alta e cujos serviços apresentam mais vantagens. Para realizar esta substituição é necessário identificar os requisitos do componente a ser substituído, pesquisar no repositório os componentes que os atendem, escolher dentre os candidatos aquele que melhor irá substituí-lo, realizar as adaptações necessárias para resolver possíveis incompatibilidades e finalmente realizar a implantação do componente. Como se pode notar a atualização consiste em executar novamente as quatro etapas anteriores da definição com reuso.

3.5. Apoio ferramental

Um apoio ferramental para a abordagem proposta foi desenvolvido com o objetivo de auxiliar as organizações a definirem seus processos de software através da reutilização de processos. A ferramenta é capaz de considerar termos definidos por diversas ontologias utilizadas na descrição dos diferentes aspectos e itens de conhecimento de um processo de software. Tal ferramenta consiste do repositório de componentes e ontologias de processo de software e no suporte às etapas da definição “com” reuso.

4. Resultados obtidos

Alguns resultados foram obtidos durante a execução do projeto em questão.

Durante o ciclo de 2010 foram submetidos três artigos a eventos nacionais e internacionais, tendo um deles sido premiado com o primeiro lugar. Estes resultados estão relacionados a seguir.

O primeiro artigo foi submetido ao Concurso de Teses e Dissertações em Qualidade de Software (CTDQS), na categoria trabalhos de mestrado. Este evento é integrante do Simpósio Brasileiro de Qualidade de Software (SBQS) e caracteriza-se por avaliar e premiar os trabalhos produzidos no ano anterior em Qualidade de Software.

Neste concurso a dissertação foi premiada com o primeiro lugar da categoria de trabalhos de mestrado. O evento ocorreu entre os dias 7 e 11 de junho de 2010, em Belém-PA [Lanna and Pietrobon 2010a].

O segundo artigo foi submetido e aprovado no Distributed Multimedia Systems (DMS 2010). O evento ocorreu entre os dias 14 e 16 de outubro de 2010, na cidade de Oak Brook, Illinois, USA. Neste evento foi apresentado com detalhes a estrutura do componente de processo e, principalmente, como o conhecimento foi representado, armazenado e recuperado na proposta [Lanna and Pietrobon 2010b].

Por fim a proposta como um todo foi apresentada no encontro Software Engineering Process Group Latin America (SEPLA), evento promovido pelo Software Engineering Institute (SEI). O evento ocorreu entre os dias 10 e 12 de novembro de 2010, na cidade de Medellín, Colômbia [Lanna and Pietrobon 2010c].

5. Conclusão e perspectivas futuras

Este trabalho apresentou uma proposta para reutilização de processos de software com base nos diferentes aspectos que formam um processo de software e no conhecimento adquirido em suas execuções passadas. Tais aspectos e conhecimentos de um processo são representados em componentes reutilizáveis, os quais são descritos através de metadados e por termos estabelecidos em ontologias, além de serem armazenados em repositório. Foi apresentado também a definição “com” reuso que estabelece um método para definição de processos de software, compreendendo desde a busca do componente no repositório até a implantação no ambiente de execução de processos. Por fim foi proposta uma ferramenta capaz de auxiliar as etapas da definição “com” reuso.

O intuito desta abordagem é aliar as vantagens do reuso de software tradicional, tais como economia de tempo e custos além do aumento da qualidade aos benefícios oriundos de uma boa definição de processo de software. Deste modo acredita-se que o aumento da qualidade dos novos processos será consequência da reutilização de processos já executados e portanto conhecidos pela organização.

Nesta abordagem a estrutura reutilizável mostrou-se capaz de representar os diferentes aspectos e conhecimentos que formam um processo de software, indiferente da granularidade do processo representado (processo, atividade ou tarefa). A utilização de ontologias na definição “com” reuso permitiu que processos e componentes de processo (principalmente suas interfaces) fossem descritos de um modo mais expressivo se comparados à simples utilização de palavras-chaves. Os termos definidos por uma ontologia trouxeram significados aos elementos constituintes do processo de software, de modo que a definição de novos processos agora ocorre com base na semântica dos termos que o compõem.

Isto exposto pode-se afirmar que este trabalho apresenta como contribuições a definição de um modelo para representação de processos de software com base em diferentes aspectos e conhecimento; a definição da estrutura de componente de processo reutilizável para representação dos processos; a definição de formas de descrição voltadas ao componente de processo (metadados) e ao próprio processo (através dos termos definidos pelas ontologias e, por fim, a definição de um método para reutilização de componentes de processo.

Como continuação deste trabalho estão previstos estabelecer a definição “para” reuso, capaz de criar componentes de processo de software “do zero” para popular o repositório de componentes de processo. Além disto a utilização de alguma forma de Gerência de Conhecimento aplicada ao repositório de componentes de processo será de grande valia, principalmente para a definição “com” reuso, de modo a disponibilizar somente os processos cujos conhecimentos possuem importância na definição de um novo processo.

Portanto a implantação da Gerência de Conhecimento no repositório é considerado como um trabalho futuro.

Referências:

- Barreto, A. S., Murta, L. G. P., and Rocha, A. R. (2008). Software process definition: a reuse-based approach. In XXXIV CLEI - Conferência Latinoamericana de Informática, pages 409–419.
- Borges, L. d. M. S. and Falbo, R. d. A. (2002). Managing software process knowledge. In International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications - CSITeA'02, pages 227–232.
- Lanna, A. L. P. M. and Pietrobon, C. A. M. (2008). Proposta de reutilização de processos utilizando conhecimento e visualização. In VI Workshop Brasileiro de Teses e Dissertações em Qualidade de Software, pages 1–10, Florianópolis, SC.
- Lanna, A. L. P. M. and Pietrobon, C. A. M. (2010a). Reuso de processos de software baseado na componentização de processos e conhecimento. In Concurso de Teses e Dissertação em Qualidade de Software - CTDQS 2010, pages 1–14, Belém, PA.
- Lanna, A. L. P. M. and Pietrobon, C. A. M. (2010b). Software process reuse based on knowledge and process components. In Distributed Multimedia Systems - DMS2010, pages 1–4, Oakbrook, IL, USA.
- Lanna, A. L. P. M. and Pietrobon, C. A. M. (2010c). Software process reuse based on knowledge and process components. In Software Engineering Process Group Latin America - SEPLA 2010, pages 1–2, Medellín, Colômbia.
- Li, J., Gupta, A., Arvid, J., Borretzen, B., and Conradi, R. (2007). The empirical studies on quality benefits of reusing software components. Computer Software and Applications Conference, Annual International, 2:399–402.

- Lucrédio, D., dos Santos Brito, K., Alvaro, A., Garcia, V. C., de Almeida, E. S., de Mattos Fortes, R. P., and Meira, S. L. (2008). Software reuse: The brazilian industry scenario. *Journal of Systems and Software*, pages 996–1013.
- Montoni, M. A. (2003). *Aquisição de conhecimento: uma aplicação no processo de desenvolvimento de software*. Master's thesis, Universidade Federal do Rio de Janeiro.
- Osterweil, L. (1987). Software processes are software too. In *Proceedings of the 9th international conference on Software Engineering, ICSE '87*, pages 2–13, Los Alamitos, CA, USA. IEEE Computer Society Press.
- Osterweil, L. J. (1997). Software processes are software too, revisited: an invited talk on the most influential paper of icse 9. In *Proceedings of the 19th international conference on Software engineering, ICSE '97*, pages 540–548, New York, NY, USA. ACM.
- Sommerville, I. (2007). *Engenharia de Software*. Pearson Addison-Wesley, São Paulo, SP, 8 edition.
- Szyperski, C. (2002). *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition.
- Wang, H. and Sun, J. (2009). A semantic web environment for components. *Knowl. Eng. Rev.*, 24:59–75.
- Xu, P. (2005). Knowledge support in software process tailoring. In *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 3 - Volume 03*, pages 87.3–, Washington, DC, USA. IEEE Computer Society.

Definição do Processo APD e Certificação no Nível F do MR-MPS Utilizando Práticas do Processo Unificado Associadas a Práticas SCRUM

Tatiane Silva¹

¹Athenas Engenharia de Software - Av. Rio Branco, 12, 14º andar, Centro CEP 20090-000 - Rio de Janeiro - RJ - Brasil

tatiane@athenassoftware.com.br

Abstract. *The definition of a software process that meets the technical needs and a financial software firm needs to consider not only the best market practices, but also the most appropriate to their reality. This paper presents the evolution of software development process of Athenas Software, formalized in Version 4.0 encompasses practices that based on the existence of the Unified Process, and also the agile practices SCRUM together the practices associated with the MPS.BR Reference Model Level F practices and subsequent certification.*

Resumo. *A definição de um processo de software que atenda as necessidades técnicas e financeiras de uma empresa de software precisa considerar não somente as melhores práticas do mercado, mas também as mais adequadas a sua realidade. Este artigo apresenta a evolução do processo de desenvolvimento de software da Athenas Software, sua formalização na versão 4.0 englobando práticas que basearam a existência do Processo Unificado e, também, práticas ágeis do SCRUM juntamente as práticas associadas ao Nível F do Modelo de Referência do MPS.BR e a posterior certificação.*

1. Introdução

A Athenas Engenharia de Software é uma empresa localizada no Rio de Janeiro com 14 anos de atuação na área de desenvolvimento de sistemas e soluções de negócio baseadas nos princípios e padrões da engenharia de software. Desde sua fundação a empresa utiliza processos de desenvolvimento de software para construção de seus produtos, baseados em elementos do Processo Unificado [Jacobson *et al.* 1999]. Estes elementos possuem como principal característica a documentação durante o desenvolvimento do projeto, principalmente na etapa de análise e projeto. Após diversas evoluções e a necessidade de comportar metodologias de modelagem e de componentização, o processo de desenvolvimento da Athenas, atualmente, é resultante de uma combinação entre os elementos do Processo Unificado, algumas práticas ágeis (retiradas pontualmente das metodologias Scrum [ScrumAlliance 2011] e *Extreme Programming* (XP) [Beck e Andres 2005] e as exigências do nível F do MR-MPS [Softex 2009].

Este artigo possui o objetivo de apresentar como o processo da Athenas foi formalmente definido até a sua certificação no nível F do MR-MPS. Na Seção 2, os objetivos e justificativas com relação ao projeto são apresentados. A metodologia de execução é apresentada na Seções 3, Na seção 4 são tratados e demonstrados os resultados obtidos, na seção 5 encontra-se a descrição da aplicabilidade dos resultados, na seção 6 são vistas as conclusões e perspectivas futuras. Por fim na seção 7 são abordadas as características inovadoras do projeto.

2. Objetivos e Justificativas

As soluções providas pela Athenas têm por objetivo auxiliar o desenvolvimento não apenas de sistemas, mas também de soluções personalizadas e inovadoras, que sejam confiáveis, estáveis, seguras, escaláveis, dentro dos custos e prazos estipulados. Dentro deste objetivo, a utilização de um processo de desenvolvimento sempre se fez essencial para que, definidos os fluxos do processo, fosse possível aplicá-lo repetidas vezes e, por consequência, medir e melhorar este processo.

Com o passar dos anos, o processo utilizado, que teve suas fundamentações lançadas ainda em 1996 através de trabalho final apresentado na PUC sob nome Órion, mostrou a necessidade de adequar-se de forma mais justa as necessidades das empresas clientes e do próprio mercado que começou a direcionar modelos de referência e melhores práticas tais como CMMI e MR-MPS. O processo inicial estava permeado de muita documentação oriunda de suas bases Catalysis e Objectory, as mesmas que viriam a dar origem ao RUP, e esta documentação era questionada pela equipe e pelos stakeholders que estavam envolvidos diretamente no processo. Além disso o processo utilizado não possuía um apoio ferramental adequado, o que dificultava a sua disseminação entre os membros da equipe.

Some-se a isso ainda a necessidade quase urgente de realizar uma formalização do processo, seus templates e orientações para que ele deixasse de ser um ativo na cabeça de algumas pessoas e pudesse ser melhor absorvido por todos os interessados, assim como pudesse deixar de ser um ativo de risco da empresa, devido a dependência da presença de algumas pessoas específicas para correta utilização e disseminação do processo.

Mediante esta realidade a Athenas iniciou um estudo relacionado ao MR-MPS e a metodologia Ágil que culminou no ingresso na IOGE e no início da implementação, a princípio, Nível G, junto a instituição implementadora COPPE-UFRJ.

3. Metodologia de Execução

O projeto de melhoria de processo de software realizado em 2010, que levou a Athenas ao nível F do MPS.BR, foi dividido em: (i) fase de estudos, (ii) fase de definição do processo APD (iii) fase de execução APD (iv) fase de definição do APO(Athenas Processos Organizacionais) (v) fase de execução do APO (vi) fase de correções (vii) fase de certificação.

Fase Estudos: Foram feitos levantamentos das melhores práticas que o mercado sinalizava e dos principais modelos de referência. Nesta fase foram analisadas práticas do CMMI,MR-MPS,PMBOOK,Conceitos de Metodologias ágeis, SCRUM, XP,RUP,BABOOK, Disciplinas de Engenharia de Software, entre outras leituras. Todos estes estudos já consideravam a existência de um processo de 14 anos de maturidade. Esta maturidade foi considerada para análise e conclusões dos estudos.

Desta fase o resultado foi um possível alinhamento ao MR-MPS devido a possibilidade de atingir os níveis de forma mais específica do que o CMMI, onde grandes saltos são dados entre um nível e outro. Outro alinhamento observado foi relacionado a utilização de práticas ágeis no processo, mais precisamente, práticas existentes no SCRUM, por ultimo, algumas práticas do PMBOOK se mostraram bastante próximas da realidade encontrada na empresa e em seus projetos com relação ao gerenciamento de projeto.

Fase de definição do processo APD: Dentro desta fase definimos a ferramenta onde o APD seria formalizado(no caso o EPF - Eclipse Process Framework) e após a reunião inicial com a instituição implementadora, verificamos a real possibilidade de, apesar de inscritos para Nível G, já aumentarmos o escopo relacionado a certificação para o nível F.

O processo foi rascunhado e apresentado a COPPE e de forma interativa e incremental toda a estrutura básica do processo foi definida e já formalizada no EPF, que por padrão já segue a metodologia SPEM – Software Process engenniring Metamodel. Dentro desta fase foram ainda definidos os papeis a serem utilizados no Processo de acordo com a estrutura já existente na empresa. Com 7 pessoas foi possível implementar o nível F, a saber, uma das menores estruturas onde este nível já foi implementado.

Fase de execução APD: Foi selecionado um projeto a ser iniciado no APD para que pudéssemos colher as experiências da equipe em relação ao uso do novo processo. Durante a execução, toda a equipe recebeu treinamento no processo e foi feito um acompanhamento diário paralelo a execução, onde sempre que um ajuste se mostrava necessário era levado imediatamente para a equipe de processos avaliar, e sendo deferido, executar e implantar a alteração no processo o quanto antes. Esta proximidade foi fundamental para o correto alinhamento do processo.

Fase de definição do APO(Athenas Processos Organizacionais): Após duas iterações executadas no processo dentro do projeto inicial , houve uma estabilização e o número de ajustes a serem realizados caiu consideravelmente. Iniciou-se então a definição dos processos que seriam executados a nível organizacional, conforme definido no MR-MPS para o nível F.

Fase de execução do APO: O APO foi executado pela primeira vez e os resultados apresentados a equipe e a diretoria. Ajustes foram sugeridos e devidamente planejados para que a próxima rodada do processo, que ocorre bimestralmente, já contasse com as alterações.

Fase de correções: Com a entrada de novos projetos, o APD e o APO ainda contaram com alguns ajustes para que se pudessem atender as mais variadas situações.

Fase de certificação: A avaliação inicial foi realizada contando com poucas e simples correções a serem feitas. A avaliação final ocorreu de forma satisfatória e a unidade organizacional de desenvolvimento alcançou o nível F. Na avaliação, foram ressaltados pontos fortes no atendimento a requisitos que fazem parte de níveis mais elevados dentro do MR-MPS, superando as expectativas dos avaliadores.

4. Resultados Obtidos

(i) a adequação do processo existente aos requisitos dos processos do MR-MPS para o nível F, ultrapassando as expectativas e já atendendo a requisitos de níveis mais elevados; (ii) a transcrição do APD por meio uma ferramenta adequada para representação de processos (no caso, o EPF - *Eclipse Process Framework* [EPF 2011]); (iii) a inclusão de práticas ágeis; e (iv) inclusão de boas práticas encontradas no PMBOK [PMI 2008]. (v) Certificação nível F MR-MPS.

Os resultados foram obtidos sem deixar de lado as principais fundamentações do processo até então: a utilização de componentização (CBD), a utilização de um *framework* para geração de código e para implementação de princípios e padrões da engenharia de software e a forte característica de modelagem de componentes através da MDA (*Model Drive Architecture*).

Atualmente, o APD possui como fundamentação, além do processo unificado, práticas ágeis retiradas pontualmente do Scrum e XP, dando suporte às etapas de modelagem e internalizando as necessidades específicas para utilização de um *framework* de negócio na construção de software, tais como: (i) utilização do conceito de *sprints* e liberações; (ii) o aumento da comunicação entre a equipe através da realização de reuniões diárias; (iii) prática de reuniões de planejamento e de controle de finalizações de etapas do projeto; e (iv) realização de atividades relacionadas à utilização do *framework* como parte da construção.

5. Aplicabilidade dos resultados

A certificação e a existência do processo definido foram fundamentais para a reestruturação técnica da empresa e a satisfação de nossos clientes em participar do projeto foram resultados interessantes e até inesperados.

6. Conclusão e perspectivas futuras

O APD (Athenas Processo de Desenvolvimento) é um processo de desenvolvimento fundamentado em Engenharia de Software. Seu compromisso primário e fundamental é com o ROI (Retorno de Investimento) e o uso prático em ambientes de desenvolvimento de software. Sendo assim, orienta-se em atender as necessidades da empresa e seu negócio, bem como de seus gestores e diretores, e não unicamente com viés técnico e modismos de nossa área. O APD evoluiu por meio de um processo de melhorias e encontra-se, atualmente, em sua versão 4.0. É um processo relativamente maduro, de uso prático, alinhado com métodos aprovados pela comunidade de informática, focado nos princípios, padrões e melhores práticas de Engenharia de Software, avaliado como aderente ao nível F no MR-MPS.

Como lições apreendidas pode-se listar: (i) a definição de processos precisa ser feita de forma incremental, sendo fundamental definir e executar estes processos, pois somente assim é possível avaliar o que está adequado e o que pode ser melhorado; (ii) é necessário atender as necessidades da empresa e alinhar o processo a seus objetivos; e (iii) profissionais mais antigos na empresa podem ter mais dificuldade e, assim, exigir maior atenção com relação à execução dos processos.

O grande desafio da Athenas Software sempre foi aliar a utilização de um processo de desenvolvimento (com todas as suas regras e procedimentos) à necessidade de produzir o retorno adequado, ou seja, lucro ao cliente, por meio dos produtos com a qualidade necessária a uma empresa de Engenharia de Software. Atualmente, o APD cumpre com este objetivo e está sendo continuamente melhorado de forma a ser um dos pilares para satisfazer as necessidades dos clientes e da própria organização.

O projeto que envolve o APD no momento irá caminhar no sentido da certificação nível C. este projeto irá iniciar em novembro de 2011. Até lá o processo seguirá sendo melhorado constantemente.

7. Características inovadoras

O projeto de formalização e de melhoria de processos do APD teve como resultado a definição de um processo inovador, brasileiro e que tem por objetivo conseguir atender a qualquer empresa que trabalhe com desenvolvimento de software. O processo é fundamentado na utilização de ferramentas de apoio e na customização mediante a sua execução em cada projeto.

Esta customização permite uma ampla aplicabilidade do APD no que tange aos projetos de software. O projeto apóia equipes distribuídas ou não, projetos iterativos, incrementais e apóia ainda aqueles que optarem por um modelo cascata.

A flexibilidade com a qual foi construído o APD, o transforma em um importante aliado a qualquer fábrica de software que precise de uma base forte para trilhar os caminhos dentro da área de processos.

Referências

Beck, K.; Andres, C. (2005) "Extreme Programming Explained: Embrace Change", 2 ed. Upper Saddle River: Addison-Wesley.

D'Souza, D.F.; Wills, A.C. (1999) "Objects, Components and Frameworks with UML: The Catalysis Approach", Reading MA: Addison-Wesley.

EPF (2011) "Eclipse Process Framework Project". Disponível em: <http://www.eclipse.org/epf>. Acessado em: março/2011.

Fuzion (1999) "Metodologia Vinct", CD-ROM. E-book.

Jacobson, I. (1992) "Object-Oriented Software Engineering: A Use Case Driven Approach", ACM Press, Reading, MA.

Jacobson, I.; Booch, G.; Rumbaugh, J. (1999) "Unified Software Development Process", Addison-Wesley.

PMI – Project Management Institute (2008) "A Guide to the Project Management Body of Knowledge (PMBOK Guide)", 4ª ed., Newton Square: PMI Publications.

ScrumAlliance (2011) Disponível em: <http://www.scrumalliance.org>, acessado em março/2011.

Softex (2009) "MPS.BR - Melhoria de Processo do Software Brasileiro, Guia Geral". Disponível em <http://www.softex.br/mpsbr>. Acessado em: março/2011.

Implantação do Nível F do MR-MPS Combinando Características do Processo Unificado com Práticas SCRUM (artigo submetido ao SBQS-Curitiba-2011)

Catalysis Disponível em <http://www.catalysis.org/> . Acessado em: março/2011

Apoio Ferramental de Software Livre para Implementação do Processo de Gerência de Configuração do MPS.BR

Maurício Ronny de Almeida Souza¹, Wallace Michel Pinto Lira¹, Sandro Ronaldo Bezerra Oliveira¹
¹Programa de Pós-Graduação em Ciência da Computação – Universidade Federal do Pará (UFPA) -
Rua Augusto Corrêa, 01 - Guamá - 91501-970 - Belém - PA - Brasil
mauricio.ronny@uol.com.br, wallace.lira@gmail.com, srbo@ufpa.br

Abstract. *This paper describes and discusses the results of a study on the applicability of free software tools in the implementation Software Configuration Management process (GCO) of MPS.BR. The research involved a survey of tools and a proposed usage methodology to implement the expected results of the GCO process in accordance of the best practices included in the MPS.BR Reference Model.*

Resumo. *Este artigo caracteriza e debate os resultados obtidos por um estudo feito sobre a aplicabilidade de ferramentas de software livre na implementação do processo de Gerência de Configuração de Software (GCO) do MPS.BR. A pesquisa envolveu o levantamento de ferramentas e a proposta de uma metodologia de uso para a implementação dos resultados esperados do processo de GCO conforme as boas práticas constantes no Modelo de Referência do MPS.BR.*

1. Introdução

Modelos de qualidade de *software* tendem a melhorar os processos organizacionais e mensurar a capacidade da organização em gerenciar o desenvolvimento, aquisição e manutenção dos produtos e serviços de *software* [Softex, 2009]. Neste contexto, surgiu o MPS.BR (Melhoria de Processo de Software Brasileiro), que se trata de um programa desenvolvido pela SOFTEX (Associação para Promoção da Excelência do *Software* Brasileiro) voltado para a melhoria de qualidade em *software*. Este programa divulga um modelo de referência o MR-MPS [Softex, 2009].

Dentre os processos adotados pelo MR-MPS (Modelo de Referência do MPS.BR), destaca-se o processo de Gerência de Configuração (GCO). Este processo é responsável por aplicar procedimentos técnicos e administrativos para identificar e documentar as características físicas e funcionais das partes ou componentes do *software* (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.

Este trabalho objetiva analisar os resultados obtidos com o estudo desenvolvido no contexto do Projeto SPIDER (www.spider.ufpa.br), propondo a sistematização do processo de Gerência de Configuração do MR-MPS, utilizando um conjunto de ferramentas de *software* livres. Além desta seção introdutória, este artigo descreve na Seção 2 os objetivos e a justificativa do estudo realizado; a seção 3 aborda a metodologia de desenvolvimento do estudo; em seguida, a Seção 4 apresenta os resultados obtidos pelo projeto; a Seção 5 discute a aplicabilidade dos resultados, isto é, a relevância, impacto e abrangência dos mesmos; a seção 5 aponta as características inovadoras do projeto; e finalmente, a seção 6 expõe as conclusões e propostas de trabalhos futuros.

2. Objetivos e Justificativa

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 descritos nos objetivos dos processos inicialmente, dos níveis de maturidade G (parcialmente gerenciado) e F (gerenciado) do modelo MPS.BR, já que estes níveis são de maior complexidade para a implementação da maturidade organizacional, pois correspondem aos níveis iniciais dentro do modelo em questão [Oliveira, 2010].

A opção por *software* livre (ferramentas *free* e *open source*) decorre da possibilidade de redução de tempo e custos de uma implementação do MPS.BR [Oliveira, 2010], além de permitir a customização das ferramentas para melhor adequação ao modelo ou às necessidades da organização.

Neste contexto, este trabalho objetivou: (1) analisar as ferramentas de *software* livres disponíveis para apoiar a implementação do processo de Gerência de Configuração, em conformidade com os resultados esperados do MR-MPS; (2) identificar até que ponto (e quais) as ferramentas disponíveis oferecem o suporte necessário ao

processo de GCO; e (3) definir um manual para a implementação do processo, em conformidade com o modelo, a partir do uso das ferramentas analisadas.

Este trabalho também objetiva direcionar a escolha de ferramentas de software livre que integram o *suite* de ferramentas a ser definido no projeto SPIDER.

Adicionalmente, os objetivos deste trabalho convergem com o objetivo do projeto SPIDER de colaborar com a melhoria do processo de software das empresas nacionais [Oliveira, 2010], sobretudo as da região Norte, região caracterizada pela baixa popularidade de certificações oficiais de modelos de qualidade [Softex, 2011]. Para tal, a análise ferramental e o manual de implementação do processo de GCO através das ferramentas oferecem opções às organizações em busca da melhoria do processo de software.

3. Metodologia de Execução

A pesquisa foi conduzida, sob orientação do coordenador do Projeto SPIDER, por alunos de graduação e mestrado da UFPA participantes do projeto, com conhecimentos e experiências gerais e específicas (inclusive, indivíduos certificados nos cursos oficiais de introdução e implementação do MPS.BR). O projeto foi iniciado com um planejamento realizado pelo coordenador do projeto em concordância com os pesquisadores participantes.

A primeira etapa das atividades consistiu em um estudo do processo de Gerência de Configuração do MR-MPS, além de uma revisão da literatura sobre as técnicas atuais utilizadas na implementação do processo e consulta à empresa parceira do Projeto SPIDER, a Cobra Tecnologia, sobre práticas de gestão de configuração em um ambiente real de desenvolvimento. A partir deste estudo foram levantadas as funcionalidades (ou requisitos) necessárias em um conjunto de ferramentas para a devida implementação do processo de GCO. Com base nestes requisitos, foi levantado um conjunto de ferramentas para a análise da viabilidade de uso das ferramentas de software livre, atualmente disponíveis, para o objetivo do projeto.

Desta forma, a segunda etapa do projeto consistiu na elaboração de um manual para implementação do processo de GCO através do uso de ferramentas, definindo funcionalidades necessárias e como empregá-las. O documento foi elaborado descrevendo os resultados esperados do MR-MPS, para GCO, em termos de funcionalidades existentes na grande maioria das ferramentas de gerência de configuração encontradas durante o estudo, normalmente categorizadas entre ferramentas de controle de versão e controle de modificação.

Por muitas ferramentas de software livre disponíveis serem consideradas aptas à implementação do processo, optou-se por não fazer um guia específico para cada conjunto de ferramentas. Foi elaborada uma metodologia geral de uso de ferramentas, descrevendo que tipo de funcionalidades seriam necessárias nas ferramentas para implementação de cada resultado esperado, e como deveria ser procedida a sistematização, exemplificando seu uso com dois conjuntos de ferramentas distintos, cada um composto por uma ferramenta de controle de versão e uma ferramenta de controle de modificações. Os conjuntos para exemplo foram: (1) Mantis e CVS; e (2) Trac e Subversion.

A escolha dos conjuntos de ferramentas que seriam utilizados para demonstração da metodologia proposta baseou-se nos critérios: (1) popularidade (observada em estudo anterior sobre as práticas recorrentes em empresas); (2) disponibilidade de documentação; (3) o atendimento aos resultados esperados pelas funcionalidades da ferramenta; e (4) a integração entre as ferramentas (controle de mudança e controle de versão).

O manual encontra-se publicado no site do Projeto SPIDER^{1,3} e para validar a proposta, a metodologia foi aplicada no laboratório do Projeto SPIDER, apoiando tarefas de desenvolvimento de software de outros subprojetos paralelos, utilizando as ferramentas Subversion e Redmine. Adicionalmente um artigo por Souza *et al* (2010) foi submetido e publicado, em evento específico de Qualidade de Software e Gerenciamento de Projetos, descrevendo a metodologia utilizando o mesmo conjunto de ferramentas utilizado no laboratório.

4. Resultados Obtidos

O estudo realizado e sua aplicação, trouxe um conjunto de resultados e lições aprendidas:

- ? Levantamento de ferramentas de software livre para apoio à gestão de configuração, inclusive identificando as possíveis ferramentas candidatas à integração no *suite* de ferramentas a ser proposto no Projeto SPIDER;
- ? Capacitação e formação de recursos humanos, tendo este projeto culminado em um Trabalho de Conclusão de Curso, defendido por dois alunos do Curso de Bacharelado em Ciência da Computação da UFPA, e a capacitação dos mesmos no curso oficial de introdução ao MPS.BR;

³ 1 Manual de Implementação do Processo de Gerência de Configuração do MPS.BR disponível em <http://spider.ufpa.br/index.php?id=resultados>.

- ? Desenvolvimento de uma metodologia para apoiar a implementação do processo de Gerência de Configuração do MPS.BR através do uso de ferramentas livres, em forma de documentação disponível para acesso no site do Projeto SPIDER;
- ? Publicação de artigo no V Workshop de Gerenciamento de Projetos de Software, evento ocorrido durante o IX Simpósio Brasileiro de Qualidade de Software;
- ? Proposta de parceria entre Projeto SPIDER e o Instituto Federal do Pará (IFPA), em forma de consultoria para implantação da melhoria do processo de software na Diretoria de Tecnologia de Informação e Comunicação da instituição, com a implementação dos processos de Gerência de Projetos, Gerência de Configuração, Gerência de Requisitos e Garantia da Qualidade em conformidade com o MR-MPS.

5. Aplicabilidade dos Resultados

A formação de um grupo de pesquisa para o levantamento e propostas de utilização de ferramentas de software livre, constituiu uma oportunidade de melhor capacitar recursos humanos nesta linha de pesquisa, até então deficitária na região Norte. Como consequência, houve aumento da difusão da importância e viabilidade de implantação da Qualidade de Software na região, tanto no meio acadêmico quanto na indústria de software local, inclusive com o início de uma proposta de aplicação dos resultados obtidos desta pesquisa no IFPA, e a colaboração com a melhoria das práticas adotadas na empresa parceira Cobra Tecnologia.

Adicionalmente, a aplicação da metodologia proposta para implementação do processo de GCO no laboratório do Projeto SPIDER, solidificou os conhecimentos dos alunos, em Gerência de Configuração de Software, envolvidos não apenas nesta pesquisa, mas em todos os demais subprojetos do SPIDER, colocando-os em contato com um ambiente de gestão de configuração estruturado e devidamente gerenciado em conformidade com as recomendações do MPS.BR.

6. Características Inovadoras

Enquanto subprojeto do SPIDER, este projeto colabora com a disseminação e incentivo para a implementação da melhoria do processo de software, em uma região (Norte) caracterizada pela baixa popularidade de avaliações oficiais de modelos de qualidade [Softex, 2011]. Soma-se a isso, a motivação da possibilidade de flexibilização de custos para organizações, dispensando a aquisição de licenças de softwares proprietários, e de diminuição do tempo de implementação para empresas com foco na prestação de serviço de consultoria neste contexto.

Ao definir um manual que guia, passo a passo, a implementação do processo de gerência de configuração de software, o projeto possibilita também a aproximação de organizações que não necessariamente buscam uma avaliação MPS.BR, mas que desejam aprimorar seus processos adotando melhores práticas sugeridas pelo programa.

Ao optar pelo estudo de soluções já existentes, esta pesquisa também contribui para organizações que já possuem um conjunto de ferramentas de apoio definido e institucionalizado, que não precisariam adotar uma nova solução de software no lugar das que já estão consolidadas.

7. Conclusões e Perspectivas Futuras

Este artigo apresentou características, justificativa, objetivos e resultados de uma pesquisa sobre a aplicabilidade de ferramentas de software livre na implementação do processo de Gerência de Configuração do MPS.BR, desenvolvida como subprojeto do Projeto SPIDER, na Universidade Federal do Pará. As práticas propostas pelo projeto continuam em validação, ao longo do seu desenvolvimento em ambientes reais de desenvolvimento de software, em uma fábrica de software e em um departamento de apoio à tecnologia de informação. Em meio acadêmico, validações foram realizadas através da defesa de um Trabalho de Conclusão de Curso [Yoshidome e Souza, 2009] cuja banca avaliadora incluía dois avaliadores do MPS.BR, e a publicação de artigo [Souza *et al*, 2010] em evento específico à Qualidade de Software e Gerenciamento de Projetos de Software.

As práticas propostas no contexto desta pesquisa serviram de base para demais subprojetos do SPIDER, nos quais estudou-se a implementação de outros processos do MPS.BR que, direta ou indiretamente, utilizavam-se de práticas do processo de GCO em seu escopo, como o caso de Gerência de Requisitos [Cardias *et al*, 2010] e Gerência de Projetos [Yoshidome *et al*, 2010].

Enquanto trabalhos futuros, inclui-se a condução e posterior relato da experiência da condução de projetos conforme a metodologia de uso das ferramentas, proposta por esta pesquisa, no ambiente das organizações parceiras, a COBRA Tecnologia e o DTIC do IFPA. Durante a execução destas atividades, pretende-se encontrar pontos fortes, fracos e oportunidades de melhoria na metodologia proposta e aprimorá-la de acordo com o que foi observado em ambiente real de desenvolvimento.

A constante revisão de ferramentas e a seleção e adição de um conjunto específico de ferramentas no suíte a proposto pelo Projeto SPIDER também se enquadram entre projetos futuros.

8. Agradecimentos

O desenvolvimento deste trabalho está sendo apoiado financeiramente pela CAPES e CNPq, a partir da concessão de bolsas de mestrado.

Referências Bibliográficas

Cardias Junior, A. B., Bentes, L.N., Yoshidome, E. Y. C., Souza, M. R. A., 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, Equador, Abril, 2010.

Institute of Electrical and Electronics Engineers (1990) "Std 610.12 - IEEE Standard Glossary of Software Engineering Terminology".

Oliveira, S. R. B. *et al* (2010) "SPIDER - Um Suite de Ferramentas de Software Livre de Apoio à Implementação do modelo MPS.BR". Anais do VIII Encontro Anual de Computação – ENACOMP 2010, Catalão-GO.

Pressman, R. S. Engenharia de software. Tradução José Carlos Barbosa dos Santos. Rio de Janeiro: McGraw-Hill, 2006. 1056 p.

SOFTEX - Associação Para Promoção Da Excelência Do Software Brasileiro (2009) "Melhoria do Processo de Software Brasileiro (MPS.BR) – Guia Geral: 2009". Disponível em: www.softex.br.

SOFTEX - Associação Para Promoção Da Excelência Do Software Brasileiro (2011) "Total de organizações com Avaliação MPS (vigentes ou não): quadro-resumo por ano, níveis do MR-MPS e regiões geográficas". Disponível em: http://www.softex.br/mpsbr/_avaliacoes/default.asp. Última atualização em 18 de Janeiro de 2011.

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 V Workshop de Gerência de Projetos de Software - WGPS, Belém-PA.

Yoshidome, E. Y. C., Souza, M. R. de A. (2009) "Uma Proposta de Apoio Sistematizado à Gerência de Configuração do MPS.BR utilizando Ferramentas de Software Livre", Trabalho de Conclusão de Curso - Faculdade de Computação - ICEN/UFPA, Orientador Prof. Sandro Bezerra, Belém-PA.

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 VI Workshop Anual do MPS.BR - WAMPS, Campinas- SP.

Projeto CMMI-DTP: Institucionalização do Modelo CMMI na Dataprev

Guilherme Tavares Motta¹, Rosana Fernandes Osório²

¹Coordenação Geral de Qualidade de Software - DATAPREV - RJ - Brasil
guilherme.motta@previdencia.gov.br

²Coordenação Geral de Qualidade de Software - DATAPREV - RJ - Brasil
rosana.osorio@previdencia.gov.br

Abstract. This paper describes an overview of the development of “Projeto - CMMI-DTP: Institucionalização do Modelo CMMI na DATAPREV”, which is proposed to add value to the company's core business, with the implementation of reference models focused on quality, productivity and maturity in software development.

Resumo. O presente artigo descreve uma síntese do desenvolvimento do “Projeto CMMI-DTP - Institucionalização do Modelo CMMI na DATAPREV”, que tem como proposta agregar valor à atividade fim da empresa, com a implementação de modelos de referência com foco em qualidade, produtividade e maturidade no desenvolvimento de software.

1. Introdução

Na busca pela inovação e boas práticas da engenharia de software, deu-se início ao Projeto CMMI-DTP: Institucionalização do Modelo CMMI na DATAPREV. Este projeto teve por foco investir na análise e avaliação do processo de software em uso na empresa, como forma de identificar os pontos fortes, fracos e as melhorias requeridas, para tornar o processo de software da DATAPREV aderente aos objetivos e práticas estabelecidas pelo nível 3 do modelo CMMI Dev 1.2.

Como a obtenção da aderência exigida para o nível 3 do modelo CMMI Dev 1.2 requer que o processo de software atenda as 18 (dezoito) áreas de processo, nisso envolvendo mais de 300 (trezentas) práticas entre genéricas e específicas, foi requerido da equipe do Projeto CMMI-DTP alto grau de conhecimento e comprometimento. Cabe destacar que o cenário encontrado pelo projeto se caracterizava pelo baixo nível de utilização da metodologia de desenvolvimento de sistemas disponível, na época, na DATAPREV.

O atendimento à necessidade de definição do processo de desenvolvimento e manutenção de software da DATAPREV (PD-DATAPREV) foi resolvido a partir da integração das disciplinas preconizadas pela Engenharia de Software e do desenvolvimento do portal PD-DATAPREV. Esse portal dispõe de interface amigável de modo a facilitar o acesso às informações do processo (ativos de processo). Dessa forma, o desafio colocado foi plenamente atendido pelo Projeto CMMI-DTP.

2. Objetivos e Justificativas

O Projeto CMMI-DTP teve como objetivos principais: “Implementar as práticas do modelo CMMI - Dev 1.2, níveis 2 e 3, na DATAPREV; “Institucionalizar o uso dos processos definidos no PD-DATAPREV; e Internalizar o conhecimento de implementação do modelo CMMI Dev 1.2 na DATAPREV”.

3. Metodologia de Execução

O Projeto CMMI-DTP foi planejado e executado considerando as fases de Planejamento, Contratação de Consultoria, Institucionalização, Avaliações Não-Oficiais e Encerramento, conforme a WBS (Work Breakdown Structure) do projeto apresentado na Figura 1.

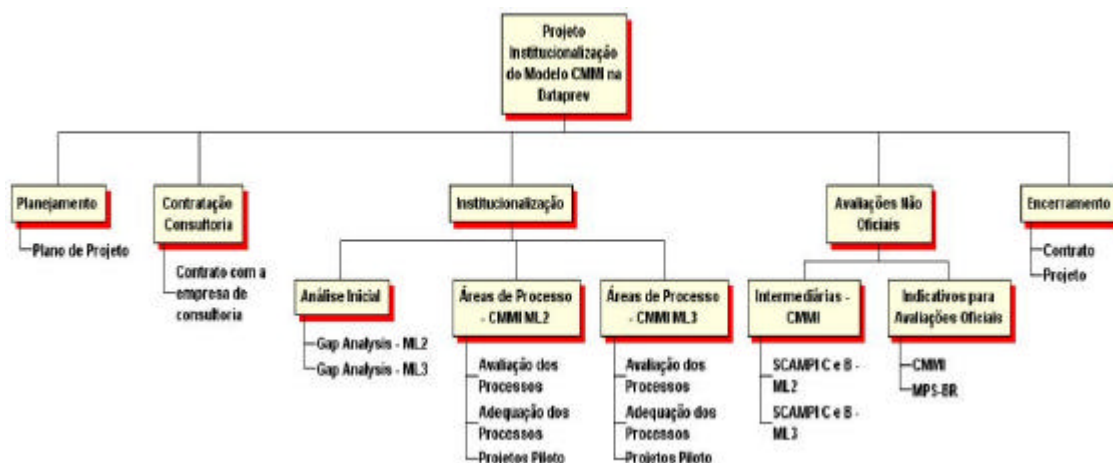


Figura 1. WBS – Projeto CMMI-DTP

Para a fase de “Contratação de Consultoria”, mais especificamente para a execução de atividades de contratação de uma empresa de consultoria especializada na implementação das práticas de nível 3 do modelo CMMI Dev 1.2, foi previsto e realizado o prazo de 06 (seis) meses. Esse prazo é considerado como indispensável ao cumprimento dos trâmites legais de aquisição de serviços.

Já para as fases de “Institucionalização”, “Avaliação Não Oficial” e “Encerramento”, o prazo foi de 24 (vinte e quatro) meses dos quais 90% dedicados às atividades de institucionalização.

Diante do porte e da importância da fase de “Institucionalização” a execução das atividades dessa fase foram tratadas segundo o fluxo metodológico ilustrado na Figura 2. O referido fluxo foi estruturado com o propósito de dar maior consistência à execução e ao monitoramento do Projeto CMMI-DTP, tendo como principal produto os Ativos de Processo.

Merece destaque o fato da definição de “Ativo de Processo” ser bastante ampla, o que permite que sejam assumidos formatos diferentes, mantendo-se, no entanto, o foco em algo que a organização considere útil para o atendimento aos objetivos do processo. Como exemplos de ativos de processo podem ser citados: políticas, processos definidos, lições aprendidas, roteiros (templates) de documentos, padrões e material de treinamento (CHRISSIS et al., 2006).

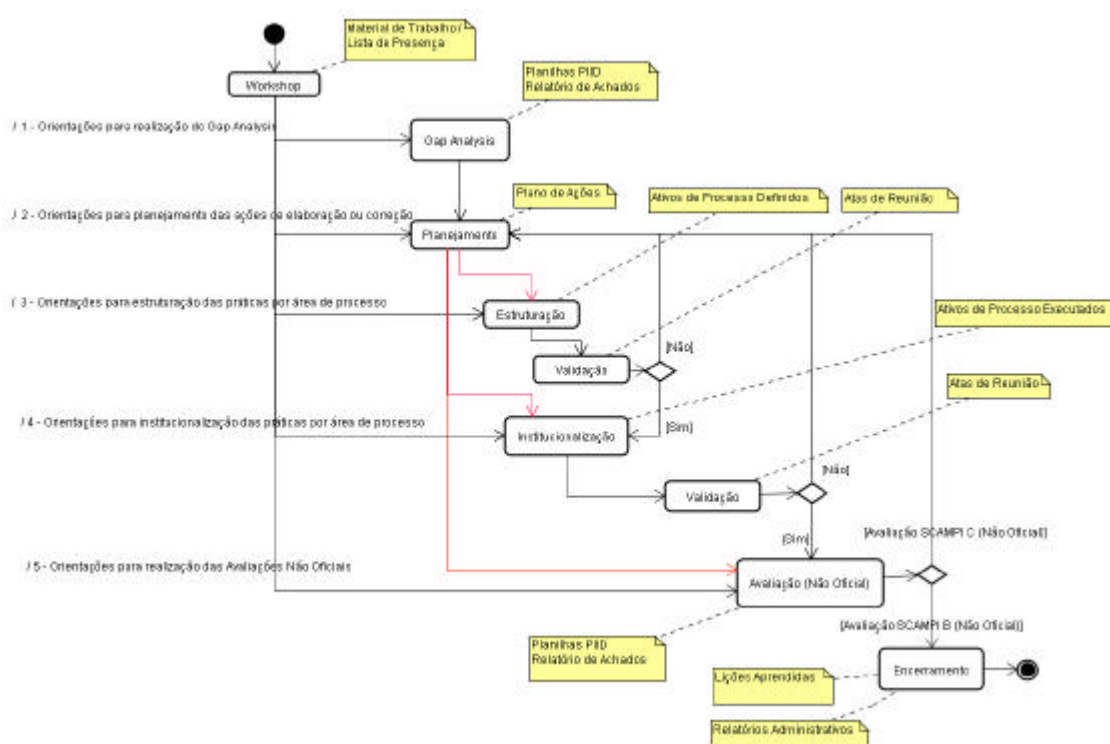


Figura 2. Fluxo metodológico aplicado ao Projeto CMMI-DTP

Cabe destacar que a consistência na condução do projeto CMMI-DTP, obtida com a aplicação do fluxo metodológico, tornou viável o atendimento a todos os objetivos definidos para o projeto e constantes no item 2 - Objetivos e Justificativas. No resumo descritivo, apresentado a seguir, podem ser observadas as atividades atribuídas a cada etapa e consideradas determinantes para o sucesso alcançado.

3.1. Workshop

Etapa em que se identificou a prática de reuniões como o principal instrumento de capacitação e transferência de conhecimento. Nessas reuniões, que sempre precederam uma nova etapa, os consultores contratados prestavam orientações de como aplicar as boas práticas da engenharia de software requeridas pelo modelo CMMI Dev 1.2. Com o conhecimento adquirido a cada workshop, os técnicos envolvidos no desenvolvimento do Projeto CMMI-DTP passavam a ter maior facilidade na execução de suas atividades. O efeito cumulativo de informações e conhecimento adquirido logo se traduziu em aumento de satisfação e produtividade da equipe, merecendo destacar as poucas ocorrências de retrabalho então registradas.

3.2. Gap Analysis

Nesta etapa foram executadas atividades de levantamento e análise dos ativos de processo em uso regular na DATAPREV. Esse trabalho possibilitou a identificação dos pontos fortes e fracos dos ativos de processo, bem como o grau de conformidade do processo existente às práticas de nível 3 do modelo CMMI Dev 1.2.

As atividades de levantamento foram realizadas pela equipe da DATAPREV, que fez uso das planilhas PIID (Practice Implementation Indicator Description) próprias ao registro das evidências. As atividades de análise foram realizadas pela equipe de consultores, tendo por resultado o relatório de achados contendo os pontos fracos identificados e respectivas recomendações de ações de ajuste e melhoria.

3.3. Planejamento

De fundamental importância ao sucesso do projeto, a etapa de planejamento tratou da elaboração dos Planos de Ação que explicitam a estratégia definida para estruturação e institucionalização dos ativos de processo, bem como da execução das avaliações não oficiais (SCAMPI B e C).

A responsabilidade pela elaboração desses planos foi compartilhada entre as equipes do Projeto CMMI-DTP e da Consultoria, de forma que as ações de implementação (ou correção), os prazos (cronogramas) de execução e os respectivos responsáveis, fossem estabelecidos em comum acordo e visando, principalmente, o comprometimento e envolvimento de todos.

3.4. Estruturação

A etapa de estruturação exigiu grande esforço por parte da equipe do Projeto CMMI-DTP, por ter reunido as atividades de definição, elaboração e correção (ajustes), do conjunto de ativos de processo necessários ao atendimento às 17 (dezessete) áreas de processo relativas ao nível 3 do Modelo CMMI. Vale destacar que a área de processo SAM (Acordo com Fornecedores), não foi contemplada no processo de institucionalização e avaliação.

Como forma tornar viável a execução das atividades dessa etapa e em conformidade ao planejamento definido, duas ações foram aplicadas. A primeira ação tratou de constituir, entre os técnicos envolvidos com o Projeto CMMI-DTP, 05 (cinco) grupos de engenharia de processo (EPG). A segunda ação cuidou de classificar as 17 (dezessete) áreas de processo, por afinidade de assunto, distribuindo-as em pacotes entre os grupos de processo conforme Tabela 1.

Tabela 1. Grupo de Engenharia de Processo / Pacote de Áreas de Processo

Grupos de Processos (EPG)	Pacotes de Áreas de Processo
1	REQM – Gerência de Requisitos; MA – Medição e Análise; CM – Gerência de Configuração.
2	PPQA – Garantia da Qualidade de Processo e de Produto; VER – Verificação; VAL – Validação.
3	RD – Desenvolvimento de Requisitos; TS – Solução Técnica; PI – Integração de Produtos; DAR – Análise de Decisão.
4	OPD – Definição do Processo Organizacional; OPF – Foco no Processo Organizacional; OT – Treinamento Organizacional.
5	PP – Planejamento de Projeto; PMC – Monitoramento e Controle de Projeto; RSKM – Gerência de Riscos; IPM – Gerência Integrada de Produtos.

3.5. Institucionalização

O termo institucionalização pode ser definido como a forma de executar negócios (processos de trabalho) adotada por uma organização, de forma rotineira, como parte de sua cultura corporativa. Isso significa dizer, que um processo institucionalizado é aquele em que todos da organização, ou apenas os diretamente envolvidos, conhecem e têm pleno domínio de “como fazer”.

O início da etapa de institucionalização se deu com o recebimento dos ativos de processo preparados na etapa de estruturação. Na sequência esses ativos foram aceitos pelos grupos de engenharia de processo (EPG) na etapa de validação (vide Figura 2).

A primeira atividade da etapa de institucionalização tratou de transmitir orientações aos seus futuros usuários (equipes de desenvolvimento / manutenção de software), por meio de treinamento ou mentoria, abrangendo a aplicação e a importância do ativo de processo no contexto do processo de desenvolvimento de software.

Concluídas as orientações o ativo de processo foi publicado e liberado para ser utilizado (executado) no fluxo normal do processo de desenvolvimento, ou ainda, dependendo das características do ativo em questão, sua execução passou a ocorrer de forma controlada em projetos piloto. Na condição de projeto piloto o ativo de processo passou a ser submetido à avaliações constantes, com o objetivo de se verificar, na prática, a efetividade de seus resultados e, só então, proceder à sua liberação no fluxo normal do processo de desenvolvimento.

Cabe mencionar que, mesmo liberado para execução no fluxo normal do processo de desenvolvimento, o ativo de processo continua sendo monitorado pelo grupo de engenharia de processo (EPG) responsável. Esse grupo, além do monitoramento com foco na melhoria contínua, acompanha as equipes de desenvolvimento e manutenção de software até que seja constatada a eficácia de sua institucionalização. Os resultados apresentados na medição de eficácia poderão gerar ações de reforço, visando ajustes nos rumos da institucionalização e garantindo vida à semente cultivada pelo Projeto CMMI-DTP.

3.6. Validação

A etapa de validação conta com a prática de reuniões de aceitação dos ativos de processo recebidos diretamente da etapa de estruturação, bem como da aceitação daqueles submetidos à etapa de institucionalização.

A dinâmica dessas reuniões baseia-se na prática do voto. Da votação participam representantes de todos os grupos de engenharia de processo (EPG), além de técnicos especialistas envolvidos na estruturação ou na institucionalização do ativo em validação.

A importância dessa etapa está no fato de que as decisões de aceite e liberação dos ativos de processo são colegiadas e envolvem o grupo de técnicos comprometidos com a qualidade do processo de desenvolvimento de software da DATAPREV.

3.7. Avaliação (Não Oficial)

Nesta etapa foram realizadas duas avaliações adaptadas do método SCAMPI (Standard CMMI Assessment Method for Process Improvement – C e B), ambas não oficiais, com o objetivo de mapear a situação atual dos processos de software da organização (ativos de processo), identificar os pontos fracos, direcionar o programa de melhoria de processos e internalizar o conhecimento do processo de avaliação de maturidade na DATAPREV.

A condução dos trabalhos de avaliação ocorreu sob a orientação da empresa de consultoria, que designou um consultor independente, ou seja, um profissional competente, porém não participante das atividades de estruturação e institucionalização desenvolvidas até então, para liderar o processo.

3.8. Encerramento

O encerramento consta das Figuras 1 e 2, como fase e etapa respectivamente. No encerramento ocorreram as seguintes atividades: obtenção das aceitações ainda pendentes; registro dos impactos da adequação de processos; documentação das lições aprendidas; arquivamento da documentação relevante; fechamento dos contratos de aquisição e formalização do encerramento do projeto.

4. Resultados Obtidos

A estratégia adotada no Projeto CMMI-DTP estabeleceu entregas parciais de serviços (produtos) associados aos resultados obtidos que, de forma incremental, incorporaram melhorias ao processo de desenvolvimento, além de internalizar o conhecimento na DATAPREV. Na relação abaixo estão registrados os serviços e respectivos resultados, representados no formato – “Serviço / Resultado”, devidamente organizados por ordem de entrega:

1. Workshop para orientar a elaboração do Gap Analysis / Aprendizado internalizado;
2. Gap Analysis - avaliação inicial / Relatório de Achados e Planos de Ação;
3. Planejamento para gestão e acompanhamento da implantação de novos processos e melhorias / Planos de ação e descrição das atividades, tarefas e papéis dos processos implantados;
4. Workshop para orientar a estruturação / institucionalização das práticas relativas às áreas de processo - Níveis 2 e 3 do modelo CMMI / Aprendizado internalizado;
5. Orientação sobre estruturação das práticas relativas às áreas de processos / Práticas estruturadas;
6. Reunião de validação da estruturação / institucionalização das práticas / Não conformidades e termos de aceite;
7. Planejamento e execução das correções das não conformidades / Plano de correção das falhas;

8. Institucionalização das práticas relativas às áreas de processo - Níveis 2 e 3 do modelo CMMI / Diagnósticos da institucionalização;
9. Workshop para orientar as Avaliações Não-Oficiais / Aprendizado internalizado;
10. Avaliações Não Oficiais (SCAMPI C e B) Níveis 2 e 3 do modelo CMMI / Relatórios de Achados e Planos de Ação.

5. Aplicabilidade dos Resultados

Ao longo do Projeto CMMI-DTP ocorreram 03 (três) avaliações. Os resultados obtidos em cada uma dessas avaliações evidenciaram a evolução das melhorias implementadas no processo de desenvolvimento de software da DATAPREV.

Além da evolução das melhorias, pode-se observar também o impacto positivo junto à equipe de técnicos envolvidos no projeto, tendo em vista o aprendizado e a percepção de melhoria declarada por todos os participantes.

De modo a explicitar a relevância dos resultados obtidos com o desenvolvimento do Projeto CMMI-DTP será apresentada a seguir uma breve descrição de cada uma das avaliações realizadas.

5.1. Primeira Avaliação - Gap Analysis

A primeira avaliação - "Gap Analysis", realizada em março de 2009, revelou que para o atendimento às 336 (trezentas e trinta e seis) práticas, entre genéricas e específicas, necessárias ao alcance do Nível 3 do Modelo CMMI – Dev 1.2, seriam necessárias 298 (duzentos e noventa e oito) ações de correção das não-conformidades identificadas.

Esse resultado demonstrou, portanto, o baixo grau de conformidade do Processo de Desenvolvimento de Software da DATAPREV, em uso regular, à época, com as práticas de Nível 3 do Modelo CMMI - Dev 1.2.

5.2. Segunda Avaliação - SCAMPI C (não oficial)

Realizada em junho de 2010, a avaliação SCAMPI C teve por objetivo verificar a evolução na implementação das melhorias e correção das não-conformidades, ou seja, o quanto a definição do processo de desenvolvimento de software da DATAPREV havia evoluído em relação ao resultado da primeira avaliação.

O resultado obtido na avaliação SCAMPI C demonstrou a evolução da melhoria do processo, tendo em vista o índice de **75%** de redução nas ações de correção das não conformidades.

Isso significa que das 298 (duzentos e noventa e oito) ações de correção, identificadas na primeira avaliação (Gap Analysis), apenas 76 (setenta e seis) ações continuavam requerendo atenção.

5.3. Terceira Avaliação - SCAMPI B (não oficial)

A terceira avaliação, compatível com o SCAMPI B teve por objetivo avaliar a evolução das melhorias na definição do processo, bem como avaliar o grau de institucionalização do processo definido.

No caso da evolução das melhorias na definição do processo, a forma de avaliação foi similar as avaliações anteriores. Já o grau de institucionalização foi verificado através do acompanhamento de 04 (quatro) projetos de softwares (Projetos Piloto), observando se os mesmos estavam utilizando os ativos de processo definidos e, ainda, se a equipe do projeto demonstrava pleno domínio (conhecimento) do processo de software (PD-DATAPREV).

Embora esta avaliação tenha sido feita compatível ao método SCAMPI B, algumas adequações se mostraram necessárias para melhor atender às expectativas das áreas funcionais da DATAPREV envolvidas com o Projeto CMMI-DTP. As adequações citadas se resumiram aos critérios de avaliação, definidos como:

- ? **Conformidade** - itens que estão em conformidade com o que preconiza o Modelo CMMI Dev 1.2 e que foram executados em todos os Projetos Piloto.
- ? **Melhoria Requerida** - itens cuja definição do processo e templates estão em conformidade com o que preconiza o modelo CMMI Dev 1.2, mas não evidenciados em, pelo menos, um dos Projetos Piloto.
- ? **Ponto Fraco** - Itens que não foram definidos ou não estão adequados ao que preconiza o Modelo CMMI Dev 1.2.

O resultado obtido na avaliação da definição do processo revelou que das 336 (trezentos e trinta e seis) práticas, necessárias ao atendimento do Nível 3 do Modelo CMMI – Dev 1.2, apenas 33 (trinta e três) práticas foram avaliadas como pontos fracos.

Isso representa o índice de **90,18%** de conformidade às práticas do modelo CMMI.

A avaliação do grau de institucionalização do processo, no entanto, não contou com resultados significativos. Nessa avaliação, os 04 (quatro) projetos de software (Projetos Piloto) iniciaram em momentos diferentes, atendendo versões diferentes da última liberação do PD-DATAPREV (versão 1.1), o que impossibilitou que se avaliasse em igualdade de condições, o nível de conhecimento das equipes dos projetos.

De todo modo foi possível observar que todos os participantes, das 04 (quatro) equipes de projeto, demonstraram, ao final do processo, bom nível de conhecimento da versão do PD-DATAPREV por eles utilizada.

6. Características Inovadoras

O Projeto CMMI-DTP tem sua justificativa pautada na inovação de processo, um tipo de inovação que se caracteriza pela alteração de processo que provoca uma mudança importante e radical (DAVENPORT, 1994).

Nessa linha, o Projeto CMMI-DTP promoveu uma mudança radical no processo de desenvolvimento e manutenção de software da DATAPREV, com destaque a: incorporação da qualidade em todo o processo, o alinhamento às boas práticas da engenharia de software e o foco na melhoria contínua. Os resultados iniciais, obtidos com a institucionalização do processo, já geraram depoimentos positivos por parte dos usuários do processo (equipes de projeto e manutenção) indicando ganhos de produtividade e facilidade de acesso à informação.

7. Conclusão e Perspectivas Futuras

Os resultados obtidos ao longo do Projeto CMMI-DTP serviram como insumo para incorporação das melhorias no Processo de Desenvolvimento de Software da DATAPREV, denominado de PD-DATAPREV. Com o encerramento do Projeto CMMI-DTP foram contabilizadas **149** (cento e quarenta e nove) melhorias, o que deu origem à versão 1.1 (Figura 3) que foi submetida à avaliação SCAMPI B (não oficial).

A expectativa é de que o processo de institucionalização do PD-DATAPREV, agora sob a responsabilidade das áreas funcionais de qualidade e escritório de projetos, continue a evoluir, tornando-se, no prazo máximo de 01 (um) ano, o processo padrão de desenvolvimento e manutenção de software em todas as unidades de organizacionais da DATAPREV.

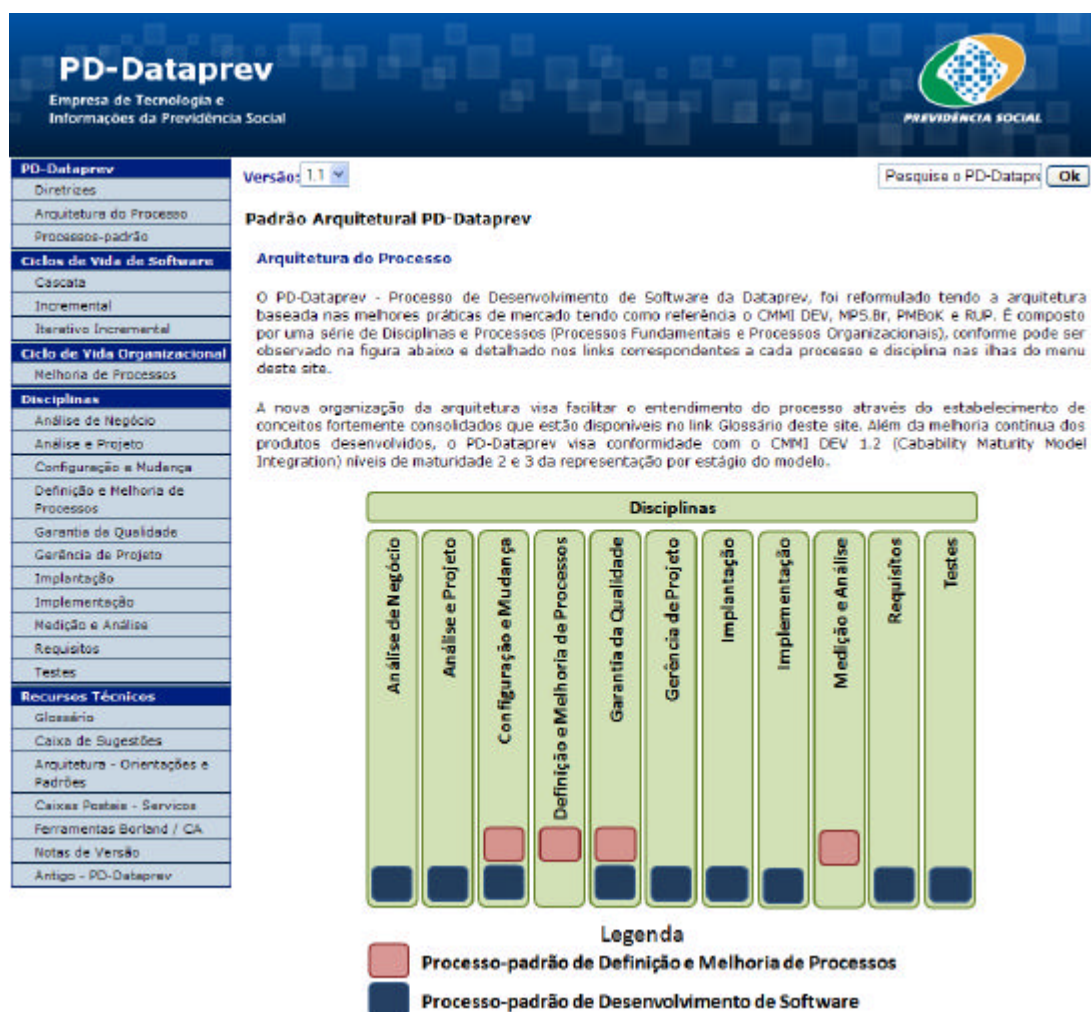


Figura 3. PD-DATAPREV

Referências Bibliográficas

Chrissis, M.B; Konrad, M. e Shrum, S. (2006) "Cmmi - Guidelines For Process Integration and Product Improvement". Addison Wesley.

Davenport, T.(1994) Reengenharia de Processos. Rio de Janeiro: Editora Campus.

ISO/IEC, 2008, "Systems and software engineering – Software life cycle processes", The International Organization for Standardization and the International Electrotechnical Commission, v. ISO/IEC 12207:2008.

SEI, 2006, CMMI for Development (CMMI-DEV), Version 1.2, Technical Report CMU/SEI-2006-TR-008. Pittsburgh, Software Engineering Institute (SEI), Carnegie Mellon University, 2006.

SEI–Software Engineering Institute. Carnegie Mellon,, Disponível em <http://www.sei.cmu.edu/>. Acesso em out 2010.

Apoio Ferramental de Software Livre para Implementação do Processo de Medição do Modelo MPS.BR

Bernardo José da Silva Estácio¹, Wallace Michel Pinto Lira², Sandro Ronaldo Bezerra Oliveira^{1,2}

¹Faculdade de Computação – Universidade Federal do Pará (UFPA) – Rua Augusto Corrêa, 01 - Guamá - 91501-970 - Belém - PA – Brasil

²Programa de Pós-Graduação em Ciência da Computação (PPGCC) – Universidade Federal do Pará (UFPA) - Rua Augusto Corrêa, 01 – Guamá – Belém - PA - Brasil

{wallace.lira, bernardojses}@gmail.com, srbo@ufpa.br

Abstract. *This paper presents a free software tool to support the Measurement process defined by MPS.BR. To this end, there will be presented the objectives, justification and expected impact of this project.*

Resumo. *Este trabalho apresenta uma proposta de uma ferramenta de software livre para apoio à implementação do processo de Medição do modelo MPS.BR. Com este intuito, serão apresentados os objetivos, justificativa e impacto esperado causado pelo projeto.*

1. Introdução

O sucesso de uma organização depende diretamente do serviço ou produto apresentado.

A qualidade deste serviço ou produto, então, é um fator determinante de sucesso. No entanto, é observado que a qualidade do produto ou serviço de uma organização, só é plenamente obtida se os processos das organizações aderirem às normas de qualidade [Oliveira, 2010].

Para suprir esta demanda, a comunidade internacional de desenvolvimento de software possui uma preocupação crescente com a Qualidade do Processo de Software, como se pode observar com o surgimento e manutenção de modelos de maturidade como o DEV-CMMI [SEI, 2010] e normas como a ISO/IEC 15504 [ISO/IEC, 2003]. No entanto, estas abordagens não são compatíveis com a realidade nacional, por possuírem custos de implementação superiores à capacidade de investimento das micro, pequenas e médias empresas brasileiras.

Com o objetivo de suprir esta carência surgiu, então, o MPS.BR (Melhoria do Processo de Software Brasileiro) [Softex, 2009] o qual foi concebido com o objetivo de promover melhoria de processo no mercado brasileiro com custo acessível. Este programa originou o Modelo de Referência do MPS.BR, ou MR-MPS [Softex, 2009].

Neste contexto, surgiu o Projeto SPIDER (www.spider.ufpa.br) [Oliveira, 2010], o qual tem como objetivo apresentar um levantamento de ferramentas de software livre adequadas para implementar as melhorias propostas pelo MR-MPS.

Entre os processos descritos pelo MR-MPS, pode-se observar o de Medição.

Este processo tem como propósito “coletar, armazenar, analisar e relatar os dados relativos aos produtos desenvolvidos e aos processos implementados na organização e em seus projetos, de forma a apoiar os objetivos organizacionais” [Softex, 2009]. Este trabalho visa apresentar a proposta do Projeto SPIDER para o processo de Medição.

O artigo está organizado da seguinte forma: na seção 2 constam os objetivos e as justificativas do projeto; na seção 3 é descrita a metodologia de execução do trabalho; a seção 4 aborda alguns dos resultados obtidos através da execução deste projeto; na seção 5 é apresentada a aplicabilidade dos resultados; na seção 6 são apresentados os impactos e as características que tornam este projeto inovador; finalmente, a seção 7 descreve as conclusões deste trabalho, além de apresentar possíveis trabalhos futuros.

2. Objetivos e Justificativa

Este artigo tem, como objetivo, apresentar uma ferramenta de apoio ao processo Medição aderente às exigências constantes no MR-MPS. O desenvolvimento de uma nova ferramenta foi necessário pois não foram encontradas ferramentas de software livre totalmente aderentes às recomendações constantes no MR-MPS para o processo de Medição.

O software resultante deste projeto deve ter a capacidade de funcionar tanto isoladamente, quanto de forma integrada com ferramentas de apoio a outros processos, trabalhando de forma conjunta e unificada, quando necessário. Esta característica é importante para assegurar compatibilidade estratégica com o Projeto SPIDER.

A utilização de ferramentas de software livre tem como objetivo diminuir os custos e o tempo de implantação do MPS.BR nas organizações. Adicionalmente, permite que as mesmas tenham controle sobre o código fonte das ferramentas utilizadas possibilitando a adequação do software às suas necessidades. Estas são algumas das liberdades garantidas aos usuários ao utilizar software livre, conforme descrito por GNU (2011).

A utilização de ferramentas deste tipo, assim como a capacidade de integração a ferramentas adequadas a outros processos do MR-MPS, garante que este projeto esteja alinhado aos objetivos estratégicos do Projeto SPIDER.

O desenvolvimento deste trabalho, portanto, está intimamente relacionado às necessidades estratégicas do Projeto SPIDER. A justificativa de projetos nesta linha de pesquisa é reforçada por Oliveira (2010):

- 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 conglomerados estrangeiros;
- Baixa quantidade de empresas na região Norte diretamente relacionadas com o desenvolvimento de software. O cenário é projetado a partir de micro e pequenas empresas de baixa competitividade;
- Os resultados significativos obtidos por algumas empresas públicas e privadas do setor de Tecnologia da Informação no estado do Pará.

Finalmente, este projeto visa melhorar o cenário da implantação do processo de Medição nas organizações desenvolvedoras de software da região norte do Brasil. Onde existem apenas treze organizações avaliadas positivamente pelo programa. O nível mais alto implementado é o F, com duas organizações.

3. Metodologia de Execução

A execução do projeto foi realizada pelo coordenador do Projeto SPIDER em conjunto com integrantes dedicados ao processo de Medição do MR-MPS. Estes integrantes foram alocados no Laboratório de Pesquisa do Projeto SPIDER. Neste ambiente, estão alunos de graduação e pós-graduação da Faculdade de Computação da Universidade Federal do Pará integrantes do Projeto SPIDER. Foi observado que a alocação em um mesmo ambiente facilitou a troca de experiência com integrantes de outros subprojetos.

Em seguida, foi elaborado o planejamento do projeto pelo coordenador no início das atividades associadas a este trabalho. O planejamento elaborado dividiu responsabilidades pela gestão técnica do trabalho desenvolvido de acordo com as habilidades individuais dos pesquisadores dedicados ao processo de Medição no contexto do Projeto SPIDER. O planejamento do projeto definiu marcos de projeto e metas para o desenvolvimento do trabalho, assim como o acompanhamento destes indicadores.

Adicionalmente, foram elicitados requisitos a partir: das recomendações do MRMPS para medição, em conjunto com a metodologia GQM, acrônimo para *Goal- Question-Metric* [Basili *et.al.*, 1994]; de visitas a empresas avaliadas pelo MPS.BR e que possuem institucionalizados em seus processos as boas práticas de Medição. A metodologia GQM foi escolhida por ser a mais difundida atualmente para o processo de Medição.

Durante o desenvolvimento do software proposto, foi realizado o mapeamento entre as funcionalidades da ferramenta e os resultados esperados do MR-MPS com o objetivo de validar o trabalho desenvolvido. Foi realizada também uma avaliação pelos membros do Projeto SPIDER com este objetivo. Vale salientar que a equipe do Projeto SPIDER conta com profissionais certificados pela SOFTEX.

Adicionalmente, a ferramenta se encontra em uso no laboratório do Projeto SPIDER para coleta de dados de seu uso para futura validação.

4. Resultados Obtidos

A equipe do Projeto SPIDER apresentou os seguintes resultados obtidos ao final do ciclo 2010 relativos a este projeto:

- Um fluxo de atividades aderente às recomendações do MR-MPS para Medição. Este fluxo consta na figura 1;
- Uma nova ferramenta concebida e elaborada agrupando as recomendações do MR-MPS para Medição e o paradigma GQM. Esta ferramenta foi batizada de Spider-Mplan;
- A Spider-MPlan foi registrada como software livre sob a licença GNU/GPL (GNU, 2011), o que garante as liberdades associadas a este tipo de software aos usuários;
- Manual da Ferramenta para a ferramenta Spider-Mplan;
- Publicações nos seguintes eventos nacionais em 2010: Escola Regional de Informática (ERIN); e no Workshop de Software Livre (WSL);
- Um trabalho de Conclusão de Curso na Faculdade de Computação da Universidade Federal do Pará;

- Formação de Recursos Humanos aptos a trabalhar na área de qualidade de software.

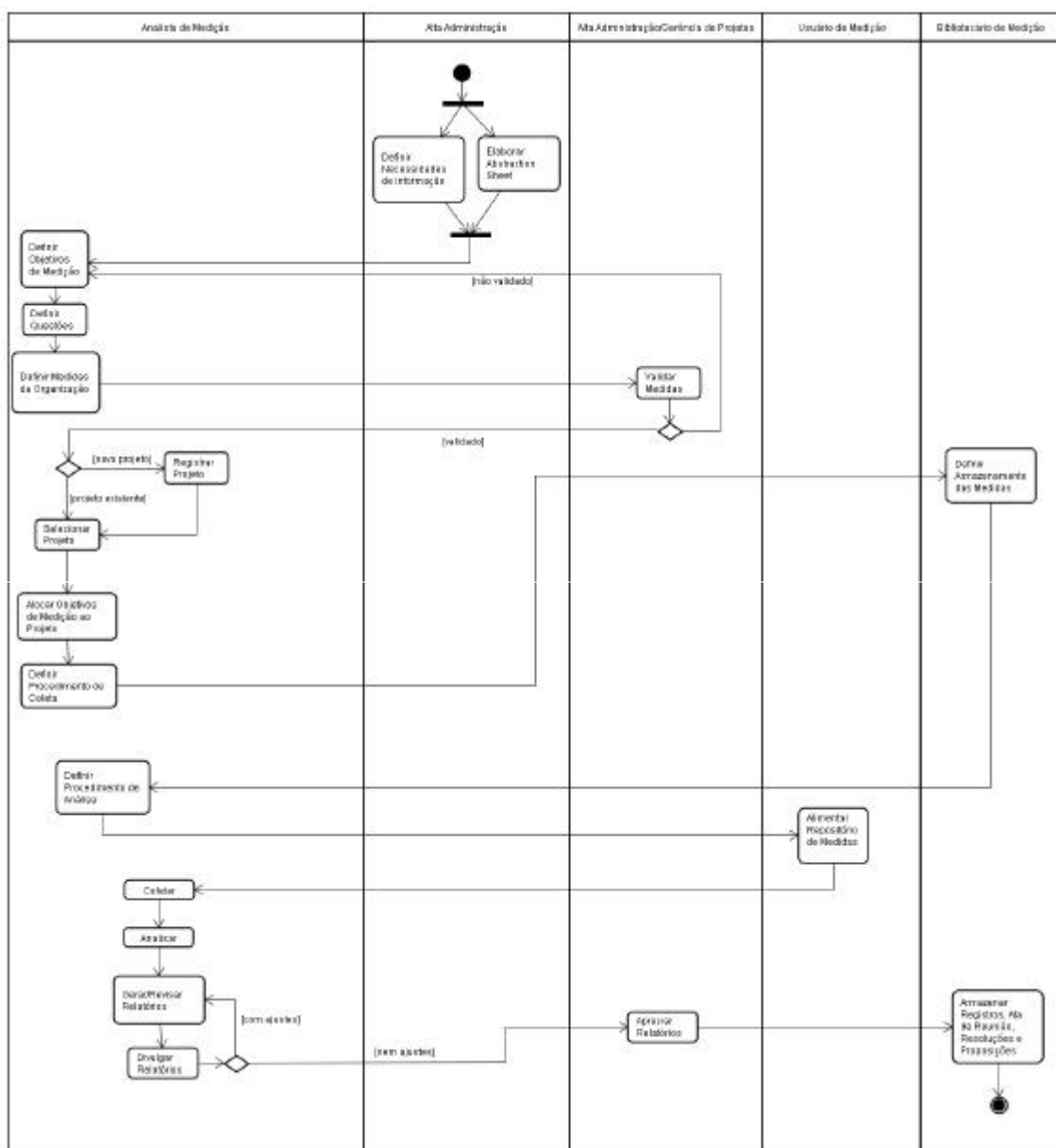


Figura 1. Fluxo de Atividades do Processo de Negócio da Spider-MPlan

5. Aplicabilidade dos Resultados

Espera-se alterar o cenário da qualidade de software nas organizações, em especial na região Norte, a qual tem poucas empresas avaliadas positivamente pelo programa MPS.BR em relação às outras regiões do Brasil [Softex, 2011]. Aspectos importantes do projeto podem ser citados para demonstrar a abrangência e os impactos positivos gerados:

- Foram capacitados recursos humanos para trabalho envolvendo qualidade de software;
- A Spider-MPlan, ferramenta criada neste projeto, possui interface *online*, facilitando a troca de informações entre as pessoas nas organizações onde for implantada;
- Aplicação do ferramental concebido em uma empresa parceira do Projeto SPIDER, a Cobra Tecnologia, com foco na implementação do processo de Medição.

6. Características Inovadoras

A aderência completa às recomendações para o processo de Medição do MR-MPS e a utilização do paradigma GQM, em adição à licença de uso livre, torna a Spider-MPlan uma ferramenta única no contexto nacional. Este fato é discutido no trabalho de conclusão de curso originado a partir deste trabalho, [Estácio, 2010].

Adicionalmente, possuir uma interface *web* aumenta consideravelmente a portabilidade e usabilidade da ferramenta, haja vista que é necessário que apenas um computador possua a ferramenta instalada: outros usuários necessitam apenas de um navegador.

7. Conclusões e Perspectivas Futuras

A Spider-MPlan tem o intuito de apoiar a implementação do MPS.BR, especificamente a do processo de Medição levando em consideração as recomendações do paradigma GQM.

Possíveis trabalhos futuros incluem:

- O refinamento do fluxo de atividades proposto na ferramenta;
- A integração da Spider-MPlan a outras ferramentas livres de apoio a outros processos do MR-MPS, conforme os objetivos estratégicos do Projeto SPIDER;
- A criação de um Manual de Implementação do processo de Medição do MRMPS, utilizando a Spider-MPlan, analogamente aos manuais de implementação criados para outros processos pela equipe do Projeto SPIDER;
- Descrição do Estudo de Caso da utilização da ferramenta no ambiente da Cobra Tecnologia.

Vale ressaltar que as boas práticas apresentadas neste trabalho encontram-se em institucionalização no Projeto SPIDER, com o objetivo de validar a efetividade do fluxo de atividades da ferramenta Spider-MPlan.

8. Agradecimentos

Este trabalho recebeu apoio financeiro do PIBIC/UFPA através do Edital PIBIC/CNPq/FAPESPA/UFPA de 2009.

Referências Bibliográficas

Basili, V.; Caldiera, G.; Rombach, H. (1994) "GQM: Goal Question Metric Paradigm". Encyclopedia of software Engineering.

Estácio, B. J. da S., Valente, K. S. da M. (2010) "Spider-Mplan: Um Ferramental para Apoio ao Processo de Medição constante em Norma e Modelos de Qualidade do Processo de Software", Trabalho de Conclusão de Curso - Faculdade de Computação - ICEN/UFPA, Orientador Prof. Sandro Bezerra, Belém-PA.

GNU Project. "General Public License". Disponível em: <http://www.gnu.org/licenses/gpl.html>. Acessado em 14 de março de 2011.

the International Organization for Standardization/ the International Electrotechnical Commission (2004) "ISO/IEC 15504-1:2004 Information technology Process assessment Part 1: Concepts and vocabulary". Geneve.

Oliveira, S. R. B. et al. (2010) "SPIDER - Um Suite de Ferramentas de Software Livre de Apoio à Implementação do modelo MPS.BR". Anais do VIII Encontro Anual de Computação – ENACOMP 2010. Catalão, Goiás.

SEI – Software Engineering Institute (2010) "Capability Maturity Model Integration for Development – CMMI-Dev". Versão 1.3.

SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2009) "MPS.BR - Guia Geral".

SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2011) "Total de organizações com Avaliação MPS (vigentes ou não)".

Apoio Sistematizado de Software Livre para Implementação do Processo de Gerência de Requisitos do MPS.BR

Ewelton Yoshio Chiba Yoshidome², Alexandre Brito Cardias Junior¹, Luciana Neves Bentes¹, Sandro Ronaldo Bezerra Oliveira^{1, 2}

¹Faculdade de Computação – Universidade Federal do Pará (UFPA) – Rua Augusto Corrêa, 01 - Guamá - 91501-970 - Belém - PA – Brasil

²Programa de Pós-Graduação em Ciência da Computação – Universidade Federal do Pará (UFPA) – Rua Augusto Corrêa, 01 - Guamá - 91501-970 - Belém - PA - Brasil

{ewelton.yoshio, alexandrecardias, lufpa06}@gmail.com, srbo@ufpa.br

Abstract. *This paper proposes a systematic support to the Requirement Management process of MPS.BR model, Brazilian Software Process Improvement Program, exclusively using free software tools, reviewing the adherence to the maturity model. In this context, it will discuss about the objectives, justifications and the goals achieved in this project.*

Resumo. *Este trabalho apresenta uma proposta de apoio sistematizado ao processo de Gerência de Requisitos do MPS.BR, Programa de Melhoria do Processo de Software Brasileiro, utilizando exclusivamente ferramentas de software livre, analisando a aderência ao modelo de maturidade. Neste contexto, serão discutidas os objetivos, as justificativas e as metas alcançadas neste projeto.*

1. Introdução

As organizações enfrentam constantes mudanças motivadas por diversos fatores. A alta concorrência exige habilidades para fornecer produtos e serviços mais inovadores. As exigências dos clientes são outros fatores causadores de mudanças, pois as organizações devem se preocupar em criar produtos e prover serviços que satisfaçam as necessidades dos seus clientes para garantir confiança e satisfação [Softex, 2009]. Neste contexto, foi desenvolvido pela SOFTEX (Associação para Promoção da Excelência do Software Brasileiro) o programa MPS.BR (Melhoria do Processo de Software Brasileiro) que tem como foco principal a implementação da melhoria dos processos de software em empresas brasileiras. Este programa divulga o modelo de referência MR-MPS [Softex, 2009].

Entre os processos presentes no MR-MPS está o processo de Gerência de Requisitos (GRE), que tem como propósito efetuar o gerenciamento dos requisitos e componentes do produto, além de ser o responsável em detectar inconsistências entre os requisitos, os planos do projeto e produtos de trabalho [Softex, 2009]. Ainda, segundo Pressman (2006), “a gestão de requisitos é um conjunto de atividades que ajudam a equipe de projeto a identificar, controlar e rastrear requisitos e modificações de requisitos em qualquer época, a medida que o projeto prossegue”.

O MPS.BR não descreve a forma que o processo deve ser implementado, mas sim o quê a empresa desenvolvedora de software necessita para alcançar o nível de maturidade desejado. A organização tem livre arbítrio para definir a forma de execução de seu processo, porém, o uso de ferramentas para sistematizar as atividades presentes no processo pode aumentar o nível de produtividade da organização, além da possibilidade de maior rapidez na implementação do processo.

Atualmente, presencia-se um crescimento considerável no desenvolvimento de ferramentas não proprietárias, devido à popularização da Internet [Gnu, 2010]. Estas ferramentas são denominadas de software livre pelo fato da possibilidade de qualquer pessoa ser capaz de utilizá-las e modificá-las. Software livre é um software o qual qualquer usuário possui a permissão de utilização, cópia e distribuição, de forma gratuita ou paga, podendo se encontrar em sua forma original ou modificada [Hexsel, 2002].

Além desta seção introdutória, este artigo apresenta na seção 2 os objetivos e as justificativas do projeto descrito neste trabalho. A seção 3 descreve a metodologia de utilizada para a execução deste projeto, em seguida, a seção 4 mostra os resultados obtidos a partir deste trabalho, desenvolvidos no projeto SPIDER. Na seção 5 tem-se uma breve apresentação da aplicabilidade dos resultados, assim como a seção 6 descreve as inovações proporcionadas por este projeto. Finalmente, na seção 7 são apresentadas as conclusões e os trabalhos futuros referentes a este projeto.

2. Objetivos e Justificativa

Este trabalho tem como foco o desenvolvimento de uma metodologia de uso de ferramentas livres de gestão de requisitos, com o objetivo de sistematizar o processo de Gerência de Requisitos na unidade organizacional, contemplando todos os resultados esperados deste processo segundo o modelo MPS.Br. Resultado esperado “é um resultado observável do sucesso do alcance do propósito do processo” [ISO/IEC 2008].

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 Gerência de Requisitos.

Ao se definir uma sistematização do processo de Gerência de Requisitos com o uso de ferramentas, isto pode implicar em uma maior facilidade de uso e produtividade do processo, proporcionando redução de esforço e tempo, devido à diminuição da necessidade de documentação, agilizando o processo como um todo [Oliveira, 2009].

Além do aumento da eficiência na produtividade da organização, o uso de ferramentas poderá afetar na redução de tempo para a implementação do processo na unidade organizacional.

Os resultados gerados durante este trabalho poderão servir de base de conhecimento para projetos que queiram seguir a mesma linha de pesquisa. Além disso, as informações produzidas podem ser utilizadas por profissionais da área como material de apoio para a capacitação e certificação em modelos de qualidade.

O alcance dos objetivos deste projeto está alinhado às metas propostas no escopo do Projeto SPIDER (www.spider.ufpa.br), o qual tem como objetivo criar um *suite* de ferramentas de software livre aderente ao MR-MPS [Oliveira, 2010]. Entende-se *suite* como sendo um conjunto de softwares integrados, com o propósito de facilitar e agilizar a comunicação e troca de informação.

3. Metodologia de Execução

O planejamento da execução do projeto foi efetuado juntamente com o coordenador, com o objetivo de definir o escopo, as metas e o cronograma de trabalho. Além disso, foram estabelecidas reuniões periódicas com todos os pesquisadores envolvidos no Projeto SPIDER para efetuar uma avaliação coletiva dos resultados que eram gerados durante o desenvolvimento do projeto.

Na primeira etapa do projeto, foram feitos estudos do modelo MPS.BR, consultando seus guias disponibilizados no site da Softex (<http://www.softex.br/>), além de pesquisas bibliográficas e entrevistas com profissionais da área de qualidade, os quais relatavam experiências de implementação e avaliação em empresas desenvolvedoras de software. O objetivo deste estágio é a elaboração de uma base de conhecimento, conceitual e técnica, do escopo do trabalho para a execução das metas seguintes.

Em seguida, foi feita uma pesquisa com o levantamento de ferramentas livres de gerência de requisitos disponíveis no mercado, cada ferramenta encontrada foi analisada e testada verificando a existência de funcionalidades que implementassem os resultados esperados do processo de Gerência de Requisitos, além de analisar aspectos como portabilidade, linguagem de programação utilizada, possibilidade de integração com outras ferramentas.

Após a realização dos mapeamentos entre as ferramentas e os resultados esperados, durante uma reunião com toda a equipe do projeto SPIDER, foi definida a ferramenta OSRMT (*Open Source Requirement Mangement Tool*) [Sourceforge, 2010], disponível em <http://sourceforge.net/projects/osrmt/>, como a mais adequada às necessidades do modelo MPS.BR, não havendo a necessidade de mudanças em seu código fonte.

Também foram definidas outras ferramentas como o Mantis [Mantis, 2010] (<http://www.mantisbt.org/>) e Trac [Trac, 2011] (<http://trac.edgewall.org/>) para efetuar o controle das mudanças e solicitações de revisões dos requisitos. Para a definição de critérios objetivos, foi utilizada a ferramenta Spider-CL [Barros, 2010] (ferramenta desenvolvida pelo Projeto SPIDER para a elaboração e aplicação de *checklists*), disponível em <http://www.spider.ufpa.br/index.php?id=resultados>.

Depois de estabelecido o *kit* de ferramentas para a implementação do processo de Gerência de Requisitos, foi desenvolvida uma *guideline* que propunha a metodologia de uso desse conjunto de ferramentas com funcionalidades aderentes ao MR-MPS. A metodologia foi desenvolvida alinhada às metodologias que estavam sendo criadas por outros trabalhos do escopo do Projeto SPIDER.

Finalmente, espera-se que a solução proposta seja implementada em processos de unidades organizacionais desenvolvedoras de softwares, com o objetivo de avaliar a qualidade da metodologia proposta em um ambiente real de trabalho. Após isso, executar ajustes e refinamentos na metodologia a partir de *feedbacks* provenientes da organização.

4. Resultados Obtidos

Até o presente momento, o Projeto SPIDER, no que se refere à sistematização do processo de Gerência de Requisitos do MPS.Br, foi capaz de produzir os seguintes resultados:

- Uma publicação de artigo no Workshop de Engenharia de Requisitos (WER), como pode ser visto em <http://spider.ufpa.br/index.php?id=publicacoes>;

- Um trabalho de conclusão de curso em Bacharelado em Sistema de Informação da Faculdade de Computação da UFPA, o que se pode evidenciar a formação de recursos humanos na área de melhoria do processo de software;
- Uma parceria de consultoria de implementação nos níveis de maturidade G e parte do F na Diretoria de Tecnologia de Informação e Comunicação, localizado no Instituto Federal do Pará (IFPA), utilizando o conjunto de ferramentas livres proposto na metodologia;
- Um *guideline* contendo informações sobre instalação e configuração das ferramentas utilizadas na metodologia adotada, além de descrever, detalhadamente, as sugestões propostas na metodologia;
- Um documento contendo o mapeamento das funcionalidades ou serviços que implementam os resultados esperados do processo de Gerência de Requisitos no modelo de qualidade;
- Capacitação de profissionais provenientes da certificação do curso de introdução do MPS.Br.

5. Aplicabilidade dos Resultados

Durante o desenvolvimento deste projeto, foram analisados conteúdos bibliográficos provenientes da literatura especializada e experiências provenientes de profissionais da área. Por esse motivo, os resultados obtidos neste trabalho possuem grande relevância acadêmica e comercial. Para evidenciar isto, pode-se citar:

- Uma parceria de consultoria na implementação do processo de software no Instituto Federal do Pará, onde será sistematizado, além do processo de Gerência de Requisitos, os processos de Gerência de Projetos, Gerência de Configuração e Garantia da Qualidade;
- A contínua participação de estudantes em projetos de melhoria de processos, provenientes da produção de conteúdo técnico-científico pelo Projeto SPIDER os quais são disponibilizados para comunidades acadêmicas e comerciais.

6. Características Inovadoras

Este trabalho destaca-se pelo fato de utilizar um conjunto de ferramentas de software livre para o apoio à implementação do processo de Gerência de Requisitos do modelo MPS.BR em unidades organizacionais.

Outra característica é a facilidade de adaptação das ferramentas às necessidades do MPS.BR, devido ao fato de ser *open source*, permitindo que o software seja modificado às exigências do modelo brasileiro.

Um aspecto importante que acompanha a sistematização do processo é a possibilidade do aumento da eficiência na gerência de requisitos, devido à minimização de burocracia.

7. Conclusões e Perspectivas Futuras

O conjunto de ferramentas avaliadas e selecionadas durante este trabalho, possibilitam a implementação sistêmica do processo de Gerência de Requisitos, satisfazendo totalmente todos os resultados esperados do MR-MPS, essa contemplação só será alcançada desde que as ferramentas sejam utilizadas de forma correta e planejada, conforme descrito na metodologia.

Este trabalho proporciona uma importante alternativa para empresas desenvolvedoras de software que desejam implementar o processo de Gerência de Requisitos no nível de maturidade G do MPS.Br, principalmente para empresas de pequeno e médio porte as quais, geralmente, não possuem recursos suficientes para investirem em ferramentas proprietárias.

É importante ressaltar que a metodologia proposta neste trabalho não define um processo de Gerência de Requisitos que deve ser institucionalizado na organização, o objetivo deste trabalho é apresentar boas práticas de uso do conjunto de ferramentas para atender as sugestões propostas do MR-MPS.

Como um projeto futuro, pretende-se desenvolver uma metodologia para o processo de Desenvolvimento de Requisitos (DRE) do MPS.BR, utilizando o mesmo *kit* de ferramentas propostos neste trabalho. Esta nova metodologia funcionará como uma extensão da metodologia proposta neste trabalho.

Também, planeja-se efetuar a integração das ferramentas propostas no escopo do Projeto SPIDER, com o intuito de desenvolver o *suite* de ferramentas de software livre para o apoio ao modelo. Inicialmente, a integração agregaria os conjuntos de ferramentas de todos os processos presentes até o nível de maturidade E do modelo MPS.BR.

Por fim, tem-se como projeto, a consultoria na implementação do processo de Gerência de Requisitos, utilizando o conjunto de ferramentas propostas neste trabalho, na Diretoria de Tecnologia de Informação e Comunicação, o qual faz parte do Instituto Federal do Pará.

Além do processo de Gerência de Requisitos, será feito a consultoria dos processos de Gerência de Projeto, Gerência de Configuração e Garantia da Qualidade utilizando outros trabalhos desenvolvidos no Projeto SPIDER, implementando parte do nível F do MPS.BR.

8. Agradecimentos

O desenvolvimento deste trabalho está sendo apoiado financeiramente pelo CNPq, a partir da concessão de bolsa de mestrado.

Referências Bibliográficas

Associação para Promoção da Excelência do Software Brasileiro. MPS.BR – Guia Geral:2009”. Disponível em: www.softex.br. Último acesso em março/2011.

Barros, R. S., Oliveira, S. R. B. (2010) “Spider-CL: Uma Ferramenta de Apoio ao Uso de Critérios Objetivos no Contexto da Qualidade de Software”. II Escola Regional De Informática. Manaus.

GNU Operating System (2010) “Free Software Foundation”. Disponível em: www.gnu.org. Último acesso em março/2011.

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.

International Organization for Standardization/ International Eletrotechnical Commission. ISO/IEC 12207 Systems and software engineering – Software life cycle processes. Geneve: ISO, 2008.

Mantis Bug Tracker. Disponível em: <<http://www.mantisbt.org/>>. Acesso em: março/2011.

Oliveira, S. R. B. et al. (2010) “SPIDER - Um Suite de Ferramentas de Software Livre de Apoio à Implementação do modelo MPS.BR”. Anais do VIII Encontro Anual de Computação – ENACOMP 2010. Catalão

Pressman, R. S. Engenharia de software. Tradução Rosângela Velloso Penteado. São Paulo: McGraw-Hill, 2006.

Sourceforge. Open Source Requirements Management Tool. Disponível em: <http://sourceforge.net/projects/osrmt/>. Último acesso em novembro/2010.

Trac - Integrated SCM & project management. Disponível em: <<http://trac.edgewall.org/>>. Acesso em: março/2011.

SPIDER: Uma Proposta de Solução Sistêmica de um SUITE de Ferramentas de Software Livre de Apoio à Implementação do Modelo MPS.BR

Sandro Ronaldo Bezerra Oliveira^{1,2}, Ewelton Yoshidome², Wallace Lira², Júlio Furtado², Jñane Neiva², Marília Paulo², Renan Barros¹, Maurício Ronny², Leonardo Mezzomo², Bernardo Estácio¹, Kaio Valente¹, Gleise Baldez¹, Pedro Afonso¹, Kleverton Macedo¹, Fabrício Alho², Sivaldinho Oliveira¹

¹Faculdade de Computação - Universidade Federal do Pará (UFPA) - Rua Augusto Corrêa, 01 - Guamá - 91501-970 - Belém - PA - Brasil

²Programa de Pós-Graduação em Ciência da Computação (PPGCC) - Universidade Federal do Pará (UFPA) - Rua Augusto Corrêa, 01 - Guamá - Belém - PA - Brasil

srbo@ufpa.br, {ewelton.yoshio, wallace.lira, julio.furtado, jineiva, mariliapaulo, renansalesbarros, maaalk, leopossamai, bernardojses, ksmvalente, esielg.inha, afonso.baco, kleverton.macedo}@gmail.com, fabricioalho@ig.com.br, sivaldo.oliveira@yahoo.com.br

Abstract. This paper presents a proposal of a free software tool suite to support the implementation of MPS.BR model. This way, they will be discussed the goals, reasons and marks of this project in the quality improvement program for organization processes.

Resumo. Este trabalho apresenta uma proposta de um suite de ferramentas de software livre de apoio à implementação do modelo MPS.BR. Para tal, serão discutidos os objetivos, a justificativa e as metas do projeto no contexto do programa de melhoria da qualidade dos processos organizacionais.

1. Introdução

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

Visando a qualidade do processo, o modelo MPS.BR - Modelo de Processo de Software Brasileiro [Softex, 09], se propõe a ser um guia evolucionário para o aperfeiçoamento e adequação de processos organizacionais. O Projeto 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. Não é objetivo do projeto 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 [Softex, 09].

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, faz-se necessária a utilização de ferramentas de software para possibilitar a implantação do modelo MPS.BR 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 [Gnu, 08]. 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, 08], criam políticas de incentivo para difusão destes softwares. A definição de software livre (*free software*) é o software 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, 02]. A possibilidade de modificações implica que o código fonte esteja disponível (*open source*) e o software deve poder ser executado em um sistema operacional também livre.

Além desta seção introdutória, este artigo apresenta na seção 2 os objetivos e as justificativas do projeto descrito neste trabalho. A seção 3 descreve a metodologia de execução do trabalho, já a seção 4 aborda alguns dos resultados já obtidos com o desenvolvimento do projeto SPIDER. Na seção 5 tem-se uma breve apresentação da aplicabilidade dos resultados, assim como na seção 6 as características que enquadram o projeto como inovador são discutidas. Por fim, a seção 7 apresenta as conclusões do trabalho descrito, bem como relata alguns trabalhos futuros.

2. Objetivos e Justificativa

O projeto SPIDER (www.spider.ufpa.br) 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 inicialmente, dos níveis de maturidade G (parcialmente gerenciado) e F (gerenciado) do modelo MPS.BR, já que estes níveis são de maior

complexidade para a implementação da maturidade organizacional, pois correspondem aos níveis iniciais dentro do modelo em questão [Oliveira, 2010].

Em decorrência deste levantamento, pretende-se especificar e desenvolver um SUITE^{1 4} de ferramentas a fim de propiciar um uso mais integrado das suas funções/operações disponíveis, de modo a apoiar a implementação dos processos do modelo MPS.BR, obedecendo o fluxo de dependência proposto por este modelo de qualidade de processo. Visa, ainda, apresentar alternativas viáveis com relação a ferramentas de software para auxiliar a implementação do modelo MPS.BR 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 de pesquisas em parcerias, que são mantidos com a institucionalização deste projeto, com núcleos de renome na área de Engenharia de Software, como o Centro de Informática da Universidade Federal de Pernambuco, através do projeto de pesquisa *ImPProS* - Um Ambiente de Implementação de Processos de Software (www.cin.ufpe.br/imppros), e com o núcleo ProQualiti - Núcleo de Estudos em Engenharia e Qualidade de Software (www.proqualiti.org.br), órgão destinado à finalidade de congrega profissionais, estudantes e empresários atuantes nas áreas de Engenharia e Qualidade de Software.

Este projeto possui também como foco corroborar no atendimento dos objetivos listados no projeto "Rede Paraense de Pesquisa em Tecnologias de Informação e Comunicações (TIC)" no sub-projeto "Laboratório de Tecnologia de Software" aprovado e financiado pela FAPESPA - Fundação de Amparo à Pesquisa do Estado Pará, e desenvolvido pelo Laboratório de Engenharia de Software da UFPA - Universidade Federal do Pará.

Como melhor caracterização do cenário descrito anteriormente, pode-se listar as seguintes justificativas que enfatizam o desenvolvimento do projeto descrito [Oliveira, 2010]:

- 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 conglomerados estrangeiros;
- ? Baixa quantidade de empresas na região Norte diretamente relacionadas com o desenvolvimento de software. O cenário é projetado a partir de micro e pequenas empresas de baixa competitividade, estando completamente alheio à corrida por mercados estrangeiros;
- Notória capacidade das universidades paraenses 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 local;
- Os resultados significativos obtidos por algumas empresas públicas e privadas do setor de Tecnologia da Informação no estado do Pará: a Regional Belém do SERPRO, a PRODEPA - Empresa de Processamento de Dados do Estado do Pará, o CTIC-UFPA e a EquilibriumWeb.

Assim, a este projeto tende modificar esta realidade, reforçando o cenário das empresas locais de desenvolvimento de software, auxiliando na obtenção de certificações nacionais e internacionais do programa de melhoria da qualidade do processo e de produto de software.

3. Metodologia de Execução

A execução do projeto foi orientada pelo coordenador e pelos pesquisadores do projeto e realizada pela equipe do projeto alocado no Laboratório de Pesquisa do Projeto SPIDER, que é composta por alunos de graduação e pós-graduação. Os pesquisadores associados à proposta possuem especialidades distintas e específicas tornando suas contribuições cruciais ao sucesso do projeto.

O coordenador elaborou, inicialmente, um planejamento do projeto, em conjunto com os pesquisadores, e atribuiu para cada pesquisador a responsabilidade pela gestão técnica de uma etapa do projeto. Esta divisão foi baseada nas experiências e qualificações de cada pesquisador.

⁴ 1 Amplo conjunto modular de tecnologias integradas, facilitando a aceleração de fluxo de dados unificados para enfrentar alguns dos desafios de integração mais sérios da organização.

No planejamento inicial, foram definidos indicadores de progresso e resultados esperados, de acordo com o plano geral do projeto. Reuniões foram agendadas pelo gestor técnico da etapa, sendo exigida a presença dos participantes envolvidos na execução das atividades em questão.

Foram alocados recursos humanos do Laboratório de Pesquisa do Projeto SPIDER e equipes temporárias para execução do projeto. Os integrantes das equipes temporárias realizaram atividades condizentes com sua qualificação e *know-how*, visando à transferência de conhecimento. Todos os objetivos da proposta foram desenvolvidos pela equipe alocada para o projeto. Os requisitos foram elicitados em reunião com toda equipe do projeto.

Após o desenvolvimento das soluções propostas, foi realizado um estudo de caso (projeto piloto) a fim de validar o modelo concebido. Este estudo de caso exercitou o modelo, utilizando as novas métricas, processos e ferramentas propostos. Caso fosse necessário, ajustes seriam realizados ao fim do estudo de caso.

Por fim, pretendeu-se aplicar a(s) solução(ões) proposta(s) ao longo do desenvolvimento de um projeto de software, em uma fábrica de software, com o propósito de avaliar a implementação do programa de qualidade a partir do uso de práticas e resultados esperados contidos nos modelos e normas de qualidade para o desenvolvimento de software. Posteriormente, refinar gradativamente o modelo proposto para atender a real demanda deste tipo de organização para o foco do trabalho contido neste projeto de pesquisa.

4. Resultados Obtidos

A partir das pesquisas realizadas pelo grupo de pesquisadores do projeto SPIDER e das lições aprendidas das empresas parceiras do projeto, podem-se listar os seguintes resultados já obtidos:

- ? Concepção e desenvolvimento de uma ferramenta de software livre de apoio ao gerenciamento de listas de verificação (*checklist*), a SPIDER-CL, para subsidiar a geração e uso de critérios objetivos para avaliar requisitos, a gerência do projeto, a garantia da qualidade e a gerência de configuração;
- ? A reestruturação arquitetural e gráfica da ferramenta de software livre de apoio ao processo de avaliação do modelo MPS.BR, a WISE, para prover a sistematização de atividades presentes no Guia de Avaliação do MPS.BR;
- ? Levantamento de ferramentas livre e *open source* para apoio aos processos de Gerência de Projetos, Gerência de Requisitos, Gerência de Configuração e Gerência de Portfólio, onde foram gerados *guidelines* que mapeiam os resultados esperados destes processos aos serviços oferecidos por estas ferramentas. Com base nestes *guidelines*, algumas melhorias funcionais foram propostas e desenvolvidas aos serviços destas ferramentas para o atendimento total das recomendações propostas pelo modelo MPS.BR, fazendo que estes processos fossem contemplados ao *suite* do SPIDER;
- ? Concepção e elaboração funcional e não-funcional de uma ferramenta de apoio aos resultados esperados do processo de Garantia da Qualidade do modelo MPS.BR, a Spider-QA, caracterizando o atendimento parcial ao nível F do modelo, com base, também, nas recomendações existentes na ISO/IEC 12207 [ISO/IEC, 2008], SWEBOK [IEEE, 2004] e CMMI [SEI, 2010];
- ? Desenvolvimento de ferramentas de software livre (Spider-APF, Spider-UCP e Spider-CoCoMo) para apoio a estimativas de projeto no que tange à especificação de Tamanho, Esforço e Custo de um Projeto de Software, com base nos modelos APF - Análise por Pontos de Função, UCP - Use Case Points e CoCoMo - Constructive Cost Model;
- ? Concepção e desenvolvimento de uma ferramenta de software livre de apoio à implementação do processo de Medição do MPS.BR, a Spider-Mplan, tomando como base o modelo proposto por GQM - Goal, Question, Metrics, e as boas práticas recomendadas pelo modelo CMMI;
- ? Concepção e desenvolvimento de uma ferramenta de software livre de apoio à implementação do Guia de Aquisição do MPS.BR, a Spider-ACQ, tomando como base as boas práticas recomendadas pelo CMMI-ACQ;
- ? Com base nestes resultados e nas orientações propostas pelos Guias de Implementação do MPS.BR, foram gerados Manuais de utilização das ferramentas pesquisadas para apoio à implementação do modelo MPS.BR. Estes manuais descrevem detalhadamente o uso dos serviços funcionais providos pelas ferramentas no atendimento dos resultados esperados dos processos listados anteriormente, tendendo a ajudar as Instituições Implementadoras (II) nas suas atividades de consultoria, quanto à metodologia partir pelo uso de ferramentas de software livre;
- ? Foram gerados mapeamentos de integração entre os resultados esperados dos processos dos níveis G e F do MPS.BR para caracterizar a dependência de implementação destes resultados nos processos organizacionais. Este resultado possibilitou a integração funcional entre as ferramentas de apoio, atendendo a um dos objetivos específicos do projeto SPIDER;
- ? Concepção e elaboração de uma ferramenta de apoio à modelagem do processo de software seguindo às especificações promovidas pelo SPEM - Software Process Engineering Metamodel, a Spider-PM, para promover a integração do ferramental de apoio com base nos processos implementados para as organizações aderentes ao programa de melhoria do MPS.BR;

- ? Concepção e elaboração de *frameworks* de processos relacionados à Aquisição, Gerência de Reutilização e Avaliação Integrada de Processos de Software, a fim de proporcionar à comunidade um modelo de processo alinhado e aderente às recomendações do CMMI, MPS.BR e ISO/IEC 12207, que possa ser instanciado ao atendimento específico de cada organização/projeto;
- ? 23 (vinte e três) publicações em eventos científicos e periódicos nacionais e internacionais, como: WER, SBQS, SBES, WSL, WGPS, WAMPS, ERIN, ENACOMP, SEA, SBGames e Revista Traços, como podem ser vistos em <http://spider.ufpa.br/index.php?id=publicacoes>;
- ? 13 (treze) Trabalhos de Conclusão de Curso dos cursos de Bacharelado em Ciência da Computação e Bacharelado em Sistemas de Informação da Faculdade de Computação da UFPA foram orientados no período e contribuíram para a realização do trabalho;
- ? 8 (oito) Dissertações de Mestrado estão em andamento, que já geraram resultados para o projeto. 4 (quatro) destes trabalhos estão com defesas previstas para Junho de 2011;
- ? Foi realizada uma parceria com a fábrica de software da Cobra Tecnologia, implantada no Pará, para transferência de conhecimento dos resultados obtidos com as pesquisas realizadas no Laboratório do Projeto SPIDER. Esta parceria rendeu inúmeros estudos de caso da aplicação dos ferramentais listados anteriormente;
- ? Proposta de parceria entre Projeto SPIDER e o Instituto Federal do Pará (IFPA), em forma de consultoria para implantação da melhoria do processo de software na Diretoria de Tecnologia de Informação e Comunicação da instituição, com a implementação dos processos de Gerência de Projetos, Gerência de Configuração, Gerência de Requisitos e Garantia da Qualidade em conformidade com o MR-MPS.

O projeto e seus sub-projetos foram selecionados para participar dos Ciclos 2009 e 2010 do PBQP-Software (Programa Brasileiro de Qualidade e Produtividade) do MCT (Ministério da Ciência e Tecnologia).

5. Aplicabilidade dos Resultados

Os resultados obtidos com a execução do projeto não podem ser tomados apenas como resultados de uma implementação de ferramental de apoio à melhoria de processos.

Antes do projeto não havia equipe de estudos sobre ferramentais de apoio à implementação oficial MPS.BR no Norte do país e por isso a formação de recursos humanos nesta linha eram definicórias. Vários aspectos importantes do projeto podem ser citados para demonstrar a abrangência e os impactos positivos gerados:

- ? O desenvolvimento deste projeto propiciou a consolidação de um grupo de pesquisa formada por alunos da graduação e pós-graduação (mestrando e doutorando) da UFPA, potencializando a formação de recursos humanos com ênfase na melhoria de processos. A partir disso, um laboratório de pesquisas foi estruturado e institucionalizado na Faculdade de Computação da UFPA a fim de permitir estudos e experimentações dos ferramentais produzidos, com foco no desenvolvimento dos projetos de software;
- ? Os processos implantados na empresa parceira, Cobra Tecnologia, não visaram tão somente a melhoria dos processos da fábrica de software, mas a estruturação da Cobra Tecnologia como um todo. A implantação dos ferramentais produzidos pelo SPIDER, que oferece serviços para a adoção das boas práticas recomendadas pelo MPS.BR, tende a auxiliar na transferência de conhecimento entre as equipes;
- ? Ressalta-se a disseminação da importância da melhoria dos processos na comunidade acadêmica e comercial, que pode ser percebida pela participação contínua de estudantes e profissionais em disciplinas ofertadas pelos membros do grupo no âmbito da Graduação e Pós-Graduação na UFPA.

6. Características Inovadoras

O projeto destaca-se por se pretender dispor um *kit* de ferramentas de software *free* e *open source* de apoio à implementação do programa de melhoria da qualidade dos processos organizacionais proposto pelo modelo MPS.BR, o que tendência uma possível flexibilização dos custos para a organização e demanda uma redução do tempo considerável para empresas com foco na prestação de serviços de consultoria neste contexto. Além disso, incentiva, cada vez mais, a adoção de procedimentos ágeis na comunidade, minimizada pela baixa burocracia que tais ferramentas poderiam agregar quando da institucionalização do processo de desenvolvimento.

7. Conclusões e Perspectivas Futuras

Este artigo apresentou as características de um projeto de ferramental sistematizado de software livre para apoio à implementação das boas práticas de desenvolvimento de software constantes no modelo MPS.BR, em desenvolvimento pelo grupo de pesquisa SPIDER, institucionalizado no Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará. Este projeto visa apoiar as organizações de desenvolvimento de software a alcançar níveis de maturidade dos seus processos com base em práticas consolidadas por modelos de

qualidade, adotando como premissa a possibilidade de customização de ferramentas de software e o baixo custo advindos do uso de *free software* e *open source*. A avaliação de seus resultados encontra-se em execução ao longo do processo de desenvolvimento do projeto a partir da implantação destas ferramentas em um contexto de uma fábrica de software na região Norte e posteriormente uma avaliação simulada do modelo MPS.BR, para caracterizar a efetividade do projeto.

A empresa foco da avaliação do projeto foi fundada em 1974 e foi a primeira empresa genuinamente nacional a produzir e comercializar tecnologia no segmento de informática. Nos anos 90, já fazendo parte de estrutura de um banco público nacional, tornou-se uma integradora de solução. Em 2004 esta empresa iniciou o desenvolvimento de um programa de excelência tecnológica com um cliente bancário, situado na região Norte e no ano de 2007, diante da necessidade deste cliente em atualizar tecnologicamente (mudança de plataforma operacional) um sistema de Fomento e em função da sua complexidade, o projeto ganhou uma equipe dedicada para seu desenvolvimento. Com isso, a empresa criou seu projeto piloto de uma fábrica de software no Pará. A fábrica possui aproximadamente 98 (noventa e oito) recursos humanos desempenhando papéis bem definidos e seguindo um processo de software, com vistas à implementação e avaliação do modelo MPS.BR.

Atualmente, o projeto está em fase de consolidação dos resultados já obtidos com as ferramentas customizadas e desenvolvidas para apoio ao programa de melhoria do processo organizacional, e novas pesquisas já se encontram em desenvolvimento com foco na Definição, Avaliação e Melhoria do Processo Organizacional, Gerência de Recursos Humanos, Gerência de Reutilização, Desenvolvimento de Requisitos, Projeto e Construção do Produto, Verificação e Validação, para o atendimento dos níveis de maturidade E e D do MPS.BR.

Como trabalhos futuros esperam-se ter: uma adaptação, se necessária, dos serviços (operações) das ferramentas para prover o total atendimento ao MPS.BR; uma integração funcional destas ferramentas a fim de caracterizar uma relação operacional dos resultados esperados constantes no MPS.BR, o que caracteriza o *suite*; e por fim a implantação deste *suite* em um contexto organizacional de desenvolvimento de software, evidenciando a implementação do modelo de qualidade.

8. Agradecimentos

O desenvolvimento deste trabalho está sendo apoiado financeiramente pelo CNPq, CAPES, FADESP-PA e dos programas PARD/UFPa e PIBIC/UFPa/CNPq.

Referências Bibliográficas

IEEE Computer Society (2004) "SWEBOOK – Guide to the Software Engineering Body of Knowledge".

the International Organization for Standardization/ the International Electrotechnical

Commission (2008) "ISO/IEC 12207 Systems and software engineering– Software life cycle processes". Geneve.

GNU Operating System (2010) "Free Software Foundation". Disponível em: www.gnu.org. Último acesso em outubro/2010.

Governo Federal Brasileiro (2010) "Portal Software Livre". Disponível em: www.softwarelivre.gov.br. Último acesso em outubro/2010.

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.

Oliveira, S. R. B. *et al* (2010) "SPIDER: Uma Proposta de uma Solução Sistêmica de Apoio à Implementação do programa MPS.BR". Anais do VIII Encontro Anual de Computação – ENACOMP 2010. Catalão.

Softex - Sociedade para Promoção da Excelência do Software Brasileiro (2009) "MPS.BR - Melhoria de Processo do Software Brasileiro - Guia Geral".

SEI – Software Engineering Institute (2010) "Capability Maturity Model Integration for Development – CMMI-Dev". Versão 1.3.

Uma Investigação sobre os Fatores Críticos de Sucesso em Iniciativas de Melhoria de Processos de Software

Mariano Angel Montoni, Ana Regina Cavalcanti da Rocha

COPPE/UFRJ – Universidade Federal do Rio de Janeiro Caixa Postal 68511 – CEP 21945-970 – Rio de Janeiro – RJ – Brasil

{mmontoni,darocha}@cos.ufrj.br

Abstract. *The organizations face a variety of difficulties in the conduction of software processes improvement initiatives. The causes of such difficulties are related to socio-cultural, technological and organizational aspects. The understanding of the critical success factors in software processes improvement initiatives is essential to support the management of improvement initiatives and implementation best practices. This work presents an investigation conducted by applying both qualitative and quantitative data analysis methods, aiming to construct a theoretical framework that helps to explain the social process of human behavior that governs software processes improvement implementation. The theoretical framework is constituted of concepts and influence relationships, grounded on a set of propositions (hypotheses), representing the vision and the perspective of software processes improvement practitioners.*

Resumo. *As organizações enfrentam diversas dificuldades na condução de iniciativas de melhoria de processos de software. As causas dessas dificuldades estão relacionadas a aspectos de caráter sócio-cultural, tecnológico e organizacional. A compreensão dos fatores críticos de sucesso em iniciativas de melhoria de processos de software é fundamental para apoiar a gerência de iniciativas de melhoria e de melhores práticas de implementação. Este trabalho apresenta uma investigação conduzida por meio da aplicação de métodos qualitativos e quantitativos de análise de dados, visando construir um framework teórico que ajude a explicar o processo social do comportamento humano que rege a implementação de melhorias em processos de software. O framework teórico é constituído de conceitos e relacionamentos de influência, fundamentados em um conjunto de proposições (hipóteses), representando a visão e a perspectiva de implementadores de melhorias em processos de software.*

1. Introdução

Para apoiar a implementação de melhorias em processos de software, diversos programas têm sido conduzidos com o propósito de desenvolver e aprimorar *frameworks* de melhores práticas de desenvolvimento de software, por exemplo, o CMMI (Sei, 2006) e o MPS.BR (Softex, 2009). No entanto, apesar de ter crescido de forma geral, nos últimos anos, a adoção de normas e modelos de referência para melhoria de processos, a quantidade de organizações que adotam esses modelos é uma parcela reduzida da população total de organizações de software (Staples *et al.*, 2007).

Estudos conduzidos para entender as razões do porque as organizações não adotam as normas e os modelos para melhoria de processos apontam para questões associadas ao alto custo e à burocracia relacionada à grande quantidade de recursos demandados pela execução dos processos (Staples *et al.*, 2007; Coleman e O'connor, 2008).

Considerando, então, que a implementação de melhorias em processos é um fenômeno sócio-cultural, para obter um entendimento preciso da prática de melhoria de processos é fundamental ir além do estudo de fatores tecnológicos (Dyba, 2000; El- Emam *et al.*, 2001; Coleman e O'connor, 2007). No entanto, o estudo de questões não tecnológicas, como o comportamento social humano, tem recebido pouca atenção pelos pesquisadores da área (Bertelsen, 1997; Seaman, 1999; Coleman e O'connor, 2007; Adolph *et al.*, 2008).

Este trabalho apresenta uma investigação conduzida por meio da aplicação de métodos qualitativos e quantitativos de análise de dados, visando construir um *framework* teórico que ajude a explicar o processo social do comportamento humano que rege a implementação de melhorias em processos de software. O *framework* teórico é constituído de conceitos e relacionamentos de influência, fundamentados em um conjunto de proposições (hipóteses), representando a visão e a perspectiva de implementadores de melhorias em processos de software.

A seção 2 apresenta os objetivos e a justificativa do trabalho. A seção 3 apresenta a metodologia e a estrutura geral da pesquisa. A seção 4 apresenta os principais resultados obtidos na investigação conduzida no contexto deste trabalho. A seção 5 descreve os principais aspectos de aplicabilidade do trabalho. A seção 6 apresenta as características inovadoras do trabalho. Finalmente, a seção 7 apresenta a conclusão e os próximos passos da investigação.

2. Objetivos e Justificativa

O objetivo geral deste trabalho é conduzir uma investigação sobre os fatores críticos de sucesso em iniciativas de melhoria de processos de software, visando construir um *framework* teórico que ajuda a explicar o processo

social do comportamento humano que rege a implementação de melhorias no contexto do setor de software do Brasil.

Devido ao fato dos relatos da literatura apontarem falhas comuns na metodologia aplicada em estudos qualitativos nas áreas de Engenharia de Software e melhoria de processos (ADOLPH et al., 2008), a investigação, conduzida neste trabalho, buscou adotar uma metodologia de pesquisa que garanta o rigor e formalismo na realização de estudos sobre as questões críticas do processo social do comportamento humano, segundo os conceitos e princípios dos métodos tradicionais de coleta e análise de dados para investigações dessa natureza.

Considerando a importância do Programa MPS.BR para o aumento da competitividade das organizações de software brasileiras, bem como a existência de uma rica fonte de dados de experiências de implementadores do modelo MPS na coordenação de iniciativas de melhoria de processos em diversas regiões do Brasil, buscou-se, também, considerar a visão e a perspectiva desses implementadores na construção, neste trabalho, do *framework* teórico sobre os fatores críticos de sucesso em iniciativas de melhoria.

3. Metodologia de Execução

Considerando as principais dificuldades e os problemas na adoção de modelos e normas para melhoria de processos, bem como as limitações dos estudos conduzidos na área sobre fatores críticos de sucesso em iniciativas de melhoria, foi identificada a necessidade de construir uma Teoria de Implementação de Melhorias de Processos de Software que pudesse contribuir para explicar as questões críticas relacionadas ao processo social do comportamento humano e que podem influenciar o sucesso de iniciativas de melhoria.

Sendo a Teoria de Implementação de Melhorias de Processos de Software uma teoria substantiva sobre um fenômeno social da área de melhoria de processos, métodos qualitativos são considerados adequados para guiar a sua construção (Seaman, 1999; Coleman e O'connor, 2007). Decidiu-se utilizar o método *Grounded Theory*, segundo a linha straussiana, para guiar a condução da investigação, neste trabalho, pois esse método estabelece um conjunto de procedimentos para condução de pesquisa qualitativa em áreas pouco exploradas, bem como é um método com efetividade comprovada para apoiar a coleta e análise de experiências de profissionais da área de software, visando a construção de teorias substantivas sobre o comportamento humano (Orlikowski, 1993; Coleman e O'connor, 2007; Crabtree et al., 2009).

A metodologia adotada procurou seguir os principais conceitos do método *Grounded Theory*: princípio de emergência, comparação constante, amostragem teórica e procedimentos de codificação. A descrição completa da metodologia é apresentada em (Montoni e Rocha, 2010).

Procurou-se aplicar a metodologia de pesquisa de forma a evitar que idéias preconcebidas sobre o fenômeno estudado, exercessem algum tipo de influência no resultado da investigação. Para tanto, foram definidas as seguintes questões de pesquisa de forma aberta: (i) *QP1 – Quais são os fatores capazes de influenciar o sucesso de iniciativas de melhoria?* e (ii) *QP2 – Como os implementadores de melhoria tratam os fatores capazes de influenciar o sucesso de iniciativas de melhoria?*

Para investigar essas questões, dividiu-se a pesquisa em duas fases. A primeira fase envolveu a realização de um *survey* com consultores e membros de organizações de software para identificar fatores críticos de sucesso, bem como uma revisão sistemática da literatura sobre fatores críticos de sucesso. Segundo o princípio de emergência do método *Grounded Theory*, o resultado da primeira fase da pesquisa foi utilizado para guiar a fase seguinte de forma a focar a investigação apenas nos fatores mais críticos, capazes de influenciar o sucesso de iniciativas de melhoria.

Na segunda fase da pesquisa, foram realizados um *survey* e entrevistas com implementadores de uma organização específica de consultoria em melhoria de processo. Os participantes desses estudos foram selecionados segundo o princípio de amostragem teórica do método *Grounded Theory*, o qual estabelece que as fontes de dados devem ser escolhidas de acordo com a capacidade em explorar os conceitos do problema ou fenômeno de interesse. Portanto, a organização selecionada para esta etapa da investigação foi a COPPE/UFRJ, devido à grande experiência da instituição na coordenação de iniciativas bem sucedidas de implementação de melhorias em processos de software.

Como resultado dos estudos realizados, foi construído um *framework* teórico, constituído de um conjunto de proposições (hipóteses) e um conjunto de categorias (conceitos) que tentam explicar as questões críticas relacionadas ao processo social do comportamento humano e que podem influenciar o sucesso de iniciativas de melhoria.

Na etapa final da investigação, foram conduzidos estudos específicos para avaliar o *framework* teórico construído. Esses estudos foram os seguintes: (i) uma avaliação das proposições (hipóteses) que compõem o *framework* teórico por meio de um *survey* com coordenadores de diferentes instituições implementadoras do modelo MPS; e (ii) uma verificação da existência das categorias do *framework* teórico com base na análise de relatórios produzidos por diferentes instituições implementadoras do modelo MPS, contendo descrições de melhores práticas e lições aprendidas na implementação de melhorias em processos de software. Para reduzir

o viés na avaliação do *framework* teórico, foi realizada, ao final da investigação, uma auditoria da verificação por um especialista em melhoria de processo.

4. Resultados Obtidos

Conforme descrito na seção 3, o trabalho executado foi dividido em duas fases. A primeira fase foi conduzida nos ciclos anteriores do projeto (do período de 2007 a 2009). Os resultados obtidos nesses ciclos foram utilizados como base para a execução do ciclo atual do projeto (período de 2010). Os principais resultados obtidos neste ciclo atual do projeto são apresentados a seguir.

4.1. Outros produtos gerados disponibilizados para o mercado: *Framework* Teórico Sobre os Fatores Críticos de Sucesso em Iniciativas de Melhoria

Nos ciclos anteriores do projeto, foi realizada uma investigação de caráter exploratório, envolvendo a realização de um *survey* com implementadores de melhorias em processos e membros de diferentes organizações de software, e um estudo baseado em revisão sistemática da literatura sobre fatores críticos de sucesso em iniciativas de melhoria.

Técnicas de análise estatística de dados foram aplicadas, nesta primeira fase da pesquisa, visando identificar grupos de fatores com relacionamento estatisticamente significativo. A metodologia aplicada na primeira fase da investigação é detalhada em (Montoni e Rocha, 2007).

O objetivo do ciclo atual do projeto foi investigar como implementadores de melhoria tratam os fatores capazes de influenciar o sucesso de iniciativas de melhoria.

Esta fase do estudo foi conduzida em cinco etapas: (i) Condução de um *survey* com os implementadores da COPPE/UFRJ, (ii) Realização de entrevistas semi-estruturadas com os implementadores experientes da COPPE/UFRJ, (iii) Condução de um *survey* com os coordenadores das instituições implementadoras do modelo MPS, (iv) Verificação da teoria construída nesta fase do estudo e (v) Auditoria da verificação dessa teoria.

Como resultado do estudo, foram identificados dois conjuntos de categorias centrais denominadas de “Contexto Institucional para Implementação de Melhorias em

Processos de Software” e “Comportamento Estratégico na Implementação de Melhorias em Processos de Software”, representando, respectivamente, as características de contexto que podem influenciar o sucesso de uma iniciativa de melhoria, e as ações estratégicas realizadas por atores da implementação de melhorias em processos, capazes de reforçar ou modificar tais influências.

Para facilitar a análise das relações de influência entre os conceitos, foram extraídos grupos de propriedades inter-relacionadas. O conjunto de propriedades de conceitos inter-relacionadas foi denominado de “Componente”. De maneira diferente da primeira fase do estudo, na qual foram aplicadas técnicas de análise estatística de dados para extração de fatores inter-dependentes, nesta segunda fase do estudo, havia mais conhecimento sobre os fatores, possibilitando que a identificação dos componentes fosse realizada de forma qualitativa por meio da análise das proposições da teoria, e que definem os relacionamentos entre as propriedades.

Foram identificados seis componentes de relacionamento entre as propriedades, denominados de: (i) “Aceitação e Comprometimento”, (ii) “Apoio e Relacionamento”, (iii) “Melhorias nos Processos de Software”, (iv) “Capacitação e Gerência”, (v) “Recursos”, e (vi) “Retorno do Programa de Melhoria de Processos”.

Esquemas teóricos e análises também foram elaborados para as cinco categorias de subprocessos que agregam os conceitos que representam o comportamento ou o processo de mudança organizacional, do ponto de vista da instituição de consultoria, e que descrevem a interação entre os atores da iniciativa de melhoria e o contexto institucional no qual as melhorias são implementadas.

Os últimos passos da segunda fase da investigação envolveram a avaliação do *framework* teórico por meio da condução de dois estudos. Primeiro, foi realizada uma avaliação do *framework* teórico por meio de um *survey* com coordenadores de instituições implementadoras do modelo MPS. O objetivo do *survey* foi verificar o quanto a teoria desenvolvida, representa a visão e perspectiva dos implementadores de melhoria de diferentes instituições de consultoria. Como resultado dessa avaliação, foi possível distinguir entre as proposições que representam, de forma geral ou específica, a perspectiva das instituições implementadoras do modelo MPS. Em seguida, foi realizada uma avaliação do *framework* teórico por meio da verificação da existência dos conceitos da teoria (categorias) em um conjunto específico de fontes de dados, contendo descrições sobre melhores práticas e lições aprendidas na implementação de melhorias em processos. Foram utilizados como fontes de dados, todos os relatórios semestrais elaborados pelos coordenadores das instituições implementadoras do modelo MPS, credenciadas pela SOFTEX. O resultado dessa verificação foi a identificação de passagens de textos, extraídas dos relatórios, que evidenciam a existência dos conceitos da teoria desenvolvida no estudo. Visando reduzir o viés dessa verificação, foi feita uma auditoria da comprovação da existência dos conceitos por um especialista em melhoria de processo.

4.2. Métodos e/ou Algoritmos Desenvolvidos

Na execução do trabalho, foi desenvolvido um método de investigação qualitativo baseado nos princípios do método *Grounded Theory*. Esse método é descrito na seção 3 deste trabalho. Espera-se que o método desenvolvido possa apoiar futuras investigações qualitativas em processos de software e outras áreas correlatas.

4.3. Artigos publicados

Os seguintes artigos foram publicados no ciclo atual do projeto:

- ? MONTONI, M.A., ROCHA, A.R., 2010, "Aplicação de Grounded Theory para Investigar Iniciativas de Implementação de Melhorias em Processos de Software". In: *IX Simpósio Brasileiro de Qualidade de Software (SBQS)*, pp. 167-182, Belém, Junho.
- ? MONTONI, M.A., ROCHA, A.R., 2010, "Applying Grounded Theory to Understand Software Process Improvement Implementation". In: *7th International Conference on the Quality of Information and Communications Technology*, pp. 25-34, Porto, Portugal, 29 de setembro a 02 de outubro.

4.4. Dissertações e/ou teses geradas

O trabalho apresentado no ciclo atual foi conduzido no contexto da tese de doutorado de Mariano Angel Montoni, sob a orientação da Profa. Ana Regina Cavalcanti da Rocha. A tese foi defendida com sucesso em Agosto de 2010. Os dados da publicação da tese são os seguintes:

- ? Montoni, M.A., 2010, Uma Investigação sobre os Fatores Críticos de Sucesso em Iniciativas de Melhoria de Processos de Software, Tese de D.Sc., Programa de Pós-graduação em Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil.

5. Aplicabilidade dos Resultados

Os resultados obtidos com a execução do trabalho são de extrema relevância para o aumento da competitividade das empresas brasileiras de desenvolvimento de software.

Com base no *framework* teórico construído, estratégias mais eficientes de implementação de melhoria de processos podem ser definidas. Além disso, organizações desenvolvedoras de software poderão utilizar os conceitos (categorias) identificados, bem como as proposições que estabelecem relações de influência entre esses conceitos, como um mecanismo para avaliar o potencial de sucesso (e fracasso) das iniciativas de melhoria, antes mesmo de serem iniciadas. Dessa forma, espera-se que a disseminação do conhecimento sobre os fatores críticos de sucesso em iniciativas de melhoria de processos ajude as empresas a aumentarem as chances de sucesso dos seus esforços em prol da melhoria de processos.

Em termos de abrangência, os resultados obtidos se aplicam a qualquer empresa desenvolvedora de software, independente do seu tamanho e área de atuação, como um elemento chave para apoiar a condução de iniciativas de melhoria. Além disso, os resultados obtidos poderão ser utilizados por pesquisadores na área com interesse em aprofundar a investigação em um conjunto específico de fatores críticos de sucesso.

O impacto dos resultados obtidos é bastante expressivo, pois o conhecimento sobre fatores críticos de sucesso resultante do trabalho executado poderá aumentar significativamente o sucesso das iniciativas de melhoria de processos de software conduzidos pelas empresas brasileiras, bem como pode ajudar a aumentar a adoção de modelos de referência de processos como o CMMI e o MPS pelas empresas. Dessa forma, as empresas brasileiras aumentarão suas vantagens competitivas frente a outros competidores do mercado nacional e internacional de software.

6. Características Inovadoras

O projeto possui diversas características inovadoras. A principal delas é a utilização de uma metodologia da área de ciências sociais para o estudo de fatores críticos de sucesso em iniciativas de melhoria de processos de software. Alguns estudos já foram iniciados por outros pesquisadores com base na metodologia empregada no projeto, bem como nos resultados obtidos.

Apesar de existirem na literatura, alguns trabalhos sobre fatores críticos de sucesso em iniciativas de melhoria, nenhum desses trabalhos fornece indícios claros, objetivos e concretos sobre a relação entre os fatores e como tratá-los de forma adequada visando aumentar as chances de sucesso das iniciativas de melhoria. Este trabalho, então, pode ser considerado como o primeiro trabalho da área capaz de descrever de forma abrangente, a influência de contexto nas ações de atores chave da implementação de melhoria.

7. Conclusão e Perspectivas Futuras

Este trabalho apresentou uma investigação conduzida por meio da aplicação de métodos qualitativos e quantitativos de análise de dados, visando construir um *framework* teórico que ajude a explicar o processo

social do comportamento humano que rege a implementação de melhorias em processos de software. A metodologia adotada na investigação foi apresentada, bem como os principais resultados obtidos.

Os resultados desta pesquisa confirmaram que a implementação de melhorias em processos de software é de fato um processo social. Essa constatação pode ser comprovada pela identificação de questões críticas associadas ao relacionamento entre os consultores da instituição implementadora e os membros da organização alvo da implementação das melhorias. Um fator que influencia esse relacionamento é a confiança dos membros da organização na consultoria especializada.

Como principal contribuição deste trabalho, pode-se destacar a construção de um *framework* teórico que ajuda a explicar o processo social do comportamento humano que rege a implementação de melhorias em processos no contexto do setor de software do Brasil. Além disso, também constitui uma contribuição, a aplicabilidade do método *Grounded Theory* para investigar os fatores críticos de sucesso em iniciativas de melhoria de processos de software, bem como o meta-modelo desenvolvido para organizar os conceitos e relacionamentos do *framework* teórico.

Os próximos passos deste trabalho envolvem a realização de novas investigações com o propósito de replicar a pesquisa em diferentes contextos, bem como aprofundamento da avaliação do *framework* teórico.

8. Referências Bibliográficas

Adolph, S., Hall, W., Kruchten, P., 2008, "A methodological leg to stand on: Lessons learned using grounded theory to study software development", pp. IBM Toronto Software Lab.; IBM Centers for Advanced Studies, CAS, Richmond Hill, ON, Canada.

Bertelsen, O.W., 1997, "Toward a unified field of SE research and practice", *IEEE Software*, v. 14, n. 6, pp. 87-88.

Coleman, G., O'connor, R., 2007, "Using grounded theory to understand software process improvement: A study of Irish software product companies", *Information and Software Technology*, v. 49, n. 6, pp. 654-667.

Coleman, G., O'connor, R., 2008, "Investigating software process in practice: A grounded theory perspective", *Journal of Systems and Software*, v. 81, n. 5, pp. 772-784.

Crabtree, C.A., Seaman, C.B., Norcio, A.F., 2009, "Exploring language in software process elicitation: A grounded theory approach", *IEEE Computer Society*.

Dyba, T., 2000, "An Instrument for measuring the key factors of success in software process improvement", *Empirical Software Engineering*, v. 5, n. 4, pp. 357-390.

El-Emam, K., Goldenson, D., Mccurley, J., et al., 2001, "Modelling the likelihood of software process improvement: An exploratory study", *Empirical Software Engineering*, v. 6, n. 3, pp. 207-229.

Montoni, M., Rocha, A.R., 2007, "A Methodology for Identifying Critical Success Factors that Influence Software Process Improvement Initiatives: An Application in the Brazilian Software Industry", *Lecture Notes in Computer Science (LNCS)*, LNCS 4764, EuroSPI - European Systems & Software Process Improvement and Innovation (Setembro), pp. 175-186.

Montoni, M.A., Rocha, A.R., 2010, "Aplicação de Grounded Theory para Investigar Iniciativas de Implementação de Melhorias em Processos de Software". In: *IX Simpósio Brasileiro de Qualidade de Software*, pp. 167-182, Belém, Junho.

Orlikowski, W.J., 1993, "CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development", *Management Information Systems Quarterly*, v. 17, n. 3.

Seaman, C.B., 1999, "Qualitative methods in empirical studies of software engineering", *IEEE Transactions on Software Engineering*, v. 25, n. 4, pp. 557- 572.

Sei, 2006, *CMMI® for Development (CMMI-DEV), V1.2*, CMU/SEI-2006-TR-008, Software Engineering Institute.

Softex, 2009, "MPS.BR – Melhoria de Processo do Software Brasileiro, Guia Geral (v. 2009)". In: http://www.softex.br/mpsbr/_guias/default.asp.

Staples, M., Niazi, M., Jeffery, R., et al., 2007, "An exploratory study of why organizations do not adopt CMMI", *Journal of Systems and Software*, v. 80, n. 6, pp. 883-895.

Teste de Software no Contexto do Software Público Brasileiro

Adalberto Nobiato Crespo, Paulo Bueno, Miguel Argollo, Celso Barros, Mario Jino*

Divisão de Melhoria de Processos de Software – DMPS
Centro de Tecnologia da Informação Renato Archer – CTI

{adalberto.crespo, paulo.bueno, miguel.argollo, celso.barros}@cti.gov.br

Faculdade de Engenharia Elétrica e de Computação – FEEC
Universidade Estadual de Campinas – Unicamp

jino@dca.fee.unicamp.br

Abstract. A Estrutura Tecnológica de Teste de Software (ETTS), desenvolvida no contexto do Software Público Brasileiro, disponibiliza em um portal Web recursos elaborados com base em modelos de teste de software consolidados, como TMMi e IEEE-829, e em boas práticas de teste de software. A ETTS está organizada na forma de documentos e guias que abordam aspectos importantes para a realização do teste: planejamento, projeto, execução e registro, acompanhamento, finalização, ferramentas de apoio, técnicas, tipos, níveis e ciclos de teste. A estrutura visa a apoiar os diversos atores do SPB em diferentes situações que envolvam atividades de teste. Este artigo contém: a descrição da estrutura tecnológica; situações e formas de uso desta estrutura; e uma breve conclusão indicando o estágio atual do projeto e possíveis desdobramentos. A contribuição desta iniciativa consiste na criação e disponibilização de um conteúdo público de teste de software cuidadosamente estruturado e passível de evolução contínua.

Keywords: Teste de Software, Processo de Teste, Software Público Brasileiro, Estrutura Tecnológica de Teste de Software.

1. Introdução

As Tecnologias da Informação e Comunicação (TIC) têm hoje um papel fundamental, afetando de forma significativa vários segmentos na sociedade. Como consequência, essas tecnologias assumem a cada dia um papel mais importante na produção e distribuição de riqueza, fato que justifica a adoção de políticas públicas para estimular o desenvolvimento da indústria de TIC no Brasil. Neste contexto a iniciativa do Software Público Brasileiro (SPB) estabelece uma nova etapa na geração, gestão e disponibilização de soluções tecnológicas.

O Software Público Brasileiro, uma iniciativa da Secretaria de Logística e Tecnologia da Informação do Ministério do Planejamento, Orçamento e Gestão (SLTI/MPOG), consiste essencialmente em uma série de soluções de tecnologia da informação e comunicação disponibilizadas para uso público por meio de um portal Web [9]. Diversos recursos de software encontram-se disponíveis relacionados a comunidades de usuários que atuam de forma colaborativa para apoiar o uso e aprimorar as soluções.

Dentre as áreas contempladas pelas soluções do SPB encontram-se: saúde; educação; saneamento; gestão de TI, administração pública, comunicação e geoprocessamento.

O SPB estabelece o desenvolvimento cooperativo e o compartilhamento de soluções de software, associados a mecanismos de suporte legal e de negócios. Além de contribuir para a disseminação do uso dessas tecnologias, o SPB permite tratar as soluções de TIC no contexto da política pública, considerando o papel estratégico dessas soluções na sociedade. Esta iniciativa vem sendo reconhecida em diversas instâncias, com o destaque por parte do Open World Fórum [2] como “melhor formulador de políticas públicas na área de software livre e aberto”.

Com a finalidade de promover confiança nos recursos disponíveis no SPB, encontra-se em andamento um projeto para definir um “modelo de referência para o Software Público Brasileiro” (MR-SPB) [9]. O projeto está sendo desenvolvido no Centro de Tecnologia da Informação Renato Archer (CTI/MCT) e visa à criação, em conjunto com as comunidades de software do portal, de um modelo abrangente de qualidade de software, que estabeleça características, propriedades e práticas desejadas no SPB. Um dos elementos deste modelo de qualidade é o Teste de Software.

Atividades de teste são essenciais para a avaliação de vários tipos de produtos (aeronaves, circuitos elétricos, etc.), elaborados segundo diferentes disciplinas de engenharia (aeronáutica, mecânica, elétrica, etc). O Teste de Software visa a avaliar a qualidade de um produto de software por meio da execução sistemática e cuidadosa deste produto. O teste de software, quando bem realizado, permite obter evidências da qualidade do software, além de identificar problemas existentes antes que eles causem danos aos usuários.

2. Objetivos e Justificativa

O desenvolvimento de uma Estrutura Tecnológica de Teste de Software – ETTS tem como objetivo apoiar os diferentes agentes participantes do SPB em atividades relacionadas ao teste de software. Esta Estrutura

Tecnológica de Teste de Software disponibiliza por meio de um portal Web recursos elaborados com base em boas práticas de teste de software [9]. Esses recursos abordam aspectos importantes para a realização do teste e buscam viabilizar atividades de teste eficazes no SPB, levando em conta a heterogeneidade das organizações participantes, a forma de trabalho colaborativo e o compartilhamento de informações, características importantes deste ambiente.

Qualquer instituição que deseje disponibilizar um software no Portal do Software Público Brasileiro - SPB deve se responsabilizar pela qualidade do seu produto. O teste de software, que é uma das atividades de verificação e validação, pode ser utilizado como uma iniciativa para obtenção da qualidade de um produto de software.

O teste consiste em executar o software de uma forma controlada com o objetivo de avaliar se o software se comporta conforme especificado. É uma atividade fundamental para: avaliar se o software atende aos requisitos levantados; identificar deficiências que podem existir no software e; ainda, obter evidências da confiabilidade do software. Portanto, para a obtenção de produtos de software de qualidade é de suma importância a adoção de boas práticas de teste de software.

3. Metodologia de Execução

A ETTS foi estabelecida considerando-se os requisitos das empresas produtoras de software.

Conteúdos de teste foram organizados em documentos na forma de modelos e guias. Esses documentos abordam aspectos do teste, como: planejamento, projeto, execução e registro, acompanhamento e finalização. São tratadas também questões dimensionais do teste, como: técnicas de teste, tipos de teste, níveis de teste e critérios de teste. A Figura 1 ilustra as três dimensões do teste de software.

Os documentos elaborados priorizam a prática do teste e têm como características gerais o uso de linguagem clara, a descrição de conceitos de forma simples, e exemplos ilustrativos.

A criação da ETTS está fundamentada não só na parte teórica sobre teste de software, mas também nas experiências práticas de sucesso vivenciadas por empresas nacionais e estrangeiras.

O desenvolvimento da ETTS está fundamentado nos seguintes pontos:

- a) Prospecção das tendências tecnológicas mundiais e das necessidades das empresas nacionais;
- b) Seleção e domínio de soluções tecnológicas em teste de software; e
- c) Repasse das tecnologias dominadas e das informações obtidas para as empresas produtoras de software.

Com um caráter genérico e flexível, essa estrutura tecnológica de teste deve servir para as diferentes organizações desenvolvedoras de software do SPB melhorarem suas práticas de teste. Como consequência almeja-se um resultado mais abrangente: permitir o aprimoramento da qualidade do software brasileiro.

As peculiaridades do ambiente do SPB estabelecem requisitos da ETTS, a saber: R1) deve permitir que diversos agentes realizem esforços colaborativos de teste, por meio de tecnologias de comunicação, trabalhando em redes virtuais; R2) deve oferecer uma base que permita o aprimoramento das práticas de teste de forma autônoma por organizações que disponham de recursos reduzidos; e finalmente, R3) deve contribuir para a expansão progressiva do conhecimento público em teste de software, realizada colaborativamente e de forma descentralizada.

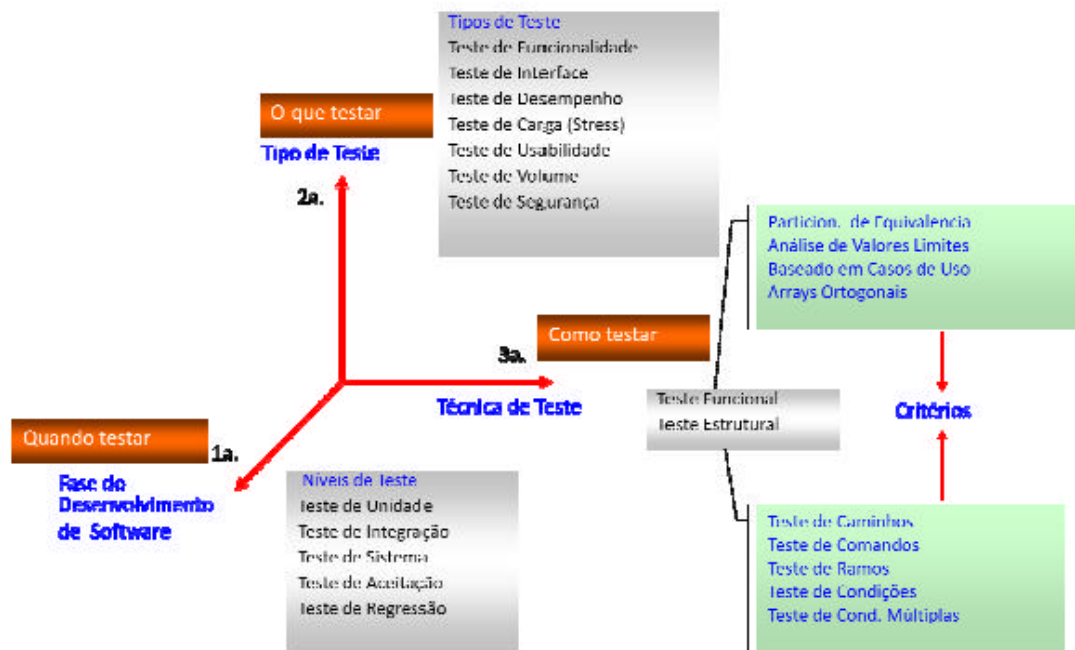


Figura 1: As três Dimensões do Teste de Software

Com o objetivo de satisfazer a esses requisitos foi proposto um modelo conceitual apoiado nos três pilares para a estruturação e melhoria de organizações: Pessoas, Processos e Tecnologia [3]. Mais especificamente, esta estrutura propõe-se a: apoiar a capacitação e treinamento de profissionais de software nas melhores práticas de teste; fornecer subsídios para a definição de processos de teste adequados às necessidades de teste específicas; e identificar métodos, técnicas e ferramentas de apoio às atividades de teste de software.

4. Resultados Obtidos

A seguir são descritos brevemente. Observe-se que todos os guias contemplam: as diretrizes para a realização das atividades definidas, os participantes dessas atividades, os requisitos de início e término das atividades, as informações entrada, as informações geradas e seus beneficiários.

Teste de Software: Motivação e Conceitos

É um documento que apresenta de teste; os procedimentos; o processo; os tipos;

Modelo de Processo Genérico de Teste

Um modelo de processo genérico de teste de software é um modelo de processo abrangente que envolve as situações de teste de um software e contém as atividades de teste correlacionadas. São proposto possui os seguintes subprocessos: Planejamento do Teste; Projeto do Teste; Execução e Registro do Teste; Acompanhamento do Teste; Finalização do Teste. O documento que descreve o processo genérico apresenta diferentes cenários de sua utilização, ilustrando, por exemplo, como o processo de teste pode ser customizado para diferentes modelos de desenvolvimento (cascata, incremental, etc).

Guia para Planejar o Teste

O planejamento do teste de software é o subprocesso que antecede o projeto e a execução do teste. Tem como propósito estabelecer, em linhas gerais, quais e como serão conduzidas as atividades do teste de um software.

As atividades do planejamento do teste são: estabelecer o objetivo do teste; determinar os riscos associados ao software; estabelecer o escopo do teste; definir a abordagem do teste; estabelecer as tarefas de teste; estabelecer artefatos de teste; definir o ambiente do teste; definir a equipe de teste; construir o cronograma de tarefas; identificar responsabilidades; definir recursos adicionais para o teste; selecionar métricas; e analisar os riscos do teste.

Guia para Projetar o Teste

O projeto de teste é o subprocesso que antecede a execução do teste. Tem como propósito principal gerar casos de teste a partir dos requisitos e características do software, e da abordagem e objetivo do teste, definidos durante o planejamento do teste.

As atividades do projeto do teste são: refinar a abordagem do teste; identificar e priorizar cenários de teste; projetar e priorizar casos de teste; documentar e organizar casos de teste; elaborar procedimentos de teste; definir base de dados de teste; refinar critérios de suspensão e retomada do teste; criar associações entre casos de teste e artefatos; refinar ambiente de teste; refinar cronograma de teste.

Guia para Executar e Registrar o Teste

A execução e registro dos testes é o subprocesso que tem como propósito a aplicação dos testes planejados e projetados anteriormente, assim como a documentação dos eventos e resultados associados à execução dos testes.

As atividades de execução e registro do teste são: preparar o ambiente de teste; preparar a base de dados de teste; realizar o teste preliminar do ambiente de teste e registrar resultados; realizar o teste preliminar do componente a ser testado e registrar resultados; executar os casos de teste de acordo com os procedimentos definidos; registrar os resultados obtidos; comparar os resultados obtidos com os esperados; analisar, registrar e relatar incidentes de teste; coletar e documentar as informações sobre a execução do teste.

Os resultados da execução do teste são consolidados em relatórios de execução do teste e de incidentes de teste.

Guia para Acompanhar o Teste

O acompanhamento do teste é o subprocesso que tem como propósito assegurar que o teste seja executado conforme o Plano de Teste e o Projeto de Teste anteriormente elaborados.

As atividades de acompanhamento são: a) monitorar o progresso do teste, o que inclui: confrontar os resultados e os recursos utilizados em relação aos valores estimados; monitorar os compromissos acordados, os riscos do projeto de teste e o envolvimento dos “stakeholders”; revisar o progresso do teste e os marcos do progresso do teste. b) monitorar a qualidade do produto, o que inclui: verificar os critérios de entrada para o teste; monitorar defeitos, riscos do produto, critérios de saída para o teste, critérios de suspensão e retomada do teste; revisar a qualidade do produto e os marcos da qualidade do produto. c) gerenciar ações corretivas até a sua finalização, o que inclui: analisar os problemas; tomar ação corretiva; gerenciar a ação corretiva.

Guia para Finalizar o Teste

A finalização dos testes é o subprocesso que tem como propósito o encerramento do teste e inclui a aceitação formal do produto pelo cliente.

As atividades de finalização do teste são: avaliar as medidas coletadas; avaliar o desempenho da equipe; avaliar as estimativas iniciais; avaliar o processo de teste; identificar oportunidades de melhoria; comunicar formalmente o término do teste; e relatar e publicar as informações do teste.

Artigo Publicado

No desenvolvimento da ETTS, o seguinte artigo foi publicado:

Teste Colaborativo no Software Público Brasileiro - Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento (JIISIC), 2010 – Mérida, México

5. Aplicabilidade dos Resultados

A disponibilização da ETTS no 5CQualiBr permite que diferentes usuários possam realizar ou aprimorar suas atividades de teste. Usuários também podem contribuir para o aperfeiçoamento da ETTS

postando conteúdos em fóruns, blogs, ou disponibilizando, em uma área apropriada, artigos, guias, *templates*, ou outros recursos de teste. Esta interação bidirecional deve levar ao aprimoramento da estrutura e é coerente com a visão das comunidades do SPB formadas por “*prosumers*” (*producersconsumers*), ou seja, os envolvidos na construção e melhoria da estrutura são também os usuários e os beneficiados pela estrutura [11].

A ETTS apóia diferentes tipos de usuários na realização de atividades relacionadas a teste de software. Podem ser caracterizadas quatro principais *situações de uso* da estrutura:

a) Ao longo do desenvolvimento do software: desenvolvedores de software têm interesse em implantar ou aprimorar um processo de teste de software. O processo de teste é alinhado ao processo de desenvolvimento e o software desenvolvido pode ou não ser disponibilizado no Portal do SPB.

b) Na homologação inicial de um software para o Portal do SPB: usuários desejam disponibilizar no Portal do SPB um produto de software já desenvolvido. Podem ser realizadas atividades de homologação do software e geradas evidências dessa homologação.

c) Na disponibilização de novas versões de software no Portal do SPB: usuários desejam homologar algum software do Portal do SPB que tenha passado por modificações, caracterizando uma nova versão do software.

d) Na avaliação de evidências de teste: usuários desejam analisar sistematicamente e de forma criteriosa os artefatos produzidos em um esforço de teste para julgar a qualidade do teste realizado e estimar a qualidade do produto testado.

Independentemente da situação (a, b, c, ou d) podem ser identificadas duas *formas de uso*, que classificam ortogonalmente diferentes maneiras pelas quais as situações de uso podem ser efetivadas no SPB:

i. Apoiar atividades de teste realizadas individualmente por uma organização, focadas nos seus

interesses específicos;

ii. Apoiar a realização colaborativa de atividades de teste, focadas em interesses coletivos e envolvendo diferentes atores e organizações.

A primeira forma de uso dá-se em uma organização na qual usuários, desenvolvedores e testadores conduzem atividades de teste de software. A segunda forma de uso caracteriza o Teste Colaborativo que envolve tipicamente dois ou mais atores de diferentes organizações que cooperam para testar um software.

6. Características Inovadoras

A ETTS tem um caráter de uso nacional dado a sua disponibilização no Portal do Software Público Brasileiro onde se constata a existência de usuários de natureza particular, municipal, estadual e federal.

Além disso, a ETTS é um conjunto de recursos de teste de software que está disponível a vários usuários.

Esses recursos concentram os conhecimentos já existentes sobre teste de software, ou seja, não existem novas técnicas, novos tipos de teste ou novos critérios de teste de software. A inovação se caracteriza pela forma de disponibilizar os conhecimentos já existentes sobre teste de software. Além disso, os resultados estão documentados numa linguagem clara e de fácil compreensão por parte dos usuários, para a descrição dos conceitos, das técnicas e procedimentos de teste.

7. Conclusão e Perspectivas Futuras

Este artigo apresenta a Estrutura Tecnológica de Teste de Software desenvolvida no contexto do Software Público Brasileiro. Baseada em conceitos de teste de software consolidados em modelos como TMMi, IEEE-829 e em boas práticas, essa estrutura visa a apoiar os diversos atores do SPB em diferentes situações que envolvam atividades de teste. Trata-se de uma iniciativa que deve ser estendida e aprimorada com as contribuições e experiências dos usuários.

Os conceitos e técnicas utilizados na ETTS não são inovadores; pelo contrário, são amplamente conhecidos e têm sua eficácia comprovada. Os documentos elaborados priorizam aspectos práticos do teste e têm como características gerais o uso de linguagem clara, a descrição de conceitos de forma simples, e exemplos ilustrativos. Desta forma, a estrutura tecnológica procura diminuir a distância entre a teoria e a prática do teste de software. A contribuição desta iniciativa consiste na criação e disponibilização de um conteúdo público de teste de software cuidadosamente estruturado e passível de aprimoramento contínuo.

A promoção da colaboração ainda não está contemplada na ETTS; recursos direcionados especificamente à comunicação, coordenação e cooperação no teste precisam ser definidos. Não obstante, a organização da ETTS é compatível com o teste colaborativo. Aspectos relacionados à colaboração na ETTS são abordados em um modelo em desenvolvimento destinado a fornecer diretrizes para o teste colaborativo de software no SPB.

8. Referências

1. Crespo A.N., Jino, M., Argollo, M., Bueno, P.M.S., "Teste de Software no Desenvolvimento Colaborativo", Revista InfoBrasil, Ed. Especial Software Público Brasileiro, No 7., Junho/Agosto, 2009.
2. FLOSS in an Open World: Innovations and best practices from Brazil. In 2020 FLOSS Roadmap 2009 Version, Open World Forum, December 2009. (<http://www.2020flossroadmap.org/>)
3. Humphrey, Watts S., Managing the Software Process, Addison-Wesley Professional, 1989
4. IEEE Std 829 (1998), "IEEE Standard for Software Test Documentation", IEEE, New York.
5. ISO/IEC 15504 (2004), "The International Organization for Standardization and the International Electrotechnical Commission", ISO/IEC 15504.
6. Koomen, T. and Pol M., (1999), "Test Process Improvement: A practical step-by step guide to structured testing". ACM Press, London, England.
7. Messerschmitt, D.G., Szyperski, C., 2003. Software Ecosystem: Understanding an Indispensable Technology and Industry. MIT press.
8. SEI (2006), CMMI for Development, Version 1.2, Carnegie Mellon University, Software Engineering Institute, CMU/SEI 2006-TR-008 ESCTR- 2006-008, August, 2006.

9. SPB 2009, "Software Público Brasileiro", Revista InfoBrasil, Ed. Especial Software Público Brasileiro, No 7., Junho/Agosto, 2009.
10. SWEBOK Guide to the Software Engineering Body of Knowledge, IEEE Computer Society 2004.
11. Tapscott, D. & Williams A. Wikinomics (2007): How Mass Collaboration Changes Everything. Portfolio Publisher.
12. TMMi - Test Maturity Model Integration, Version 2.0. TMMi Foundation (2009), (<http://www.tmmifoundation.org/downloads/>).
13. Weber et al. (2005) Kival C. Weber, Eratóstenes E. R. Araújo, Ana Regina C. da Rocha, Cristina A. F. Machado, Danilo Scalet and Clênio F. Salviano, Brazilian Software Process Reference Model and Assessment Method, in Proc. of The 20th Int. Symp. on Computer and Info. Sciences - ISCIS'05, October 26-28th, Instambul, Turkey, 2005.

Definição de um Guia de Apoio às Decisões em Definições de Arquiteturas de Software

Júlio César de Paula Silva¹, André Luiz Peron Martins Lanna²

¹Faculdade Cotemig - Belo Horizonte - MG

²CEFET-MG - Campus VIII - Varginha - MG

julio cps.br@gmail.com, andreperon@varginha.cefetmg.br

Abstract. The software architecture has an important role at software quality. Using architectural evaluation techniques allow evaluate if an architecture attends the requirements defined for a software. This paper shows that evaluation techniques haven't been used in large scale on software companies, beyond identifies the most important and recurrent software requirements. From this analysis, architectural scenarios were proposed for attending these common requirements, what allows companies to develop their software from scenarios previously defined and evaluated.

Resumo. A arquitetura de um software desempenha um importante papel na qualidade do software. O uso de técnicas de avaliações arquiteturais permite avaliar se uma arquitetura atende ou não aos requisitos do software. Este artigo mostra que as técnicas de avaliação não têm sido amplamente utilizadas em empresas de desenvolvimento além de identificar os requisitos de software mais importantes e recorrentes. A partir desta análise estão sendo propostos cenários arquiteturais que atendam a tais requisitos para que as empresas possam desenvolver software a partir de cenários já definidos e avaliados.

1. Introdução

Qualidade de software é um tema que ultimamente tem recebido bastante atenção, devido aos impactos positivos que dela podem decorrer. Organizações (privadas, governamentais, de pesquisa e de ensino) têm apresentado diversas propostas para tratar deste assunto, considerando-o sob diferentes aspectos [Lanna and Pietrobon 2010]. Dentre as abordagens existentes destacam-se o reuso de software (artefatos, ideias, boas práticas, dentre outros) e arquiteturas de software.

A arquitetura de um software pode ser vista como a infraestrutura sobre a qual ele será construído através da integração (reuso) de componentes que determinarão suas funcionalidades. Apesar de uma boa arquitetura não garantir que os requisitos de um software serão atendidos, uma arquitetura ruim dificilmente permitirá que tais requisitos sejam satisfeitos [Hofmeister et al. 1999]. Além disso, é na arquitetura de um software que está concentrada grande parte dos requisitos, sobretudo aqueles relacionados a atributos de qualidade tais como confiabilidade, modificabilidade, reusabilidade, dentre outros.

Estes requisitos são fortemente acoplados de modo que uma alteração em algum requisito pode impactar no atendimento aos demais requisitos [Sommerville 2006]. Assim a definição da arquitetura de um software é o resultado da priorização de seus requisitos e de uma série de decisões tomadas para melhor atendê-los. Diversos métodos têm sido propostos para avaliar arquiteturas tais como Quality Attribute Workshop (QAW) e Architecture Tradeoff Analysis Method (ATAM), apresentados em [Clements et al. 2002].

Entretanto a utilização destes métodos de avaliação demanda tempo não apenas do responsável pela arquitetura do software¹⁵ mas de grande parte dos membros envolvidos no projeto do software. Nem sempre as organizações estão dispostas a interromper suas atividades de desenvolvimento para realizar tais análises, principalmente quando os projetos apresentam cronogramas apertados. Como consequências deste fato tem-se a diminuição da qualidade do software e futuramente a dificuldade em desenvolver e realizar manutenções. Isto exposto pode-se afirmar que a qualidade da arquitetura deve ser levada em consideração e quaisquer iniciativas que busquem aumentá-la são bem-vindas.

O presente artigo apresenta o relatório final do projeto "Definição de um guia de apoio às decisões em definições de arquiteturas de software", submetido ao Programa Brasileira de Qualidade e Produtividade em Software (PBQP-SW) no ciclo 2010. Ele está organizado como segue: a seção 2 apresenta os objetivos do projeto e sua justificativa, a seção 3 apresenta a metodologia de execução do projeto, na seção 4 estão apresentados os resultados obtidos até o momento e por fim a seção 5 encerra o artigo com as conclusões e perspectivas futuras.

2. Objetivos e justificativa

¹⁵ 1 Nem todas as organizações dispõem de um arquiteto de software, muitas vezes elegendo uma pessoa com maior conhecimento técnico para executar esta função. Portanto este o motivo para não definir o arquiteto de software como o único responsável por definir a arquitetura.

A definição de uma arquitetura de software não é uma tarefa simples pois devem ser considerados diversos fatores cujo grau de complexibilidade envolve software, hardware e pessoas. Uma arquitetura deve ser pensada de modo estratégico para o ciclo de vida do software compreendendo as fases de planejamento, modelagem, construção e teste.

Pode-se afirmar que um dos grandes problemas é o fato da arquitetura ser definida durante a construção. Neste caso é possível encontrar (geralmente em fóruns de discussão) definições de pilhas arquiteturais sendo consideradas como arquiteturas de software, o que a princípio são objetos distintos. Daí então que desenvolver uma arquitetura de software coesa e com a adoção de boas práticas (tais como padrões) possibilita um software de qualidade mais alta e evita problemas como retrabalho.

Como visto na seção anterior apesar de avaliação arquitetural ser considerada importante, sua utilização parece não ser uma prática bem estabelecida entre as organizações. Para confirmar este cenário uma entrevista foi realizada pelos autores em organizações públicas e privadas de diversos tamanhos. Pelos resultados foi identificado que apenas uma dentre as vinte e uma empresas consultadas aplicava métodos de avaliação arquitetural como ferramenta de apoio às decisões dos arquitetos. Nas demais empresas a definição da arquitetura era realizada baseada no expertise dos arquitetos ou como resultado de discussões da equipe de desenvolvimento quando não há a figura de um arquiteto na organização. Este cenário justifica a criação de algum mecanismo de auxílio à definição de arquiteturas nas organizações como uma alternativa para aumentar a qualidade dos softwares desenvolvidos por elas.

Portanto o objetivo principal deste projeto é prover algum tipo de auxílio aos desenvolvedores das organizações nas tarefas de definição e avaliação de arquiteturas de software. Com base nos benefícios do reuso de software (tais como diminuição de retrabalho e aumento da qualidade pela utilização de elementos previamente definidos), foi definido que este auxílio deverá ocorrer pela reutilização de esboços arquiteturais previamente definidos, testados e avaliados, disponibilizados à comunidade sob a forma de um guia de boas práticas. Como forma de fomentar a utilização deste guia, as avaliações dos esboços arquiteturais são realizadas com base nos reconhecidos métodos QAW (para definir os atributos de qualidade), no ATAM (para a avaliação e priorização dos requisitos) e na norma ISO-9126 (responsável por definir os atributos de qualidade de software).

Para a criação do guia alguns objetivos secundários foram delineados. O primeiro deles é definir um método (através de um conjunto de atividades) que considere desde os atributos de qualidade que os esboços arquiteturais devem apresentar quando finalizados, passando pela definição do próprio esboço arquitetural e seguida pela sua avaliação. O segundo objetivo é avaliar a aplicabilidade do método em uma plataforma específica, cuja a escolhida foi J2EE e suas API's. Por fim o terceiro objetivo traçado é analisar a receptividade do método proposto em organizações de diferentes tamanhos da região de Belo Horizonte.

3. Metodologia de execução

O modo de execução deste projeto pode ser dividido em 4 etapas bem distintas. A primeira diz respeito à um estudo sobre métodos de avaliação arquitetural e sua utilização na região de Belo Horizonte. Posteriormente foi estabelecido um conjunto de atividades para a definição e avaliação de esboços arquiteturais, incluindo decisões sobre quais métodos de avaliação arquiteturais seriam considerados. A terceira etapa corresponde à execução das atividades previamente citadas com base em atributos de qualidade freqüentemente considerados pelas organizações estudadas. Por fim, a quarta etapa é responsável por compilar os esboços arquiteturais em um guia e analisar os resultados obtidos. Cada uma destas fases será descrita a seguir.

O projeto teve início quando um dos autores percebeu a necessidade de documentar boas práticas em Arquiteturas de Software e disponibilizá-las aos desenvolvedores em geral. Naquele momento a questão a ser respondida era: como afirmar se uma arquitetura é boa ou não? Iniciou-se assim um estudo sob métodos e ferramentas que permitissem analisar e avaliar criticamente uma arquitetura de software. Dentre os métodos estudados destacam-se dois desenvolvidos pelo Software Engineering Institute (SEI): o QAW e o ATAM. Resumidamente o primeiro método é responsável por elicitar os atributos de qualidade de uma arquitetura (considerados então como requisitos), tais como confiabilidade, reusabilidade, dentre outros. Como a arquitetura de um software é um ponto de equilíbrio entre o atendimento a uma série de requisitos (muitas vezes conflitantes), o método ATAM permite realizar avaliações de custo-benefício em decisões arquiteturais, face ao atendimento dos requisitos arquiteturais. Como se pode perceber os dois métodos são complementares. Por meio de uma pesquisa realizada em organizações de diferentes tamanhos de Belo Horizonte constatou-se a fraca utilização de tais métodos: apenas uma dentre as vinte e uma empresas consultadas conhecia e aplicava tais métodos.

Baseados nos resultados a segunda etapa foi marcada pelo planejamento do modo como cada esboço arquitetural seria definido e avaliado. Para tal foram estabelecidas algumas atividades e artefatos como representado pela figura 1.

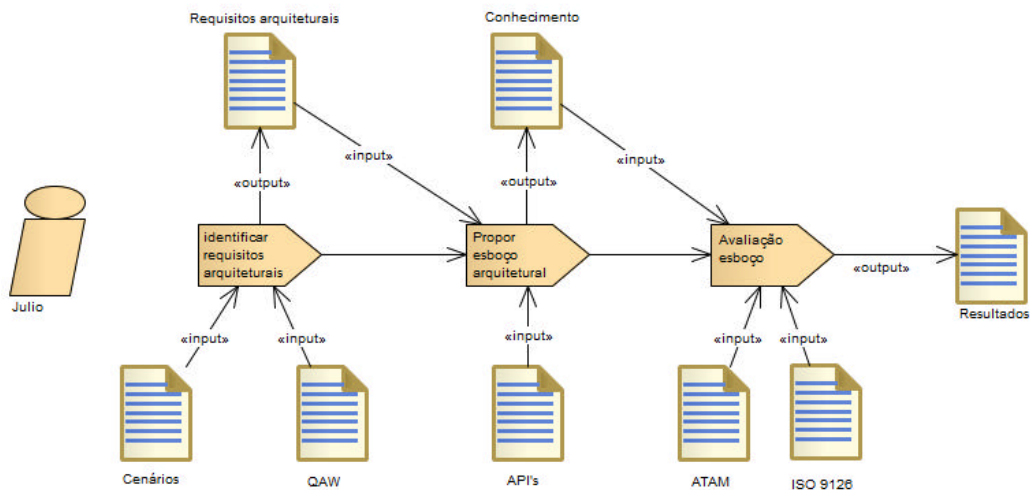


Figura 1. Atividades de definição de um esboço arquitetural para o guia de arquiteturas de referência

A primeira atividade a ser executada é responsável pela elicitação dos atributos de qualidade arquiteturais. Para a realização desta tarefa são necessários que sejam apresentados os esboços arquiteturais e o entendimento do método QAW. Como resultado da aplicação deste método sobre os cenários um artefato contendo os requisitos arquiteturais é produzido e servirá como entrada para a segunda atividade. Baseado nos requisitos elicitados e nas API's candidatas à comporem a arquitetura do software, a segunda atividade é responsável por definir e propor o esboço arquitetural. Importante notar que o resultado desta atividade não é apenas o esboço arquitetural mas também as decisões tomadas durante a sua definição, na figura representados pelo artefato "Conhecimento". A última atividade é responsável por avaliar o esboço arquitetural através da aplicação do método ATAM. Considerando que o esboço arquitetural é um produto de software deste processo de definição e avaliação, optou-se por confrontá-lo com os atributos de qualidade da norma ISO-9126 tais como confiabilidade, reusabilidade, dentre outros.

A terceira etapa foi marcada pela execução das atividades apresentadas anteriormente.

É importante ressaltar que a ISO-9126 determina muitos atributos de qualidade que estão intrinsecamente relacionados ao produto final, ou seja o software que vai ser utilizado pelo usuário (como exemplo pode-se citar a usabilidade). Fez-se necessário identificar quais os atributos da ISO (e seus graus de importância) são considerados durante a definição da arquitetura. Um novo questionário foi aplicado de modo que as respostas das organizações fossem apresentadas em uma lista de prioridades. Por exemplo: o atributo de qualidade "Confiabilidade" pode ser considerado através da "Maturidade", "Tolerância a Falhas" e "Recuperabilidade". As organizações deveriam então criar um ranking destes atributos, conforme os software que elas desenvolvem. À partir do momento em que as respostas foram coletadas foram escolhidos os seguintes atributos tolerância a falhas, precisão, operacionalidade, utilização de recursos, estabilidade e instabilidade. À partir de então foi possível executar as atividades propostas neste projeto e com base nestes atributos alguns cenários que os consideravam foram analisados pelo QAW e propostos esboços arquiteturais que posteriormente foram avaliados pelo método ATAM.

Esboço Arquitetural

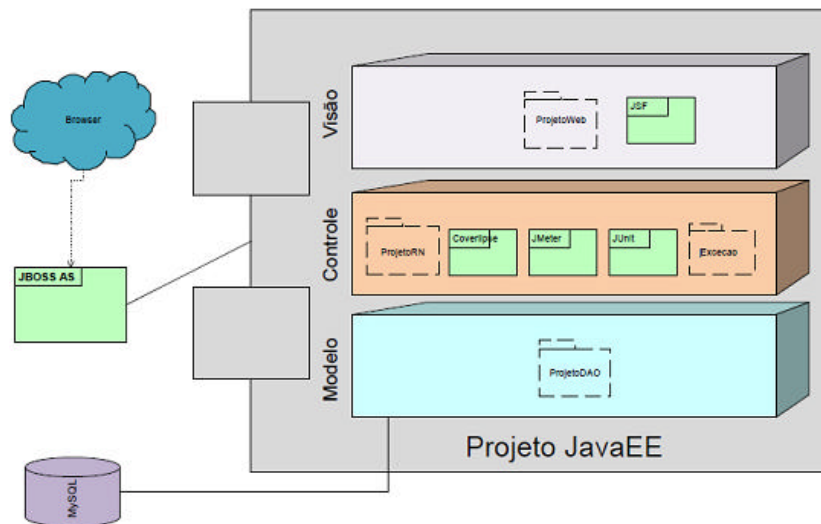


Figura 2. Primeiro esboço arquitetural criado pelo projeto, voltado para plataforma JavaEE

Era de responsabilidade da última etapa a compilação dos esboços arquiteturais no guia e sua disponibilização à comunidade. No entanto esta atividade ainda não foi possível devido ao baixo número de esboços gerados até o momento da escrita deste texto.

4. Resultados obtidos

Durante o ciclo de 2010 foram obtidos dois importantes resultados. O primeiro foi a apresentação da proposta no Encontro de Qualidade e Produtividade em Software em Brasília. O segundo resultado mais recente foi a submissão e aprovação de um artigo para o evento Computer on the Beach, a ser realizado em Florianópolis de 29 de abril a 1 de maio deste ano.

Além desses pode-se considerar também como resultado o primeiro esboço arquitetural.

Como visto na seção anterior ele tem como objetivo atender aos seguintes atributos de qualidade: tolerância a falhas, precisão, operacionalidade, utilização de recursos, estabilidade e instabilidade. Inicialmente tais requisitos foram considerados em um esboço arquitetural voltado para a plataforma JavaEE, conforme mostra a figura 2.

Conforme o método apresentado para chegar a este esboço arquitetural algumas decisões foram tomadas, o que representa o conhecimento adquirido durante sua definição. Neste caso a pilha arquitetural está sendo formada pelos elementos presentes na figura. Alguns exemplos de justificativas para o cenário em questão estão apresentados a seguir:

_ MVC é um padrão aplicado no projeto para possibilitar um software operacional onde tratamos a camada de visão separada podendo trabalhar essa camada com mais atenção para tratar o requisito Operacionalidade.

_ DAO é um padrão que acompanha o MVC atuando na camada de modelo ou seja a camada que acessa a base de dados. Importante lembra que separando as camadas diminuimos o acoplamento da aplicação.

_ Coverlipse é uma API para cobertura parcial ou total do software, identificando qual parte do código não foi chamada e quais são as mais chamadas ajudando na verificação das atualizações de versões atendendo a Estabilidade.

_ JMeter para testar a utilização dos recursos, com esta ferramenta podemos testar tempo de latência entre paginas e sessões, taxa de chegada de novos usuários, taxa de crescimento de usuários virtuais e tempo de repetição do teste.

_ JUnit é uma API para teste de unidade com ela garantimos a precisão no retorno de cada objeto conforme o combinado com o cliente. Portanto atendemos o ultimo requisito arquitetural Precisão.

5. Conclusão e perspectivas futuras

Este texto apresentou o relatório final do projeto “Definição de um guia de apoio às decisões em definições de arquiteturas de software” para o ciclo 2010 do Programa Brasileiro de Qualidade e Produtividade em Software. Apesar dos objetivos traçados no início do projeto não terem sido totalmente atingidos até o momento, os resultados obtidos têm mostrado a aceitação da proposta tanto pela comunidade acadêmica quanto pelas organizações desenvolvedoras de software. Contudo uma avaliação embasada em pesquisas nas organizações está prevista como trabalhos futuros, como uma maneira de validar a proposta.

Atualmente as propostas de esboços arquiteturais e suas avaliações têm sido feitas na plataforma JavaEE devido a familiaridade dos autores com ela. No entanto assim que a proposta estiver mais consolidada e aceita pelas organizações, vislumbra-se um cenário em que outras plataformas sejam atendidas por este projeto.

Por fim pode-se afirmar que esta pesquisa é relevante ao prover esboços arquiteturais voltados ao atendimento a atributos de qualidade que freqüentemente aparecem como requisitos de software em desenvolvimento. Deste modo acredita-se que com a utilização deste guia pelas organizações a qualidade de seus software aumente pela reutilização de protótipos arquiteturais previamente definidos e avaliados.

Referências

Clements, P., Kazman, R., and Klein, M. (2002). Evaluating Software Architectures: Methods and Case Studies. Addison Wesley.

Hofmeister, C., Nord, R., and Soni, D. (1999). Applied Software Architecture. Addison-Wesley.

Lanna, A. L. P. M. and Pietrobon, C. A. M. (2010). Reuso de processos de software baseado na componentização de processos e conhecimento. In Concurso de Teses e Dissertação em Qualidade de Software, Belém.

Sommerville, I. (2006). Software Engineering. Addison-Wesley.

Estimativa de Esforço de Teste no Auxílio da Garantia da Qualidade de Software

Priscila Pereira de Souza¹, Marcelo Werneck Barbosa¹, Anybal Rocha da Silva¹

¹Instituto de Informática – Pontifícia Universidade Católica de Minas Gerais (PUC Minas) – Belo Horizonte – MG – Brasil

priscila.souza@oi.com.br, mwerneck@pucminas.br, anybalrocha@gmail.com

Abstract. *Estimating the necessary time to perform activities specifically related to planning and executing tests is a poorly-explored task. Due to this, organizations use estimation techniques that consider the software development cycle as a whole or simply perform the same intuitively causing inaccurate results hampering project management and product quality. The objective of this study was to define a test process that would enable the application of Test Point Analysis effort estimation technique and make some adjustments to facilitate its use and to correctly and completely estimate time leading to better product quality.*

Resumo. *Estimar o tempo necessário especificamente para atividades relacionadas ao planejamento e execução de testes é uma tarefa pouco explorada. Devido a isso, as organizações optam por utilizar técnicas de estimativa de tempo que consideram o ciclo de desenvolvimento de software como um todo ou simplesmente realizam as mesmas intuitivamente causando resultados imprecisos prejudicando o acompanhamento dos projetos e a qualidade dos produtos. O objetivo deste trabalho foi definir um processo de testes que possibilitasse a aplicação da técnica de estimativa de esforço análise de pontos de teste e realizar adaptações nesta técnica a fim de facilitar seu uso e estimar tempo suficiente para a correta e completa validação dos produtos, garantido assim a qualidade dos mesmos.*

1. Introdução

Atualmente, existe um crescente interesse por parte das organizações em produzir software de maneira eficiente, com qualidade, de fácil entendimento e manutenção, a um baixo custo e em um período de tempo mínimo.

Em mercados competitivos, organizações que disponibilizam produtos com baixa qualidade podem rapidamente perder seus clientes. Consequentemente, as organizações devem garantir que seus produtos estão de acordo com as expectativas de seus clientes.

Uma atividade comum realizada para garantir a qualidade de software é a execução de testes [Aranha e Borba, 2007].

O teste de software é uma atividade que vem agregando cada vez mais qualidade ao processo de desenvolvimento e ao produto final, pois, vêm sendo enriquecido por metodologias próprias, ferramentas de automação e equipes independentes e capacitadas. Mas, assim como os projetos de desenvolvimento de software, os projetos de teste possuem os mesmos desafios e problemas em relação a custo e prazo. Por isso, é importante definir bem um processo de teste e estabelecer estimativas precisas e controláveis ao longo do projeto. Se por um lado valores superestimados elevam os prazos e os custos dos projetos, causando insatisfação dos clientes e prejudicando a competitividade das empresas de desenvolvimento, por outro, valores subestimados causam cronogramas mal dimensionados e baixa qualidade dos produtos. Os gerentes de testes devem ser capazes de planejar seus recursos apropriadamente, estimar o esforço necessário para executar um dado conjunto de testes e justificar demandas por mais recursos ou por extensões de prazo [Aranha e Borba, 2007].

Estimar o esforço de testes não é uma tarefa simples, pois há diversos fatores internos e externos que podem influenciar o resultado. Além disso, é um tema pouco explorado. Existem poucos relatos de sua utilização na literatura e, consequentemente, pouco conhecimento sobre a efetiva e correta utilização de técnicas de estimativas de esforço específicas para testes pelas organizações. No entanto, há algumas técnicas que podem possibilitar a obtenção de estimativas mais precisas ao considerar os fatores que afetam as atividades de testes.

Dentre as técnicas de estimativa de esforço para teste de software, a Análise de Pontos de Teste (APT) [Veenendaal e Dekkers, 1999] é, segundo [Rios, 2006] considerada na literatura como a mais consistente. Mas, apesar dos autores da técnica afirmarem que a mesma foi utilizada por algumas organizações e obtiveram bons resultados, não foram encontrados relatos detalhados ou medições que comprovassem tais resultados.

A APT é uma técnica de estimativa de tamanho de teste que gera previsões de esforço e tempo. É utilizada para estimar o esforço para definir, desenvolver e executar testes funcionais. A APT é baseada na técnica Análise de Pontos de Função. A Análise de Pontos de Teste considera relevante, além do tamanho funcional determinado pelos pontos de função, outros dois elementos: a estratégia de testes e a produtividade [Veenendaal e Dekkers, 1999].

O maior benefício da Análise de Pontos de Teste está em conseguir reunir de forma sistemática os fatores que influenciam o esforço do teste de software. Além disso, permite avaliar o esforço de teste por atividade e leva em consideração as etapas de planejamento e controle que são essenciais para o sucesso de qualquer projeto e devem ser consideradas em técnicas de estimativas de esforço.

A APT é uma técnica que permite estimar o esforço independentemente do processo de desenvolvimento e considera as características de qualidade do sistema, o fator do ambiente de teste e a qualificação e produtividade da equipe envolvida, o que possibilita uma correta estimativa proporcionando tempo suficiente para a equipe de testes realizar suas atividades garantindo assim a qualidade dos produtos.

2. Objetivos e Justificativa

As atividades de testes fazem parte do ciclo de vida do desenvolvimento do software, e muitas vezes, as organizações não realizam estimativas específicas quanto ao esforço de testes. Geralmente, é realizada uma estimativa de esforço baseada em pontos de função para todo o desenvolvimento de software e designada uma porcentagem deste esforço para as atividades de testes. Mas, essas estimativas incluem apenas os testes executados pela equipe de desenvolvimento (testes unitários), por isso, há necessidade de uma medida específica para o esforço de teste de software referente à equipe de testes [Rios e Moreira, 2006].

A realização de estimativas de esforço realizada de maneira inadequada pelas organizações pode influenciar diretamente na qualidade do produto final, pois realizando uma estimativa de esforço incorreta o tempo alocado para a equipe de testes muitas vezes não é suficiente para a execução de todas as suas atividades.

Este trabalho apresenta a definição de um processo de teste de software composto por diversas atividades, artefatos e ferramentas que possibilitam a aplicação de uma versão adaptada e detalhada da técnica Análise de Pontos de Teste a fim de padronizar as atividades de testes, facilitar a utilização da técnica e alcançar estimativas mais confiáveis e precisas para a correta e completa execução dos testes garantindo assim a qualidade dos produtos.

3. Metodologia de Execução

O presente trabalho seguiu uma metodologia experimental apoiada em referencial teórico da área. A partir da leitura das referências bibliográficas levantadas a cerca de trabalhos realizados, foram adotadas as seguintes etapas a fim de alcançar o objetivo proposto:

A primeira etapa deste trabalho consistiu em avaliar as técnicas de estimativa de esforço de teste de software existentes e selecionar a considerada mais consistente. A partir da seleção da técnica, Análise de Ponto de Teste (APT), a segunda etapa consistiu na realização de algumas adaptações para facilitar e auxiliar a aplicação da mesma.

Na terceira etapa, foi proposto um processo constituído de diversas atividades e artefatos com foco na técnica selecionada a fim de alcançar resultados mais precisos e confiáveis. Foi desenvolvida uma ferramenta para automatizar a aplicação da técnica APT adaptada para auxiliar a realização das estimativas.

Na etapa seguinte, estudos de caso foram realizados aplicando o processo de testes juntamente com a técnica adaptada em uma empresa privada e em um projeto de desenvolvimento de software na disciplina de trabalho de conclusão de curso.

Para finalizar, foi realizada uma avaliação dos resultados obtidos com aplicação do processo, contrapondo os resultados reais com as estimativas realizadas e relatando as dificuldades encontradas. Após essa avaliação foi possível analisar a eficácia do processo e da APT adaptada.

4. Resultados Obtidos

4.1. Produtos de software gerados

A ferramenta APTool utilizada no processo proposto e desenvolvida neste trabalho, automatiza o processo de estimativa e auxilia a aplicação da técnica Análise de Pontos de Teste Adaptada. Além de executar os cálculos e gerar a estimativa de esforço para o teste de software, a ferramenta armazena as estimativas calculadas, permitindo uma análise comparativa e a recuperação desta informação.

A APTool permite ainda a geração de bases históricas das estimativas dos projetos e é uma fonte para a elaboração de indicadores. As principais funcionalidades bem como a descrição técnica da APTool estão em uma versão mais completa deste artigo.

4.2 Processos que contribuem para a melhoria da qualidade de software descritos e institucionalizados

O processo de testes proposto possui dois papéis:

- ? Projetista de testes: responsável pelas atividades de planejamento de testes, incluindo a atividade de realização de estimativas.
- ? Testador: responsável pelas atividades de execução dos testes planejados.

Conforme a Figura 01, o processo de testes proposto é constituído por 5 atividades principais: Definir Rastreabilidade dos Requisitos, Planejar Testes, Realizar Estimativas, Executar Testes e Concluir Testes. A descrição destas atividades pode ser encontrada em uma versão mais completa deste artigo.

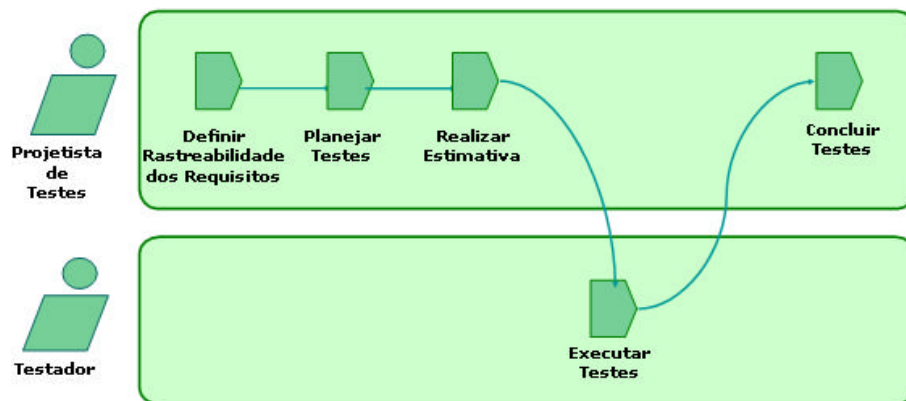


Figura 01 – Processo de Testes Proposto

4.3. Métodos e/ou algoritmos desenvolvidos

A técnica Análise de Pontos de Teste não é explícita na identificação de onde e como cada informação necessária para a realização do cálculo da estimativa pode ser encontrada. E, não explica claramente como cálculos ou classificações são realizadas e também não leva em consideração alguns aspectos importantes do processo de testes. Por esses motivos, foram necessárias algumas adaptações na técnica, descritas em detalhe no tutorial elaborado neste trabalho e em uma versão mais completa deste artigo.

4.4. Estudos de Casos Realizados

Para validar o processo de testes proposto e demonstrar a aplicabilidade da técnica APT adaptada, foram realizados alguns estudos de caso em uma empresa privada e em um sistema desenvolvido para a disciplina de Trabalho de Conclusão do Curso de Sistemas de Informação na PUC Minas. A seguir segue a descrição de alguns dos estudos realizados. Os demais resultados podem ser encontrados em uma versão mais completa deste artigo.

A empresa na qual os estudos de caso foram realizados, é situada em Contagem - MG e reconhecida por ser uma empresa que prima pela excelência no desenvolvimento de soluções de alta tecnologia eletrônica, de software e de serviços.

A empresa possui uma equipe independente e capacitada para a realização das atividades de teste de software e firmware, porém, não aplica nenhuma técnica para realizar estimativas de esforço. As mesmas são realizadas intuitivamente pelo gerente de projetos baseando-se apenas na experiência da equipe de testes. Todo o processo de testes proposto foi aplicado na organização bem como a técnica de estimativa de esforço Análise de Ponto de Testes com as devidas adaptações.

O projeto piloto foi realizado em um software que visa comunicação local com um sistema inteligente de medição, utilizando comandos de leitura e alteração. Este projeto será chamado de Projeto A.

Um dos testadores da organização foi responsável por executar as atividades definidas no processo proposto bem como realizar o preenchimento de todos os artefatos definidos para essas atividades. O apontamento do esforço foi realizado pelo utilizando a ferramenta de gerenciamento de horas da própria organização.

O testador foi monitorado durante a execução dessas atividades a fim de identificar dificuldades na utilização da técnica e no preenchimento dos artefatos e para constatar se o esforço estava sendo apontado realmente conforme o realizado.

De acordo com o observado na Tabela 01, o número total de horas previstas de testes para o Projeto A alcançado com as estimativas através da aplicação da APT adaptada no processo proposto foi igual a 58,96 horas. A estimativa prevista pelo gerente de projetos para se realizar as atividades de testes para este mesmo projeto, foi de 30 horas. O total de horas real gastas para executar as atividades de testes deste projeto foi igual a 54 horas (de acordo com o apontamento realizado).

Podemos observar que a estimativa realizada aplicando a técnica APT adaptada obteve um resultado melhor em comparação à estimativa realizada intuitivamente pelo gerente de projetos. Além disso, a diferença foi de apenas 9,19% em comparação ao número de horas realmente executado.

O segundo projeto no qual o processo proposto juntamente com a técnica APT adaptada foi aplicado é um projeto de firmware de um medidor de energia. Mesmo sendo um software embarcado, a organização aplica a

técnica Análise de Pontos de Função para dimensionar o tamanho do sistema e isso possibilitou a aplicação da APT. Este projeto será chamado de Projeto B.

A estimativa do gerente de projetos para o esforço de testes deste projeto foi realizada de acordo com a experiência da equipe que iria planejar e realizar os testes. O valor alcançado foi de 89,36 horas. Com a aplicação da técnica APT adaptada no processo proposto, a estimativa obtida foi de 153,12 horas. Após a conclusão de todo o processo de teste o tempo efetivamente gasto pela equipe de testes foi de 209,00 horas.

A estimativa realizada pela APT adaptada não foi tão próxima do esforço realizado como no primeiro estudo de caso, porém mais uma vez o seu valor obtido teve uma taxa de erro menor do que a estimativa realizada pelo gerente de projetos, que não utilizou nenhuma técnica para realizar a mesma.

O terceiro estudo de caso foi aplicado em um sistema desenvolvido como trabalho de conclusão de curso de um aluno de sistemas de informação da PUC Minas. Trata-se de uma plataforma Web que possibilita o armazenamento de requisitos de software de uma organização a fim de viabilizar a reutilização. Este projeto será chamado de Projeto C.

Como este estudo de caso não teve um gerente de projetos foi realizada uma pesquisa na literatura a fim de encontrar a produtividade para se testar um sistema com as mesmas características. Em função da confidencialidade de dados de produtividade, não foi possível calcular o esforço específico para a fase de teste com base em produtividades de mercado, porém foi encontrada a produtividade de 11 horas por ponto de função para toda a fase de desenvolvimento (ISBSG). Vale ressaltar que essa produtividade é uma média para projetos realizados em diversas linguagens, plataformas, organizações, etc. Acreditamos que no caso de um sistema simples, desenvolvido por apenas um profissional e na linguagem PHP, a produtividade seria maior, entretanto, não foram encontradas referências informando a produtividade em um projeto com essas características.

De acordo com a produtividade encontrada, o esforço total do projeto seria de 836 horas ($11h \times 76PF$). Para calcular o esforço específico da fase de testes, é necessário calcular o número de horas com base em alguma distribuição média de esforço por fase. Foi utilizado o esforço recomendado por [Pressman, 2005], que seria de 30% a 40% do esforço total do projeto. Assim, a estimativa de esforço considerada para os testes foi de 260,70 horas. Ao aplicar a APT adaptada para a realização da estimativa foi obtido o valor de 85,71 horas. O esforço real utilizado para validar o sistema foi de 63,30 horas. Um resultado ainda melhor poderia ter sido alcançado neste projeto. Devido à falta de recursos e à realização de outras atividades em paralelo ao planejamento e execução dos testes alguns casos de testes não foram criados impactando assim em um tempo gasto inferior ao estimado.

O quarto estudo de caso foi realizado em um projeto para desenvolver uma aplicação de Palm. Este projeto é chamado de Project D. A estimativa de esforço do gerente de projeto para validar este software foi realizado apenas com base na contagem de pontos de função. O valor estimado por ele foi igual a 140,00 horas. A técnica APT adaptada obteve a estimativa de 290,96 horas. Após a conclusão do processo de teste completo, o trabalho real gasto pela equipe de teste de acordo com o apontamento realizado foi igual a 252,26 horas. A estimativa utilizando a APT adaptada obteve um resultado melhor em comparação com a estimativa realizada pelo gerente de projetos. A diferença foi de 15,34% em relação ao número real de horas realizadas

Tabela 01 - Comparativo entre os projetos

Projeto	Número de Pontos de Função	Número de Horas Estimadas pelo Gerente	Número de Horas Estimadas pela Técnica Adaptada	Tempo Efetivamente Gasto (em horas)	Margem de erro do Gerente (%)	Margem de erro da Técnica Adaptada (%)
Projeto A	45	30,00	58,96	54,00	44,44	9,19
Projeto B	19	89,40	153,12	209,00	57,24	26,74
Projeto C	79	260,70	85,71	63,30	311,85	35,40
Projeto D	129	104,00	290,96	252,26	58,77	15,34

5. Aplicabilidade dos Resultados

Em todos os estudos, foi aplicado o processo proposto juntamente com a técnica APT adaptada. Os resultados foram satisfatórios, pois alcançaram estimativas mais próximas do esforço realizado, contradizendo alguns trabalhos relacionados [Almeida et al, 2008] e [Lopes e Nelson, 2008] nos quais a margem de erro chegou até a 350% de diferença. Isso mostra que o fato da empresa ter um processo bem definido, com informações disponíveis para a aplicação da APT e um bom entendimento da técnica podem permitir alcançar resultados mais precisos e confiáveis.

6. Características Inovadoras

O resultado deste trabalho é algo novo, pois são realizadas adaptações na técnica APT aplicada em um processo proposto e foi desenvolvida uma ferramenta específica para auxiliar na realização de estimativas de esforço de teste.

7. Conclusão e Perspectivas Futuras

O propósito deste presente trabalho foi definir um processo de teste composto por diversas atividades e artefatos que possibilitasse a aplicação da técnica de estimativa de esforço Análise de Pontos de Testes. Esta técnica pode ser considerada de difícil utilização, pois não é clara quanto à realização dos cálculos da estimativa e também não leva em consideração alguns aspectos importantes do processo de testes. Por esses motivos, foram necessárias algumas adaptações. Além disso, foi desenvolvida uma ferramenta e elaborado um tutorial para auxiliar as organizações durante a realização das estimativas.

Alguns estudos de caso foram realizados para ilustrar a aplicabilidade do processo e da técnica adaptada. Os estudos de caso foram aplicados em uma organização e para o sucesso do nosso trabalho foi necessária uma mudança de cultura na mesma. Todos os envolvidos no processo de desenvolvimento de software foram convencidos da importância de se preencher corretamente todos os artefatos de entrada no processo de teste de software para obter todas as informações necessárias na realização das estimativas. Os testadores também foram orientados sobre a importância do exato apontamento das horas.

Através dos resultados obtidos foi possível verificar que o processo atendeu ao seu objetivo que era permitir alcançar estimativas mais precisas e confiáveis e auxiliou na criação de uma base histórica consolidada permitindo assim a criação de indicadores nesta organização. A técnica APT adaptada alcançou resultados muito melhores em comparação com os recentes trabalhos relacionados [Almeida et al, 2008] e [Lopes e Nelson, 2008] que obtiveram resultados discrepantes de até 350% de margem de erro e em relação as estimativas realizadas pelo gerente de projetos.

A obtenção dos resultados positivos através do presente trabalho divulga a técnica Análise de Pontos de Teste na literatura demonstrando a importância da utilização de técnicas específicas para realizar a estimativa de esforço de teste e contribui com os gerentes de projetos e ou responsáveis pelo processo de desenvolvimento de software fornecendo instrumentos detalhados que auxiliam na aplicação da APT adaptada. Além disso, proporcionou alocação de tempo adequado para a validação dos produtos garantindo assim a qualidade dos mesmos.

Como possíveis trabalhos futuros, pretende-se aplicar o processo de testes proposto juntamente com a técnica APT adaptada em fábricas de software e atualmente esta sendo aplicada em uma grande empresa de energia elétrica de Minas Gerais. Pretende-se ainda propor melhorias e outras adaptações na técnica para obter estimativas ainda mais precisas. As melhorias poderão ser propostas após a execução de mais estudos de casos e observação de fatores de influência nos valores estimados de acordo com as características do projeto.

8. Referencias Bibliográficas

Almeida, E. R. C., Abreu, B. T., Moraes, R. e Martins, E.. Avaliação de um Método para Estimativa de Esforço para Testes baseado em Casos de Uso. In VII Simpósio Brasileiro de Qualidade de Software. Florianópolis. Junho 2008.

Aranha, E. and Borba, P (2007). An Estimation Model for Test Execution Effort. In First International Symposium on Empirical Software Engineering and Measurement.

Filho, T.R.M. Rios, E. (2006). Teste de Software. Rio de Janeiro: Alta books.

ISBSG: International Software Benchmarking Standard Group. <http://www.isbsg.org.au>

Lopes, F. A.; Nelson, M. A. V. (2008). Análise das técnicas de estimativas de esforço para o processo de teste de software. In: III Encontro Brasileiro de Testes de Software, 2008, Pernambuco. Anais do III Encontro Brasileiro de Testes de Software.

Pressman, R. S (2005). Engenharia de software. 3ª Edição, São Paulo: ed. McGraw-H

Veenendaal, E.V., Dekkers, T. (1999) Test point analysis: a method for test estimation, Project Control for Software Quality, Editors, Rob Kusters, Adrian Cowderoy, Fred Heemstra and Erik van Veenendaal. Shaker Publishing.

FAMP: Ferramenta de Avaliação para Melhoria de Processos de Desenvolvimento de Software

Elson Cardoso Siquara¹, Fábio Britto de Carvalho Almeida², Laís do Nascimento Salvador³

¹²Área de Tecnologia da Informação e Telecomunicações (ATI) - SENAI Salvador, BA - Brasil

³ Departamento de Ciência da Computação (DCC) Universidade Federal da Bahia (UFBA) - Salvador, BA - Brasil

elsoncs@fieb.org.br, falmeida@fieb.org.br, laisns@dcc.ufba.br

Abstract. *The use of the assessment methods SCAMPI and MA-MPS is not trivial, requires a detailed planning, a great amount of information, qualified professionals and the evaluation team commitment. In that way, the use of computational systems can contribute in the execution of these processes. This paper presents the project's tool development FAMP (Process Improvement Evaluation Tool) that aims to support described activities at SCAMPI and MA-MPS methods. Two of its features are highlighted: (i) action plan generation and follow-up to implementation gaps of the CMMI and MPS.BR model and (ii) web-based architecture.*

Resumo. *A utilização dos métodos de avaliação SCAMPI e MA-MPS não é trivial, requer um planejamento detalhado, levantamento de um grande volume de informações, profissionais capacitados e um amplo comprometimento da equipe. Sendo assim, a utilização de sistemas computacionais pode muito contribuir na execução desses processos. Este artigo apresenta o projeto de desenvolvimento da ferramenta FAMP (Ferramenta de Avaliação para Melhoria de Processos) que tem como objetivo principal apoiar as atividades descritas nos métodos SCAMPI e MA.MPS, com os seguintes diferenciais: (i) a geração e acompanhamento de planos de ação para as lacunas de implementação do modelo CMMI e MPS.BR e (ii) arquitetura destinada à utilização em ambiente web.*

1. Introdução

Atualmente grande parte da população mundial depende de aplicações de software para realizar as suas atividades diárias. Se alguns sistemas de uso global deixarem de funcionar, aproximadamente 40% da população mundial sofrerá as consequências do problema [REED, 2000 apud ROCHA; MALDONADO; WEBER, 2001, p.1]. Diante disto, a questão da qualidade de software para os seres humanos passa a ter uma grande relação com a sua própria evolução. A preocupação das organizações que produzem software com relação à garantia da qualidade tornou-se evidente no mercado regido pela livre concorrência, aquelas capazes de integrar, harmonizar e acelerar seus processos de desenvolvimento e manutenção terão a primazia no mercado [CURTIS, 2000 apud

ROCHA; MALDONADO; WEBER, 2001, p.2]. Portanto, tem sido um ponto determinante a busca por padrões que forneçam soluções que garantam um alto grau de competitividade, o que resulta na escolha dos modelos e normas que tratam da melhoria da qualidade através do aprimoramento dos processos da organização para alcançar a qualificação de seus produtos [SOFTEX, 2010].

O Brasil tem vivenciado um crescimento bastante significativo com relação ao investimento na melhoria e qualidade dos produtos de software. Diversas empresas iniciaram programas de melhoria com base no *Capability Maturity Model Integration* (CMMI) do *Software Engineering Institute* (SEI) e na *Melhoria de Processos do Software Brasileiro* (MPS.BR) da SOFTEX como modelos de referência. Para verificar o cumprimento dos requisitos destes modelos nas organizações, foram definidos os métodos *Standard CMMI Appraisal Method for Process Improvement* (SCAMPI) e *Método de Avaliação para Melhoria dos Processos de Software* (MA-MPS). Entretanto, a utilização do SCAMPI e do MA-MPS não é trivial, requer, dentre outros requisitos, um planejamento detalhado, levantamento de um grande volume de informações, profissionais capacitados e um grande comprometimento da equipe de avaliação. Para tanto, se faz necessária a utilização de ferramentas que forneçam subsídios ao processo de avaliação com precisão nos resultados obtidos e que favoreçam o planejamento eficiente das organizações a partir da identificação das necessidades de melhoria.

Nas pesquisas realizadas, a maioria das ferramentas de avaliação encontradas tem como principal função automatizar as atividades da norma ISO/IEC 15504 (SPICE). [Xavier, 2007] propôs a ferramenta ProEvaluator para apoiar as avaliações do modelo MPS.BR e dentre as outras pesquisadas, por sua vez a *Appraisal Assistant* do *Software Quality Institute* [SQI, 2011] foi a única localizada que suporta substancialmente avaliações do modelo CMMI. Diante disso, este artigo apresenta o projeto de desenvolvimento de uma ferramenta com características técnicas e funcionais para automatizar as atividades descritas no método SCAMPI e MA-MPS, considerando a aplicação destes em organizações que busquem melhoria em seus processos, fundamentando-se nas premissas e boas práticas do CMMI e do MPS.BR.

Corroborando esta introdução que representa o problema tratado pelo projeto, este artigo é organizado da seguinte forma: a Seção dois apresenta os objetivos e justificativa do projeto. Seguindo o objetivo deste documento, a Seção três aborda a metodologia de execução do projeto. Posteriormente, nas Seções quatro, cinco e seis são explanados os resultados obtidos, a aplicabilidade dos resultados e características inovadoras respectivamente. A Seção sete apresenta a conclusão com os benefícios esperados e as possibilidades de desdobramento em estudos futuros.

2. Objetivos e Justificativa

A institucionalização da cultura da qualidade nos processos contribui para a competitividade das empresas produtoras de software. Processos de software com qualidade levam à melhoria do produto, diminuem o retrabalho e conseqüentemente, aumentam a produtividade e reduzem o tempo de entrega do produto ao mercado. O primeiro passo para a definição de processos de software com qualidade é avaliar a qualidade das práticas da organização produtora do software.

Com base nessa avaliação, são propostas ações de melhoria que culminam com a definição e implantação de melhores práticas. No entanto, o trabalho de avaliação da qualidade de processos de software é árduo, complexo, e depende da experiência e do conhecimento da equipe responsável pelo diagnóstico de processos. Além disso, cada vez mais, as empresas de TI necessitam de maior visibilidade no mercado através da obtenção das certificações. Como consequência nota-se um crescimento no Brasil nos últimos anos do número de avaliações realizadas pelas empresas para obtenção de certificações. Neste sentido, é bastante relevante que estas se assegurem quanto ao conhecimento dos métodos de avaliação dos modelos de qualidade de software.

Uma vez providas de conhecimento sobre os métodos, as empresas podem realizar auto-avaliações e planejar gradativamente a melhoria de seus processos até estarem aptas a obterem a certificação. Porém, questões relevantes surgem quanto ao impacto dessa iniciativa no que diz respeito à eficiência das auto-avaliações em curto prazo e posteriormente, do impacto das modificações planejadas a serem implantadas no processo corrente. Neste ponto se faz necessária a utilização de ferramentas que acelerem o processo, garantindo precisão nos resultados obtidos e permitindo às organizações se planejarem eficientemente.

Além disto, diante da quantidade de atividades relacionadas a um processo de avaliação, o uso de ferramentas computacionais que possam ajudar na integração da equipe, coleta dos dados e geração instantânea dos resultados é de grande importância para as empresas na obtenção de certificações em modelos de qualidade como o CMMI e o MPS.BR. Como exemplo desta complexidade, a coleta de muitos ou poucos dados pode ser problemática para a avaliação, resultando em uso ineficiente de recursos, que já são escassos para as empresas. Sendo assim, este projeto visou promover a utilização de modelos de qualidade de software em pequenas e médias empresas, através da especificação, desenvolvimento e implantação de uma ferramenta de avaliação para melhoria de processos, com características técnicas e funcionais inovadoras, para apoiar as atividades descritas nos métodos de avaliação SCAMPI e MA-MPS dos modelos CMMI e MPS.BR.

3. Metodologia de Execução

No início do projeto foi elaborado um plano de trabalho com detalhamento sobre a constituição da equipe e todas as atividades identificadas. Além dos profissionais alocados diretamente no projeto, foram definidos outros integrantes que participaram indiretamente no apoio operacional de compras, jurídico e financeiro. Simultaneamente à constituição da equipe, foi realizada a compra dos materiais permanentes.

A equipe do projeto foi dividida em dois grupos. Um grupo ficou responsável pelo estudo do modelo CMMI e o método de avaliação SCAMPI e o outro pelo estudo do modelo MPS.BR e o método MP-MPS. As equipes foram compostas por analistas de sistemas e bolsistas de iniciação tecnológica, que tiveram o apoio do coordenador do projeto. Após a conclusão dos estudos das equipes, foram realizados seminários e reuniões com o objetivo principal de alcançar a transferência dos conhecimentos adquiridos entre todos os participantes do projeto. A partir desse ponto, iniciou-se uma análise comparativa entre os modelos de qualidade, que serviu de base para a especificação da ferramenta. Na fase de desenvolvimento da ferramenta, um grupo fez a especificação do módulo de avaliação MPS.BR/MA-MPS e o outro fez a especificação do módulo de avaliação CMMI/SCAMPI. Após isto, o projeto arquitetural do sistema foi desenvolvido em conjunto com especial atenção na integração entre estes dois módulos. Após aprovado o projeto do software, foi iniciada a fase de codificação da ferramenta, que teve as seguintes características técnicas: (i) Modelagem Orientada a Objetos utilizando UML; (ii) Independência de Sistema Operacional; (iii) Tecnologias Java EE (Servlets, JSP e JSF); (iv) Padrões de projetos Singleton, Abstract Factory e MVC; (v) Mapeamento objeto/relacional Hibernate com banco de dados Postgre.

Na fase final do projeto, o software desenvolvido foi implantado na empresa parceira. Para isto, os colaboradores foram treinados tanto para o uso do sistema quanto em relação aos processos de avaliação. Em seguida, para diagnóstico da empresa, a avaliação foi dividida em três etapas, sendo elas: (i) Planejamento da avaliação responsável por definição de objetivos, restrições e escopo. (ii) Condução da avaliação composta por

busca e análise das evidências objetivas, identificação das lacunas de implementação das práticas e geração de planos de ação. (iii) Geração dos resultados da avaliação no âmbito geral e por áreas de processo.

4. Resultados Obtidos

Como principal resultado deste projeto pode-se destacar o desenvolvimento da Ferramenta de Avaliação de Processos de Software (FAMP), adequada aos métodos de avaliação dos modelos de qualidade CMMI e MPS.BR.

A FAMP tem como objetivo geral promover a melhoria da gestão das atividades realizadas pela equipe envolvida nas avaliações dos processos abordados pelo CMMI e MPS.BR. Ela foi especificada com base nas orientações descritas pelos métodos SCAMPI e MA-MPS e possui as seguintes características: (i) pode ser utilizada como suporte a avaliações oficiais ou em auto-avaliações de organizações de qualquer tamanho, como forma de identificação das melhorias necessárias; (ii) proporciona suporte para identificação dos pontos fortes e fracos; (iii) fornece subsídios para o acompanhamento dos planos de ação elaborados quando identificadas lacunas de implementação do modelo; (iv) provê a conversão entre os modelos CMMI e MPS.BR, ou seja, se a empresa está em um determinado nível do MPS.BR, a ferramenta permite a visualização do nível correspondente desta empresa no CMMI e vice-versa. O fluxo de atividades da ferramenta está representado na Figura 1.

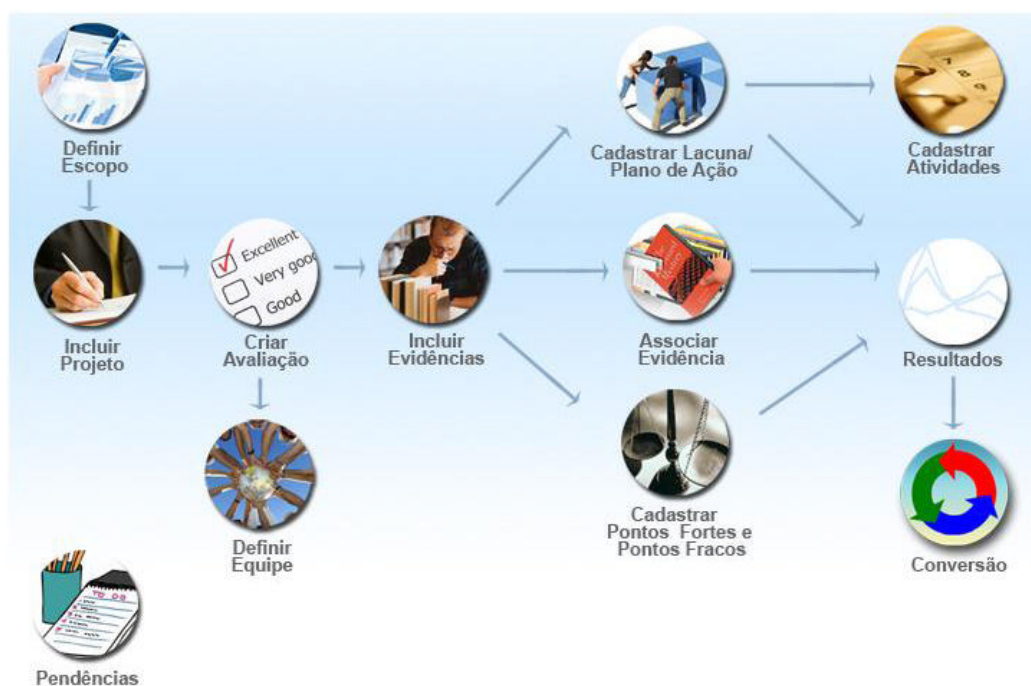


Figura 1. Fluxo de atividades da ferramenta

Conforme apresentado na Figura 1, para início do uso da FAMP, é necessário definir o escopo da avaliação, sendo que antes deve ser feito o cadastramento da estrutura do modelo de qualidade que será utilizado na avaliação – esta funcionalidade foi implementada a fim de suportar a extensão da ferramenta para futuras versões do CMMI ou MPS.BR. A ferramenta proporciona o cadastramento de várias estruturas de modelos, sendo que cada estrutura cadastrada pode ser utilizada por mais de uma avaliação. Não há obrigação em definir como escopo uma estrutura de modelo completa, por exemplo, caso a organização esteja buscando o nível 2 de maturidade do CMMI, o líder de avaliação pode escolher apenas as áreas de processo deste nível. Desta forma são usados componentes do modelo mais apropriados para a avaliação com uma definição precisa do escopo e com foco nas áreas de processo ou processos relevantes. Para completar a rotina de inclusão de uma avaliação é também necessário o cadastramento prévio dos participantes da equipe de avaliação com suas permissões no sistema e o registro dos projetos associados.

Durante o cadastramento das avaliações, pode-se fazer o registro de informações gerais (título, objetivo, data do início, data do fim, responsável). É possível também o cadastramento de avaliações, que assumem diferentes status (ativa, concluída, cancelada e atrasada) e servem de base histórica para futuras consultas. Após a abertura de uma avaliação, os usuários membros da equipe, líder de avaliação e administrador cadastrados podem iniciar a inclusão das evidências objetivas (documentos e entrevistas). Essas evidências serão depois vinculadas às práticas ou resultados esperados do modelo por meio da aprovação dos indicadores com a permissão de acesso de líder de avaliação. A Figura 2 apresenta um exemplo da tela de inclusão de indicador de uma avaliação CMMI na ferramenta, onde pode ser observado do lado esquerdo da imagem, a estrutura do modelo com a sinalização de cumprimento dos componentes com a utilização de cores.

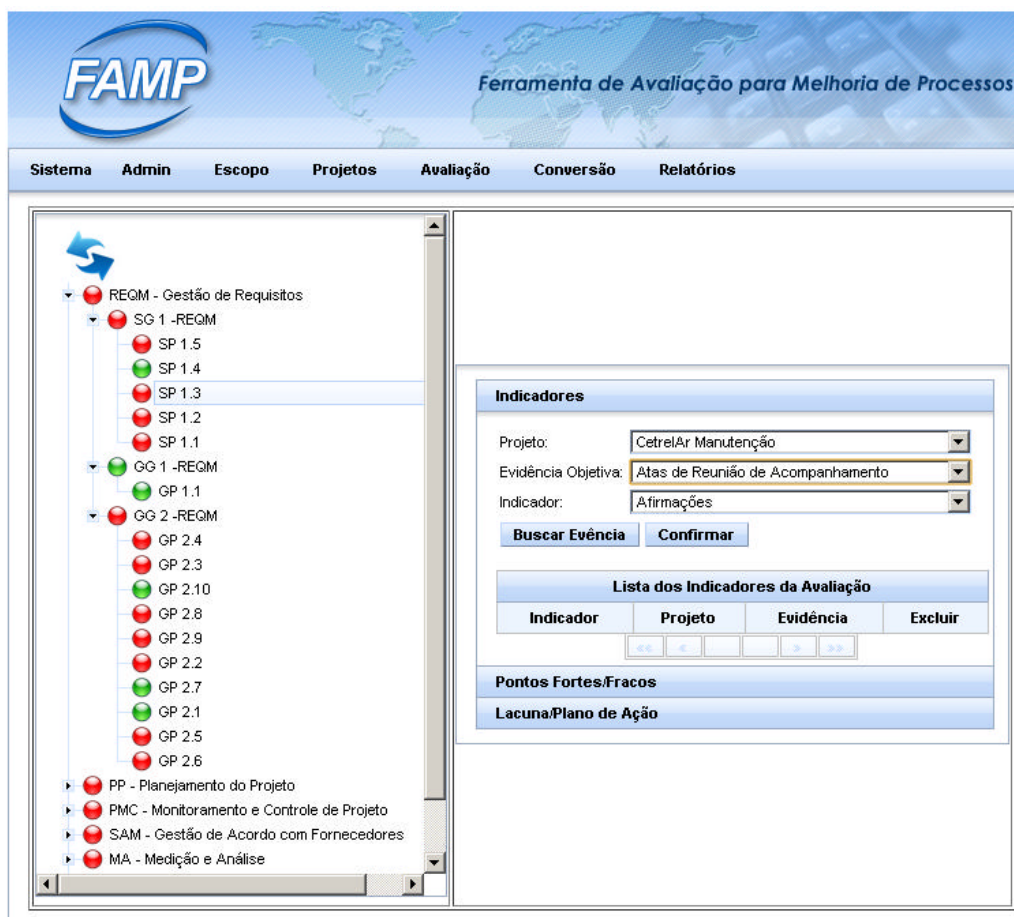


Figura 2. Tela de inclusão de indicador

Para registro de um indicador, na estrutura de um modelo selecionado, ao acessar uma prática ou resultado esperado, o líder de avaliação ou membro da equipe, faz a vinculação com uma evidência objetiva por meio da inserção do tipo (artefato direto, artefato indireto e afirmação).

A partir do registro de lacunas de implementação das práticas ou resultados esperados, a ferramenta obriga a abertura de plano de ação para solução do problema. Além disso, a qualquer momento podem ser sinalizados pontos fortes ou fracos para qualquer componente do modelo. Para auxiliar a identificação das lacunas de implementação e o acompanhamento dos consequentes planos de ação, a ferramenta possui uma funcionalidade que apresenta as pendências das avaliações, através de uma listagem dos componentes que possuem lacunas de implementação sem plano de ação, planos de ação sem atividades e planos de ação com atividades em atraso.

A geração de resultado das avaliações na ferramenta FAMP é feita para os dois tipos de representação do CMMI (em estágios e contínua) e para o MPS.BR e leva em consideração os seguintes critérios: (i) presença dos indicadores relacionados às práticas ou resultados esperados; (ii) lacunas de implementação identificadas; (iii) situação dos planos de ação; (iv) presença de pontos fortes e fracos; (v) validação da implementação dos componentes do modelo pelo líder de avaliação.

4.1. Principais funcionalidades

As principais funcionalidades implementadas pela ferramenta FAMP são: (i) Cadastramento da estrutura do modelo com informações dos componentes (áreas de processo, metas e práticas) descritas no manual do CMMI e SCAMPI e dos componentes (processos, atributos do processo e resultados esperados) descritas no manual do MPS.BR e MA-MPS; (ii) Geração e acompanhamento dos planos de ação para as lacunas de implementação das práticas e resultados esperados; (iii) Representação hierárquica do modelo de avaliação – A situação dos componentes do modelo é representada por cores e códigos, o que facilita a compreensão e a tomada de decisão por parte dos envolvidos na avaliação; (iv) Controle de níveis de permissão dos usuários do sistema (patrocinador, representante de unidade organizacional, líder de avaliação, membro da equipe de avaliação e participante); (v) Acompanhamento das pendências da avaliação – Lista das práticas não implementadas (NI), parcialmente implementadas (PI) e atividades dos planos de ações não concluídas ou em atraso.

4.2. Outros Resultados Obtidos

Após o desenvolvimento da FAMP foi elaborada uma metodologia para avaliação e implantação da ferramenta na empresa parceira do projeto. Esta metodologia foi constituída de três etapas que são elas: treinamento dos participantes para utilização da ferramenta, consultoria e auditoria. Ao final dessas etapas foi gerado na FAMP um diagnóstico dessa empresa em relação ao cumprimento do modelo de qualidade MPS.BR (Nível G) e um relatório com os planos de ação para lacunas de implementação identificadas. Como resultado deste processo, notou-se uma contribuição para a melhoria dos processos de desenvolvimento de software, imagem, produtividade e competitividade da empresa piloto. Este projeto também colaborou para internalização e transferência de conhecimento entre os participantes, tendo como fruto a publicação de artigo no Simpósio Brasileiro de Engenharia de Software - SBES 2008 [Siquara, Salvador 2008].

5. Aplicabilidade dos Resultados

De acordo com a pesquisa da ABES (2010) em parceria com a IDC, o setor de software e serviços no Brasil tem mostrado um bom crescimento. Ao menos, isso é o que indicam os números: 15,03 bilhões de dólares movimentados em 2009, um crescimento de 2,4% sobre o ano anterior. Desse total 5,4 bilhões de dólares corresponderam à venda de licença de software, enquanto os 9,9 bilhões de dólares foram relativos aos serviços relacionados.

O foco do resultado do projeto é atender principalmente as Pequenas e Médias Empresas (PME's) do setor de software do Brasil que desenvolvem produtos ou serviços associados. Sendo um pouco mais específico, são empresas que apresentam qualquer um dos diferentes tipos de modelos de negócio de software, conforme segue: (i) Desenvolvimento de serviço de levantamento e análise de requisitos e especificação da solução em um serviço de desenvolvimento sob encomenda; (ii) Desenvolvimento de serviço de programação (codificação); (iii) Software-produto (pacote); (iv) Software customizável - Adaptação e desenvolvimento sob medida de um software-produto (pacote); (v) Desenvolvimento de componentes de software; (vi) Desenvolvimento de Software embarcado.

6. Características Inovadoras

A maioria das ferramentas de avaliação existentes foca na automatização das atividades do modelo SPICE (ISO/IEC 15504), como exemplo destacamos a PISA, SEAL e a Appraisal Assistant. Este modelo é principalmente utilizado nos países da Europa. Porém, nas Américas e nos países do continente asiático o modelo mais utilizado para avaliação da capacidade de processos é o CMMI e no Brasil podemos destacar o uso também do MPS.BR. Em 2007, na Universidade Federal de Pernambuco (UFPE), foi proposta uma ferramenta (ProEvaluator) para apoiar as avaliações do modelo MPS.BR, mas que não suporta avaliações do CMMI. Dentre as ferramentas pesquisadas, a Appraisal Assistant do SQL é a única que suporta também avaliações do modelo CMMI (SCAMPI), mas não o MPS.BR. Sendo assim, esse projeto desenvolveu uma ferramenta com características técnicas e funcionais inovadoras para automatizar os processos de avaliação dos dois modelos de qualidade de software mais utilizados no Brasil, o CMMI e o MPS.BR. Mais especificamente, este projeto tem as seguintes inovações relacionadas à ferramenta: (i) Ampla automação dos métodos SCAMPI e MA-MPS, contemplando desde a fase de planejamento da avaliação, levantamento das evidências objetivas, identificação dos indicadores, registro de lacunas, pontos fortes e fracos, identificação das lacunas de implantação, até a geração dos resultados; (ii) Gestão de planos de ação que são criados para sanar as lacunas de implementação; (iii) Características técnicas de desenvolvimento avançadas. Sistema WEB de três camadas, multiusuário, e com portabilidade para a maioria dos bancos de dados utilizados atualmente; (iv) Cadastro das estruturas dos modelos com grande possibilidade de suporte a futuras versões dos mesmos.

7. Conclusão e Perspectivas Futuras

Esse projeto é fruto do trabalho de mestrado desenvolvido na Universidade Salvador – UNIFACS [SIQUARA, 2007] e foi aprovado em abril de 2008 no edital FAPESB/SECTI 008/2007 para desenvolvimento de soluções inovadoras no campo de Tecnologia da Informação e Comunicação (TIC) da Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB). Entre os benefícios esperados da sua aplicação em empresas desenvolvedoras de software, tem-se: (i) Facilitar a ocorrência de avaliações. Exemplo: melhoria interna dos processos, preparação para avaliação oficial, alcance de objetivos organizacionais e capacitação da equipe; (ii) Diminuição do tempo e redução dos custos de realização das avaliações através da agilidade na identificação e aprovação dos indicadores; (iii) Rastreabilidade das informações levantadas (evidências objetivas) e geração de bases históricas das avaliações realizadas; (iv) Definição clara do escopo da avaliação com identificação dos responsáveis resultando no comprometimento e maior integração da equipe; (v) Visão contínua da situação da avaliação e geração incremental do resultado da avaliação com base nos níveis de capacitação e maturidade.

O próximo passo desse trabalho será a execução de um novo projeto para a implantação em primeira instância da FAMP em empresas de software do Arranjo Produtivo Local (APL) da Bahia e posteriormente a nível nacional, com o objetivo de incentivar a melhoria dos processos de desenvolvimento de software brasileiro.

8. Referências

ABES – ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE. Mercado Brasileiro de Software: panorama e tendências 2010. São Paulo: Abes, 2010, p 24. Disponível em: <<http://www.abes.org.br/templ2.aspx?id=306&sub=306>>. Acesso em 08 jan. 2011.

ROCHA, A. R. C. da; MALDONADO, J. C.; WEBER, K. C. Qualidade de software: teoria e prática. São Paulo: Prentice Hall, 2001.

SIQUARA, Elson Cardoso. FAMP: uma ferramenta para apoiar avaliações CMMI baseadas no método SCAMPI. 2007. Dissertação (Mestrado)- Universidade Salvador (UNIFACS), Salvador, 2007.

SIQUARA, Elson Cardoso; SALVADOR, Lais do Nascimento. FAMP: uma ferramenta de suporte a avaliações CMMI baseadas no método SCAMPI. 2008. Simpósio Brasileiro de Engenharia de Software (SBES), Campinas, 2008.

SOFTTEX – ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO. Qualidade de software. Disponível em: <<http://www.softex.br/mpsbr>>. Acesso em: 29 nov. 2010.

SQI – SOFTWARE QUALITY INSTITUTE. Appraisal Assistant Beta. Queensland: Griffith University. Software Quality Institute. Disponível em: <<http://www.sqi.gu.edu.au/AppraisalAssistant>>. Acesso em: 03 jan. 2011.

XAVIER, Juliana Moura Cavalcanti. ProEvaluator: uma ferramenta para avaliação de processos de software. 2007. Dissertação (Mestrado)- Universidade Federal de Pernambuco (UFPE), Recife, 2007.

Projeto MDA4Eclipse.

Fabio P. Marzullo¹, Jano M. de Souza¹, Marcos M. Müller¹

¹ COPPE, Federal University of Rio de Janeiro, Technology Centre, Block H, Rio de Janeiro, Brazil

{fpm,jano,muller}@cos.ufrj.br

Abstract. *This paper presents results of the 2010 PBQP cycle for the shared development environment created with the purpose of using a domain-driven approach to shorten the development of software projects. Using a set of standards and theories such as BPM, MDA, SOA and Web Services, we propose a domain-driven development approach based on the “DEV 2.0” concept.*

Resumo. *Este trabalho apresenta resultados do ciclo 2010 do ambiente de compartilhamento e de reuso de modelos conceituais para o desenvolvimento de software. Tendo como objetivo o aumento da produtividade e a redução de custos em projetos de software e utilizando teorias como BPM, MDA, SOA e Web Services, propomos uma abordagem de desenvolvimento orientado a domínio pautada no conceito de “Dev 2.0”.*

1. Introdução

Trabalhos recentes demonstram como novas tecnologias e abordagens de desenvolvimento de software são fundamentais para a constante evolução de metodologias de gestão de projetos de software. Abordagens orientadas a domínio de negócio, por exemplo, estão entre as mais promissoras e têm apresentado bons resultados na condução e conclusão de projetos de software (Evans 2003).

Modelos conceituais são muito utilizados na representação e abstração de conceitos inerentes a diferentes domínios de negócios, e por meio destes, equipes de desenvolvimento são capazes de estruturar regras de negócio e representar elementos mais importantes dentro do domínio analisado.

Com base na teoria de Arquitetura Orientada por Modelos, doravante aqui denominada MDA (OMG 2010), surgiram diversos novos métodos de desenvolvimento que possibilitaram o aumento de produtividade e controle da qualidade sobre o software gerado. Combinando a estes conceitos e paradigmas outras tecnologias como Arquitetura Orientada por Serviços (doravante aqui denominada SOA, sigla em inglês), Business Process Management (doravante aqui denominada BPM, sigla em inglês) entre outras, já é possível garantir que produtos finais de software possuam a qualidade esperada pelo cliente final.

Logo, a principal conclusão deste ciclo foi a de saber como utilizar em conjunto todas as tecnologias descritas anteriormente para otimizar o desenvolvimento de software, tendo como resultado produtos com níveis bem definidos de qualidade.

Por exemplo, sempre que um projeto é iniciado, uma das primeiras ações que devem ser adotadas é a de estabelecer os parâmetros que orientarão o desenvolvimento, como a complexidade da solução técnica utilizada. Isso significa que é possível desenvolver um projeto ancorado por uma arquitetura com três camadas ao invés de utilizar cinco ou seis. Normalmente, o importante é saber identificar os “intermediários” e eliminá-los, sem que isso prejudique a qualidade do produto final. Essa atitude normalmente resulta em economia de tempo ou de dinheiro, e tanto uma quanto a outra me parecem muito boas.

Outra postura comum em projetos de software é a padronização das atividades de desenvolvimento. Isso não necessariamente significa utilizar padrões de projeto, mas abordagens padronizadas de desenvolvimento, as quais aumentem as chances de o produto final ter a qualidade esperada pelo usuário e respeitar prazos e custo de construção.

O fator predominante, portanto, foi a de entender como seria possível construir um ambiente que disponibilizasse serviços específicos para apoiar o desenvolvimento de software, e consequentemente, trazer mais eficiência no retorno do investimento em soluções de TI.

Finalmente nossa proposta foi a de criar uma infraestrutura, um conjunto de ferramentas, que pudesse ser utilizado por desenvolvedores e organizações em suas unidades de negócio como padrão de desenvolvimento. Dentro deste ambiente haveria uma ferramenta especial, um *framework* para geração de código, capaz de construir senão 100%, boa parte dos artefatos que compõem o software. Tendo a modelagem do domínio de negócio disponível no ambiente e a tecnologia para criar o produto de software, seria possível replicar boa parte do sistema, reduzindo proporcionalmente os custos do projeto.

Desta abordagem surgiria a seguinte expressão:

Modelagem + Automação = Economia!

A utilização desse tipo de tecnologia poderia dar à organização a facilidade de criar uma abordagem de desenvolvimento repetível e previsível, permitindo que todas as suas soluções tivessem a mesma arquitetura, os mesmos componentes, as mesmas camadas etc. O desenvolvimento poderia ser feito dentro de um

processo automatizado que a partir do modelo de domínio, fosse capaz de criar os artefatos de software necessários à solução.

Tendo-se uma abordagem única de desenvolvimento (nela incluída ferramentas de automação e geração de código) e um processo de modelagem de domínio padronizado e reutilizável, por que não compartilhá-lo e dar a possibilidade à outros órgãos iniciarem seu desenvolvimento alguns passos à frente? Não seria possível criar uma rede de compartilhamento e permitir tal ação em tempo real? Não seria possível criar um conjunto de Web Services que, quando orquestrados, pusessem esse cenário em prática? A resposta é sim, cada órgão seria obrigado a cadastrar todos os seus modelos de domínio em uma base central, então este ficaria disponível para qualquer outro que desejasse iniciar um desenvolvimento com as mesmas características.

2. Objetivos e Justificativa

Conforme apresentado em agosto de 2010 no EQPS Brasília, definimos algumas atividades para o ciclo, a saber:

i. Mapeamento dos serviços e componentes:

Este objetivo determinava a melhoria do mapeamento dos serviços e dos componentes de software necessários a correlação direta ou indireta com as regras de negócio do domínio em questão e finalizar a integração da modelagem dos processos do domínio usando a teoria de BPM.

Resultado: Com o auxílio desta teoria, subdividimos os processos em serviços e componentes de software, e geramos os arquivos denominados "process definition files", a saber:

```
<?xml version="1.0" encoding="UTF-8"?>
<process-definition xmlns="urn:jbpml-3.2" name="simple">
  <start-state name="start">
    <transition name="to_authentication" to="Authenticate"/>
  </start-state>
  <state name="Authenticate">
    <transition to="Order Request" name="to_order_request"/>
  </state>
  <state name="Order Request">
    <transition to="end" name="to_end"/>
  </state>
  <end-state name="end"/>
</process-definition>
```

Trecho de Código 1. O Arquivo *processdefinition.xml* representa o modelo de domínio.

O arquivo de definição do processo agrega informações a respeito do domínio. Cada estado é usado para representar um serviço responsável por executar as ações necessárias dentro do processo. Especificamente, cada ação deve apontar para um arquivo XML contendo a modelagem do serviço ou componente de software.

ii. Melhorias no módulo de persistência:

Este objetivo determinava a melhoria do módulo de persistência do ambiente, sendo a publicação, recuperação e utilização dos modelos de negócios.

Resultado: Cada sítio de projeto persiste todos os seus modelos desenvolvidos e disponibiliza via Web-Service o acesso a estes artefatos.

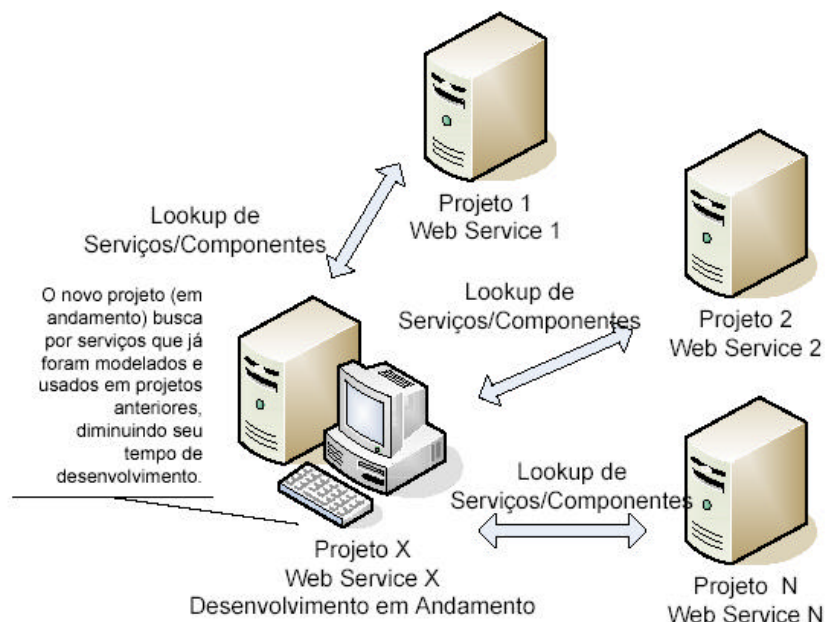


Figura 1. Processo de Compartilhamento de Modelos de Domínios: Os arquitetos responsáveis pelo desenvolvimento do sistema realizam uma busca em projetos vizinhos e tentam encontrar modelos pré-construídos, de maneira a facilitar a geração inicial de código.

O compartilhamento consiste em orquestrar um conjunto de Web Services públicos (isto é, acessíveis dentro de diferentes setores da organização, ou entre organizações), com o propósito de permitir o reuso de modelos de domínio já construídos. O Web Service publica o modelo (basicamente o arquivo de definição de processo e os demais arquivos XML) pelo arquivo WSDL que é hospedado em um servidor de aplicação, como o JBoss (JBoss 2010).

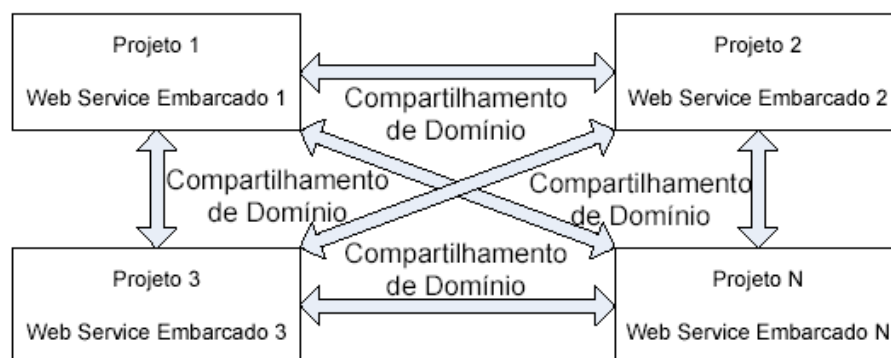


Figura 2. Arquitetura para Compartilhamento de Domínios.

Em um nível mais técnico, o compartilhamento ocorre por meio da utilização de serviços de catalogação, busca e recuperação. Cada organização disponibiliza tais serviços em um ambiente web, contendo um servidor de aplicação executando o(s) web service(s) e acessando uma base de dados local.

iii. Melhorias no núcleo de geração de código:

O MDA4Eclipse (www.mda4eclipse.com.br) foi criado com o objetivo de ser um novo ambiente de desenvolvimento baseado no uso de modelos como principal fonte de geração de código. Aprimorar a forma como os modelos de domínio são lidos, permitiria que uma maior quantidade de código pudesse ser gerada, dessa forma reduzindo ainda mais os custos relacionados ao desenvolvimento de software e permitindo que o desenvolvimento seja focado nas regras de negócio da aplicação.

Resultado: além da conclusão do cartucho básico, foi acoplado um novo cartucho capaz de produzir uma arquitetura mais simples, mas robusta e com maior capacidade de absorção de mudanças.

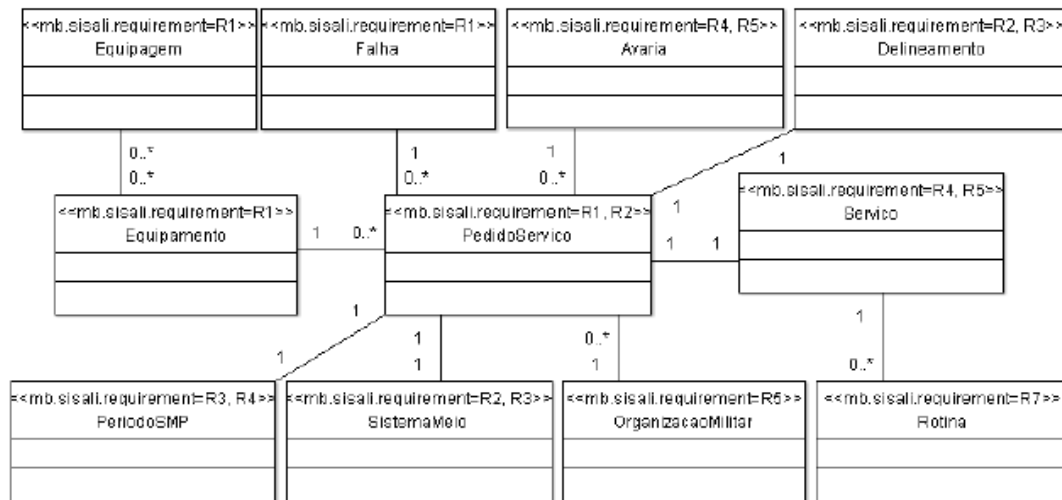


Figura 3. Exemplo de modelagem de um serviço.

iv. Criação de mecanismo de versionamento e melhoria da segurança É muito importante poder armazenar ao longo do tempo a evolução dos artefatos utilizados em projetos, bem como poder acessar a qualquer momento suas versões.

Logo, este objetivo determinava a criação de um mecanismo de versionamento e melhoraria dos padrões de segurança.

Resultado 1: o mecanismo de versionamento dos modelos foi implementado a partir de um autorrelacionamento na tabela referente a estes documentos, para assim poder persistir as diversas versões dos artefatos produzidos.

Resultado 2: os critérios de segurança adotados foram a inclusão de uma tela de autenticação e o uso de criptografia na troca de informações sigilosas.

3. Metodologia de Execução

No ciclo em questão, os objetivos propostos foram finalizados dentro do seguinte cenário: do ponto de vista da abordagem do problema foi aplicado um modelo simplificado de pesquisa operacional que considerou testes empíricos das proposições definidas anteriormente e na conseqüente obtenção dos resultados.

4. Resultados Obtidos

Abaixo seguem listados os resultados obtidos no ciclo 2010:

i. Produtos de software gerados (módulos ou programas de computador resultantes do projeto, disponibilizados para o mercado):

Apesar de ainda estar em constante evolução, já é possível utilizar o MDA4Eclipse dentro de um modelo de contratação de serviço de consultoria, no entanto estamos trabalhando para disponibilizar uma nova versão simplificada para download no site do projeto.

ii. Processos descritos e institucionalizados que contribuem para a melhoria da qualidade de software:

Em nossa metodologia de desenvolvimento, todos os sistemas são gerados baseados em uma arquitetura orientada por padrões de mercado, de simples evolução, de fácil entendimento e eficiente no que toca a construção do produto. Tal arquitetura vem sendo utilizada há mais de cinco anos, recebendo atualizações e melhorias, todas sustentadas por medições utilizando métricas de software.

A abordagem de compartilhamento de modelos possibilita um aumento de produtividade, uma vez que módulos prontos de funcionalidades desenvolvidas anteriormente podem ser prontamente reutilizados pelos arquitetos. Por fim, o módulo de análise de impacto do MDA4Eclipse ajuda a manter a qualidade do código gerado, pois baseado nos requisitos do sistema, os gerentes podem verificar, antes de proceder com uma alteração, quais os módulos e componentes do sistema que serão afetados por esta modificação.

iii. Artigos publicados:

- MARZULLO, F. P. ; SOUZA, J. M. A Case Study on Software as Service Approach to Model-Driven Development. Submitted to: The 13th International Conference on Model Driven Engineering Languages and Systems Oslo, Norway, 2010.

- MARZULLO, F. P. ; SOUZA, J. M. . An MDA Approach to Change Impact Analysis. Submitted to: The 10th International Conference on Web Engineering, Vienna, Austria, 2010.
- MARZULLO, F. P. ; SOUZA, J. M. . A Cost Model For Domain-Driven Development Using BPM, MDA, SOA And Web Services. In: IADIS Internacional Conference WWW/INTERNET 2009, 2009, Roma.

iv. Dissertações e/ou teses geradas:

- Uma tese de doutorado apresentando um modelo de negócios para utilização do ambiente de desenvolvimento em escala global e, como subproduto, a criação de um serviço de desenvolvimento de software em nuvem
- Duas teses de mestrado com o objetivo de evoluir a infraestrutura tecnológica e a garantia da qualidade.

v. Eventuais parcerias ou programas de transferência de tecnologia efetuados:

O projeto MDA4Eclipse em conjunto com os portais www.domainshare.com.br e www.projectbidding.com.br estão sendo desenvolvidos como projetos de pesquisa pela COPPE/UFRJ em parceria com as empresas MZ – Empresarial e Sinn Tecnologia.

5. Aplicabilidade dos Resultados

Ao longo deste ciclo, observou-se que a utilização do ambiente é capaz de reduzir consideravelmente custos e tempo de desenvolvimento, sem que isso comprometa a qualidade do produto gerado. O ambiente foi utilizado em 2 projetos, obtendo uma economia média de 39% em termos de artefatos de software desenvolvidos automaticamente.

Com esses resultados, espera-se em médio prazo (e gradativamente) aumentar os níveis de reuso e geração de código, elevando o projeto como referência nacional no desenvolvimento utilizando tecnologias como MDA e SOA.

6. Características Inovadoras

O MDA4Eclipse é um projeto pioneiro no território nacional na criação de um ambiente de desenvolvimento orientado a domínio de negócios. Por conceber um modelo de software em nuvem para disponibilização de um serviço de desenvolvimento seu modelo de negócio pode ser replicado em organizações públicas ou privadas. A inovação está associada ao fato de, atualmente, soluções de software com o perfil aqui apresentado ainda estarem em seus primeiros estágios de pesquisa, e em contrapartida, já possuímos resultados que nos dão certo nível de maturidade.

Finalmente, o produto final, em médio prazo, é o serviço de desenvolvimento de software, sendo este apoiado por toda a infraestrutura aqui apresentada. Algumas ações já estão em andamento para a criação do grande portal de compartilhamento de domínios de negócio, onde o usuário poderá consultar domínios de projetos já cadastrados, e efetuar o download de aplicações previamente desenvolvidas, bastando apenas incorporar suas regras de negócio.

7. Conclusão e Perspectivas Futuras

Por meio da integração de técnicas como MDA, SOA e Web Services, foi possível criar um ambiente de desenvolvimento capaz de utilizar modelos conceituais como input para geração de sistemas e compartilhá-los com outros locais de projetos.

Os resultados estão em sua fase final de análise, mas claramente é possível verificar que a união de tais tecnologias, quando bem feita, pode oferecer ganhos consideráveis de prazo e até mesmo de orçamento no processo de desenvolvimento de software.

Finalmente, pretendemos implantar em um próximo ciclo, um sistema de leilão on-line de artefatos de projeto, apoiado pelo serviço de desenvolvimento de software, onde usuários poderão colaborar com o desenvolvimento de aplicações, seguindo os padrões da arquitetura de desenvolvimento deste projeto, bem como criar um ambiente completo de desenvolvimento baseado no MDA4Eclipse, totalmente hospedado na web.

8. Referências Bibliográficas

1. EVANS, E., "Domain-Driven Design: Tackling Complexity in the Heart of Software", Addison Wesley, 2003.
2. MDA, Model Driven Architecture, <http://www.omg.org/mda>. Último acesso em outubro 2010.
3. MDA4eclipse Research Project, www.mda4eclipse.com.br. Último acesso dezembro 2010.

4. Eclipse Project. <http://www.eclipse.org>. Último acesso em dezembro 2010.
5. Business Process Management Initiative, <http://www.bpmi.org/>, 2008.
6. JBoss Application Server, <http://www.jboss.org/>. Último acesso em abril 2010.
7. MCGOVERN, J., Sims, O., Jain, A., Little, M., Enterprise Service Oriented Architectures - Concepts, Challenges, Recommendations, Springer, 2006.
8. ROGER, S. P. Software Engineering - A practitioner's Approach, McGraw-Hill, 2001.

Institucionalização de um Processo Padrão de Desenvolvimento de Software

Amélia de Lima Guedes¹, João Carlos Testi Ferreira², Anderson Ricardo Frezza³, Sidinei José Rossoni⁴

¹Mestre em Gestão do Conhecimento e TI e Especialista da Caixa Econômica Federal

²Especialista em Melhoria de Processo de Software e Analista da Caixa Econômica Federal

³Especialista em Engenharia de Software e Consultor da Caixa Econômica Federal

⁴Especialista em Engenharia de Software e Gerente Executivo da Caixa Econômica Federal

amelia.guedes@caixa.gov.br, joão-carlos.ferreira@caixa.gov.br, anderson.frezza@caixa.gov.br, sidinei.rossoni@caixa.gov.br

Abstract. *This paper describes the work of defining a standard process for developing systems in a large organization, showing difficulties and strategies for the building of a flexible process in the adoption of developments methodologies according to the specific needs of projects, but ensuring institutional control and management on their knowledge generated.*

Resumo. *Este relato descreve o trabalho de definição de um processo padrão de desenvolvimento de sistemas em uma grande organização, apresentando dificuldades e estratégias para a construção de um processo flexível na adoção de metodologias de desenvolvimento adequadas às necessidades específicas dos projetos, mas que garantisse o controle institucional e a gestão sobre o conhecimento gerado.*

1. Introdução

A adequação ao mercado e o desafio competitivo exigem que as organizações automatizem seus processos e ofereçam serviços atrativos aos clientes. Em especial, organizações financeiras têm investido muito em tecnologia da informação, oferecendo serviços automatizados e se adequando a normas reguladoras rígidas e de segurança.

Muitas vezes as organizações financeiras precisam desenvolver seu próprio software por não encontrarem no mercado solução que as atenda adequadamente. Isso exige uma área de desenvolvimento que esteja alinhada com as necessidades e estratégia da empresa e possa atender aos requisitos de segurança organizacional.

Não se pode pensar em automatizar um processo ficando refém da aplicação, que deve ser construída de forma que permita ajustes e adaptações para adequar-se às mudanças necessárias impostas pelo mercado e órgãos reguladores no tempo certo.

Assim, é importante que seja estabelecido um processo de desenvolvimento que possa atender a tais necessidades com a confiabilidade necessária. Isso exige atenção quanto aos controles e à gestão do conhecimento aplicados ao processo de desenvolvimento.

Não é tarefa fácil estabelecer um processo que atenda a todo tipo de projeto da organização.

“A dificuldade em definir processos encontra-se na ausência de um processo de software possível de ser genericamente aplicado. Os processos variam porque são diferentes os tipos de sistemas, os domínios de aplicação, as equipes, as organizações e as próprias restrições de negócio, tais como, cronograma, custo, qualidade e confiabilidade” (Machado, 2000).

Como o Guia PMBOK (2003, p. 5) define, um projeto cria um produto ou serviço único, e para tal feito é difícil estabelecer um processo ou uma metodologia que seja capaz de atender qualquer projeto devido a suas especificidades.

Segundo Pressman (2006, p. 4), o desenvolvimento de software não é uma esteira de produção, passa por processo de engenharia. A Engenharia de Software segundo Fritz Bauer “é a criação e a utilização de sólidos princípios de engenharia a fim de obter softwares econômicos que sejam confiáveis e que trabalhem eficientemente em máquinas reais” (PRESSMAN, 2006). À medida que a nossa capacidade de produzir software aumentou, também cresceu a complexidade dos sistemas requeridos.

Novas tecnologias que resultam da convergência de sistema de computadores e de comunicação trazem novas questões para os engenheiros de software e complexidade para a Engenharia de Software.

Tal ciência tem nos últimos anos evoluído muito. Seu domínio foi subdividido em diversas áreas de conhecimento como (SWEBOK, 2004): [Requisitos de Software](#); [Projeto de Software](#); [Construção de Software](#); [Teste de Software](#); [Manutenção de Software](#); [Gerência de Configuração de Software](#); [Gerência de Engenharia de Software](#); [Processos de Engenharia de Software](#); [Ferramentas e Métodos de Engenharia de Software](#); e [Qualidade de Software](#).

No caminho inverso das exigências de cada uma das áreas de conhecimento está a urgência por sistemas de informação, com prazos de entrega cada vez menores e até imediatos, associados ao *timing* do negócio. Com o propósito de atender a essas expectativas, surgiram diversas técnicas procurando buscar agilidade no desenvolvimento e manutenção de sistemas: Lean, Scrum, XP (*Extreme Programming*), TDD (*Test Driven Development*) entre outras. Tais técnicas valorizam, através de princípios de gestão do conhecimento, fortalecimento técnico da equipe, autosuficiência e motivação, com foco no produto (o código).

A urgência proposta de aplicação de técnicas ágeis confronta com os inúmeros artefatos e controles impostos pelas melhores práticas associadas a cada uma das áreas de conhecimento da Engenharia de Software.

Assim, considerando este paradoxo, há um grande desafio em definir um processo a ser seguido institucionalmente e em se definir a informação mínima necessária para dar suporte ao desenvolvimento e a manutenção dos sistemas informatizados de maneira institucional.

O trabalho foi realizado em uma das maiores instituições financeiras do Brasil, que possui mais de 1500 empregados na área de tecnologia, distribuídos em 3 grandes *sites* de desenvolvimento, dedicando-se principalmente a sistemas média e altamente críticos. Além desses *sites*, outras sete áreas regionais estavam iniciando suas atividades como uma alternativa para o desenvolvimento e dedicadas principalmente ao desenvolvimento de sistemas de não críticos. Essa instituição tem alto grau de automação e inovação e seus negócios dependem fortemente de mais de 300 sistemas de informação. Além disso, contrata a prestação de serviços técnicos especializados em desenvolvimento/manutenção de sistemas e aplicativos de sete fábricas de software fortemente especializadas, das quais se espera, dessa forma, processos otimizados e atualizados.

Este artigo apresenta a solução adotada por essa instituição para estabelecer um processo de desenvolvimento que fosse padronizado, permitindo um melhor controle, estabilidade no desenvolvimento de software, clareza e gestão do conhecimento envolvido, sem tirar a flexibilidade necessária para o atendimento das necessidades peculiares de cada projeto. Esse processo foi chamado de Processo Padrão de Desenvolvimento de Sistemas - PPDS.

2. Objetivos e Justificativa

O Processo Padrão de Desenvolvimento de Sistemas teve os seguintes propósitos:

- ✍ definir um padrão de processo aplicável a qualquer projeto de desenvolvimento de software;
- ✍ definir de forma clara controles institucionais únicos aplicáveis a todos os projetos de desenvolvimento de sistemas;
- ✍ definir cenários customizáveis a serem adotados para o desenvolvimento de sistemas;
- ✍ unificar normas do processo de desenvolvimento em um único instrumento normativo;
- ✍ viabilizar a conformidade no desenvolvimento de sistemas;
- ✍ simplificar e facilitar o desenvolvimento de sistemas;
- ✍ possibilitar a flexibilidade e a inovação nos processos de desenvolvimento de software, de forma a garantir o registro do conhecimento dentro de um padrão institucional.

A definição do Processo Padrão de Desenvolvimento veio modificar totalmente uma realidade em que diversos normativos orientavam o desenvolvimento de sistemas direcionados à técnica a ser utilizada. Tais normativos eram muito detalhados e as atividades previstas eram sempre exigidas, independentemente de estarem aderentes às necessidades do projeto. Além disso, algumas técnicas praticadas não eram atendidas pelos normativos. O processo a ser seguido não era claro. As equipes raramente podiam inovar em processos e os projetos, mesmo com necessidades peculiares, eram tratados igualmente. Não havia uma trilha de desenvolvimento, mas um trilho, que obrigatoriamente deveria ser seguido.

Assim, havia a necessidade de se definir um processo único que pudesse ser plenamente utilizado, mas que tivesse como premissa, além da conformidade na adoção dos controles institucionais, a agilidade no desenvolvimento de novas soluções, a segurança, o domínio do conhecimento e a flexibilidade, na possibilidade de customização de acordo com as necessidades do projeto. O PPDS veio trazer ao processo de desenvolvimento clareza, conformidade, transparência, possibilidade de inovação e gestão do conhecimento. Independentemente da técnica adotada o objetivo dos controles são os mesmos, validando documentos, definindo acordos entre áreas distintas e garantindo o registro e a disseminação do conhecimento.

A necessidade de um maior controle sobre os projetos em andamento ocorria tanto por razões de negócio, quanto por questões legais e normativas. Associava-se a isso um complexo modelo de terceirização, em que, a fim de se utilizar a expertise das fábricas especializadas e, em vez de se definir o processo a ser utilizado, era necessário estabelecer quais artefatos deveriam ser exigidos nos projetos e com que padrão.

Santos *et al* (2008, p. 208) destaca a necessidade da definição de regras e requisitos mínimos de processos em organizações, definição de processos-padrão e suas regras de adaptação, devido à diversidade de

processos que podem nela existir. Há que se observar, no entanto, que tais artefatos teriam que acontecer independente das metodologias adotadas nos projetos.

3. Metodologia de Execução

O trabalho que foi coordenado pela área responsável pela definição da metodologia de desenvolvimento, contou com representantes das áreas de desenvolvimento e qualidade das três representações de desenvolvimento (RJ, SP e DF) e da coordenação de desenvolvimento descentralizado, além de um representante da área de gestão de implementação de TI.

Esse grupo efetuou o descolamento do processo de desenvolvimento de software em 3 níveis de processo: o primeiro, corporativo, voltado aos controles institucionais a serem preservados e à gestão do conhecimento; o segundo, voltado à especialização técnica de acordo com o paradigma a ser adotado de acordo com as características do projeto e a preferência da equipe técnica envolvida; e o terceiro nível, instanciado a partir das necessidades peculiares de cada projeto específico.

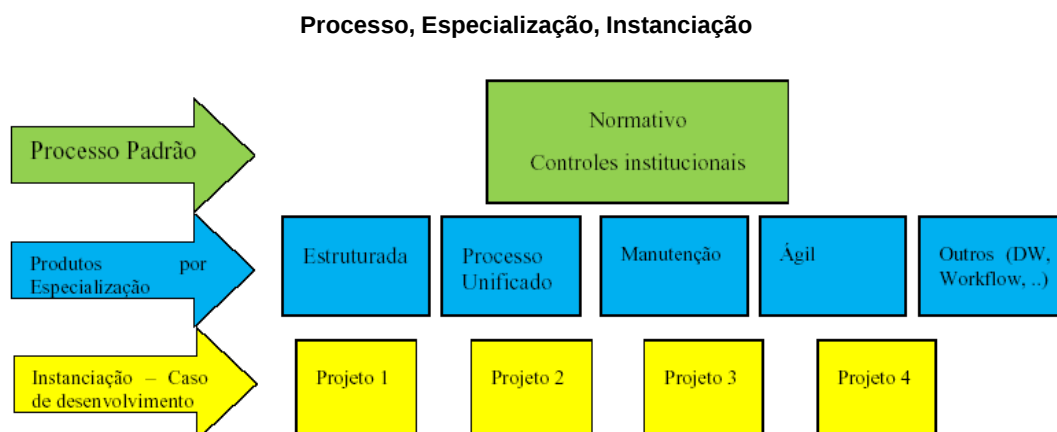


Figura 1: Camadas do Processo de Desenvolvimento

Os controles institucionais considerados essenciais e exigíveis a qualquer tipo de projeto de desenvolvimento ou manutenção de software foram identificados e passaram a compor o PROCESSO PADRÃO, definidos em único normativo sobre o desenvolvimento de software. Os demais normativos sobre o assunto foram revogados.

As definições das camadas do processo foram materializadas em um *site* com o uso de uma ferramenta de modelagem de processo, o *Rational Method Composer* (RMC).

Em uma segunda camada, dentro do RMC, foram criados cenários de acordo com a característica técnica ou paradigma a ser seguido pelo projeto. Houve a criação de cenários para Análise Estruturada, Processo Unificado, Desenvolvimento Descentralizado – Processo Unificado Simplificado, Construção de Portais *WEB*, Técnicas Ágeis (SCRUM, XP, TDD, LEAN), além de um cenário para projetos de Manutenção.

Os projetos são instanciados em uma terceira camada, em que, a critério do gerente de projetos e sua equipe, o caso de desenvolvimento a ser seguido é definido, através da escolha do cenário e ciclo de vida a ser adotado e da complementação dos artefatos a serem gerados, além daqueles obrigatórios por materializarem controles institucionais.

O *Site* do PPDS foi disponibilizado para todas as áreas de desenvolvimento, tanto no ambiente centralizado (RJ, SP e DF) como no descentralizado (distribuído em sete regionais), e, também, para as fábricas de software contratadas, inseridas nesse ambiente de desenvolvimento institucional.

4. Resultados Obtidos

O PPDS possibilitou a adoção do processo padrão pelas áreas de desenvolvimento em todos os projetos. Garantiu o atendimento às necessidades de customização de acordo com as necessidades específicas de cada projeto. Viabilizou a melhoria na agilidade, conformidade e inovação. Além disso, permitiu, com a utilização de um portal de desenvolvimento de software, que os controles institucionais definidos no processo e seus cenários alternativos fossem utilizados não só nas unidades de desenvolvimento como nos parceiros instalados fora do ambiente físico da instituição.

O novo processo de desenvolvimento de software adotado, PPDS, permitiu o melhor uso das competências disponíveis tanto internamente, na organização, como externamente, nas fábricas de software. Possibilitou, também, a inovação no desenvolvimento de software, sem abrir mão dos controles institucionais que irão garantir a continuidade dos processos de evolução do software e a gestão do conhecimento. Essa nova abordagem permite a otimização do processo de desenvolvimento de acordo com a necessidade do projeto e o

conhecimento técnico disponível e dá mais agilidade ao desenvolvimento de software possibilitando um melhor atendimento às necessidades das áreas de negócio da organização.

O PPDS permitiu que a área de Tecnologia tivesse maior eficiência na entrega dos produtos, economizando recursos e o tempo gasto na confecção de artefatos cobrados por mero formalismo nos processos anteriormente utilizados, havendo um ganho médio em torno de 45%.

5. Aplicabilidade de Resultados

Os resultados esperados para o PPDS foram plenamente alcançados. Os controles institucionais foram definidos, o *site* foi construído, os cenários previstos inicialmente foram elaborados. Todos os projetos iniciados a partir de janeiro de 2010 já se encontram aderentes ao PPDS.

Houve uma melhora substancial na conformidade dos projetos, os doze controles institucionais são conhecidos e possíveis de serem cumpridos. Os gerentes de projetos e as equipes de desenvolvimento podem utilizar todo o conhecimento disponível para traçar o melhor caminho a ser trilhado para o projeto – essa definição torna-se transparente no caso de desenvolvimento, que é um dos controles obrigatórios. Além dos controles institucionais, as equipes só desenvolvem aqueles artefatos que agregam valor ao projeto.

6. Características Inovadoras

A idéia de se adotar um processo padrão de desenvolvimento se contrapõe à necessidade de tratamento único e peculiar que cada projeto, por isso é comum ver-se nas instituições essa dificuldade. O descolamento de uma camada de controles institucionais do processo de desenvolvimento possibilitou a construção de um padrão aderente a qualquer processo, sem tolher o atendimento às características peculiares de cada projeto através de customizações. Isso, apesar de simples é inédito.

O registro em um único site dos controles institucionais, de todas as orientações e melhores práticas para o desenvolvimento de sistemas dá transparência ao processo de desenvolvimento e promove a gestão do conhecimento. Além disso, ter um processo padronizado facilita os ajustes em busca de melhorias e viabiliza ações de automação desses processos.

7. Conclusão e Perspectivas Futuras

Este trabalho mostrou ser viável a construção de um modelo de processo para desenvolvimento de sistemas a ser genericamente aplicado. A definição de um processo de desenvolvimento de sistemas corporativo é fundamental, mas sempre se contrapõe à importância de se dar tratamento único às necessidades diferenciadas dos projetos.

Aliado a esse paradoxo, há os diferentes níveis de maturidades das equipes de desenvolvimento e, ainda, os valores técnico-culturais das organizações que compõem o ambiente corporativo.

A construção de um processo padrão de desenvolvimento de sistemas adotado em toda corporação mostrou ser possível descolar a camada de controles baseados em definições organizacionais e gestão do conhecimento daquelas definições especializadas, relacionadas às técnicas, tecnologias e paradigmas de desenvolvimento.

Essas, por sua vez, também podem ser descoladas dos modelos específicos a serem aplicados em função das necessidades específicas instanciadas a cada projeto.

Este modelo em 3 camadas de níveis de processo trouxe clareza, conformidade, transparência e gestão do conhecimento ao processo de desenvolvimento.

Embora o projeto PPDS tenha sido finalizado, com a entrega do Cenário Ágil, em outubro de 2010, o trabalho relacionado ao PPDS deve continuar, através do processo de melhoria contínua vinculado aos comitês do MPS – Programa de Melhoria do Processo de Software¹⁶. Há ainda outros cenários que deverão ser construídos, à medida que novas técnicas ou tecnologias assim o exigirem. Já se encontra em andamento, em um dos comitês do MPS, responsável pela disciplina de Ambiente, a construção de cenário para *DataWarehouse* e, em breve, iniciaremos a construção do cenário para *Workflow*.

O acompanhamento, através dos comitês de disciplinas do MPS, levará o processo paulatinamente a um melhor nível de maturidade.

A padronização do processo e a utilização de ferramentas de modelagem de processo permitiram que se buscasse a automação do processo de desenvolvimento e este é o próximo desafio, que já estamos enfrentando.

8. Referências Bibliográficas

¹⁶ 1 Programa institucional de qualidade, responsável por promover a melhoria do processo de engenharia de software, por meio da adoção das melhores práticas e padrões de mercado, para otimizar a construção de aplicativos.

Alain Abran, James W. Moore, Pierre Bourque, Robert Dupuis, Leonard L. Tripp (2004). "[Guide to the Software Engineering Body of Knowledge - SWEBOK](#)". [2004 Version](#) IEEE, 2004

Machado, L. F. (2000) "Modelo para Definição de Processos de Software na Estação Taba". Orientadora Ana Regina Cavalcanti Rocha, Tese de Mestrado, COPPE/UFRJ.

Project Management Institute. (2004) "Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos". 3. ed. PMI – *Project Management Institut*, Inc. Pennsylvania.

Pressman, Roger S. (2006) "Engenharia de Software". 6. ed. São Paulo: McGraw-Hill.

Santos, Gleison; Rocha, Ana Regina; Travassos, Guilherme Horta. (2007) "Ambientes de Engenharia de Software Orientados a Corporação – Uma infraestrutura Computacional para a Definição, Execução e Melhoria de Processos de Software em Corporações" COPPE/UFRJ, In Programa Brasileiro de Qualidade e Produtividade de Software.

Spider-PM: Ferramenta de Software Livre de Apoio à Modelagem de Processos de Software

Renan Sales Barros¹, Sandro Ronaldo Bezerra Oliveira¹

¹Faculdade de Computação – Universidade Federal do Pará (UFPA) Rua Augusto Corrêa, 1 -
Guamá - 66075-110 - Belém - PA - Brasil

renansalesbarros@gmail.com, srbo@ufpa.br

Abstract. *This paper makes a presentation of the development project of the process modeling tool called Spider-PM. This paper will show the objectives, justification, methodology and results of this project. In this one, the main innovations of this project are also shown and how these innovations can be used to improve the actual software development scene.*

Resumo. *Este artigo faz uma apresentação do projeto de desenvolvimento da ferramenta de modelagem de processos denominada Spider-PM. Nesse artigo serão mostrados os objetivos, justificativa, metodologia e resultados desse projeto. Também serão mostradas as principais inovações desse projeto e como essas inovações podem ser usadas de modo a melhorar o cenário de desenvolvimento de software atual.*

1. Introdução

Um processo de software pode ser definido como o conjunto de artefatos, recursos humanos, recursos computacionais, estruturas e restrições organizacionais que se relaciona com um conjunto parcialmente ordenado de etapas que visa à produção e manutenção de um produto de software [Lonchamp, 1993]. Dessa forma, percebe-se que o processo de software apresenta uma grande quantidade de elementos que podem influenciar no seu desempenho. Os processos de software são geralmente complexos e imprevisíveis, uma vez que eles dependem muito de pessoas e circunstâncias.

Diferentemente de outros processos de produção, existem atividades em um processo de software que não podem ser automatizadas e são fortemente dependentes da comunicação, cooperação e coordenação entre as partes integrantes desse processo [Bendraou *et al*, 2010].

Assim, é um desafio para as organizações desenvolvedoras de software encontrar uma maneira efetiva para descrever e gerenciar as características de um processo de software. Uma das maneiras efetivas de se descrever um processo de software consiste na modelagem do mesmo. Um modelo de processo de software é a representação dos aspectos mais relevantes de um processo de software do mundo real [Oliveira, 2006]. A modelagem de processos provê um mecanismo para o registro, entendimento, avaliação e comunicação de um processo de software e de suas melhorias [Kellner; Hansen, 1988].

Existem diversas ferramentas e linguagens para modelagem de processos de software, no entanto, essas ferramentas apresentam características que dificultam sua utilização pela indústria de software como, por exemplo, serem proprietárias ou serem soluções com ênfase na formalidade [Bendraou *et al*, 2010]. Assim, esse artigo apresenta um projeto relacionado com o desenvolvimento de uma ferramenta denominada Spider-PM que foi concebida com foco nas principais necessidades da indústria de software referentes à modelagem de processos de software.

Dessa forma, além desta seção introdutória, a seção 2 desse artigo apresenta a justificativa para o desenvolvimento desse projeto juntamente com os seus objetivos. A seção 3 expõe a metodologia utilizada durante a execução desse projeto. A seção 4 lista os principais resultados obtidos a partir desse projeto. Em seguida, a seção 5 discute a utilização dos resultados obtidos e qual a relevância desses resultados. A seção 6, por sua vez, relata as características inovadoras do projeto. Por fim, a seção 7 discute as principais conclusões obtidas com esse projeto e as perspectivas futuras do mesmo.

2. Objetivos e Justificativa

O principal objetivo desse projeto consiste no desenvolvimento da ferramenta Spider- PM. Essa ferramenta foi desenvolvida para apoiar a modelagem de processos de software suportando algumas das principais práticas adotadas pela indústria de software, como, por exemplo, a utilização de tecnologias gratuitas e de linguagens baseadas na *Unified Modeling Language* (UML) [Bendraou *et al*, 2010].

A partir de observações realizadas durante a implementação de programas de melhoria do processo de software [Barros; Oliveira, 2010a], constatou-se que muitas organizações utilizam as notações do *Software & Systems Process Engineering Meta- Model* (SPEM). O SPEM foi criado em 2002 e é o padrão para modelagem de processos mantido pela *Object Management Group* (OMG). A OMG é a organização que mantém os padrões para as aplicações orientadas a objetos. O SPEM define um conjunto de estereótipos específicos para modelagem de processos que devem ser utilizados em conjunto com as demais notações da UML [OMG, 2008]. Dessa forma, muitas organizações utilizam o SPEM pelo fato dele ser o padrão da OMG e pela

vantagem dele reutilizar as notações da UML, notações essas que são bastante familiares aos profissionais envolvidos com o processo de software.

Por outro lado, a especificação do 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 do SPEM. Assim, para diminuir o esforço requerido para modelagem de processos, a linguagem para modelagem utilizada pela ferramenta Spider-PM, a SPIDER_ML, incorpora algumas características desses perfis, ou seja, a SPIDER_ML é uma linguagem para modelagem definida a partir das notações do SPEM e, conseqüentemente, da UML.

Dessa forma, o desenvolvimento da ferramenta Spider-PM assim como o desenvolvimento da linguagem SPIDER_ML se justificam pela ausência de uma ferramenta e uma linguagem para modelagem de processos de software de acordo com práticas de modelagem utilizadas pela indústria de software e identificadas a partir das pesquisas realizadas no decorrer da execução desse projeto. Além disso, ao facilitar a modelagem de processos, o desenvolvimento da Spider-PM também se justifica por facilitar o atendimento dos principais objetivos da modelagem de processos que consistem em prover mecanismos para a comunicação, entendimento, gerenciamento, avaliação e aperfeiçoamento do processo de software [Kellner; Hansen, 1988].

3. Metodologia de Execução

Nas primeiras fases desse projeto, a metodologia adotada consistiu na realização de pesquisas bibliográficas, onde se buscava deduzir os requisitos funcionais e nãofuncionais da ferramenta Spider-PM. Essas pesquisas apresentavam caráter descritivo, uma vez que visavam relatar os principais aspectos relacionados com modelagem de processos de software. Além disso, também apresentavam um caráter exploratório, pois foram realizadas com base em estudos de referências bibliográficas e em observações realizadas durante a implementação de programas de melhoria de processos organizacionais. Nessas observações, buscou-se identificar as práticas adotadas na indústria de software para modelagem de processos.

A linguagem SPIDER_ML foi elaborada a partir dessas pesquisas realizadas nas fases iniciais desse projeto. No decorrer da fase de implementação da ferramenta Spider-PM, eram definidos ciclos de desenvolvimento semanais. No início de cada um desses ciclos, era definido um subconjunto de funcionalidades que devia ser implementado durante o ciclo em questão. O conjunto total de funcionalidades que devia estar presente na Spider-PM também foi definido a partir das pesquisas realizadas nas fases iniciais desse projeto.

4. Resultados Obtidos

O principal resultado do projeto apresentado nesse artigo é a ferramenta de software livre Spider-PM. Essa ferramenta se encontra disponível na página do Projeto SPIDER (<http://www.spider.ufpa.br> acessando a guia "Downloads"). Como mencionado, uma linguagem de modelagem denominada SPIDER_ML também foi desenvolvida, onde a especificação técnica dessa linguagem é distribuída juntamente com a Spider-PM. Além disso, também é distribuído, em conjunto com a Spider-PM e a SPIDER_ML, um manual de utilização com foco nos usuários finais da Spider-PM.

Assim como o SPEM, a SPIDER_ML e a Spider-PM apresentam uma distinção clara entre os elementos que podem ser reutilizados em diferentes processos de software e os elementos que são únicos em um processo de software. Assim sendo, a definição de um processo de software na Spider-PM ocorre em duas etapas: a definição do processo padrão; e sua instanciação para projetos específicos. De acordo com essa abordagem, em um primeiro momento é definido o conjunto de elementos básicos e comuns a todos os processos da organização e que são independentes das características do software a ser desenvolvido [Oliveira, 2006] denominado processo padrão [ISO/IEC, 2003].

Em seguida, instanciações do processo padrão para projetos específicos são realizadas considerando aspectos como: o tipo do software; os paradigmas de desenvolvimento; ciclo de vida do processo; características do projeto e da equipe responsável pelo projeto; e os recursos disponíveis para esse projeto. A ferramenta Spider-PM também permite um gerenciamento simplificado dos recursos humanos, artefatos e ferramentas utilizadas no processo de software. Além disso a ferramenta Spider-PM apresenta um mecanismo para definição de regras de utilização de estereótipos da UML visando permitir a extensão da semântica dos componentes da linguagem SPIDER_ML.

A linguagem 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 linguagem SPIDER_ML, a mesma oferece mecanismos que permitem a modelagem desse tipo de processo. Basicamente, a linguagem 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 linguagem SPIDER_ML encontra-se de acordo com uma das quatro abordagens para linguagens para modelagem descrita por Conradi e Liu (1995). A SPIDER_ML também se encontra de acordo com todas as

fases de meta-processo que não envolvem a simulação e execução do processo modelado definidas por Conradi e Liu (1995).

Além do desenvolvimento da ferramenta Spider-PM e da linguagem SPIDER_ML, os estudos realizados durante a execução desse projeto permitiram a elaboração do artigo científico “SPIDER_ML: Uma Linguagem de Modelagem de Processos de Software” que foi publicado nos anais da II Escola Regional de Informática. Nesse artigo, as principais características dessa linguagem são expostas.

Também com base nas pesquisas realizadas nesse projeto, foi elaborado o artigo “Spider-PM: Uma Ferramenta de Apoio a Modelagem de Processos de Software” que foi publicado nos anais do VIII Encontro Anual de Computação. Ambos eventos promovidos pela SBC – Sociedade Brasileira da Computação. Esse artigo apresenta as principais características dessa ferramenta e uma breve comparação da mesma com outras ferramentas para modelagem por meio do SPEM. Além disso, a ferramenta Spider-PM foi aceita para demonstração na seção de ferramentas do VI Workshop Anual do MPS.

Ainda como resultado desse projeto tem-se a geração de um trabalho de conclusão de curso denominado “Spider-PM: Um Ferramenta de Apoio para Modelagem de Processos de Software”. Esse trabalho apresenta todos os principais aspectos da linguagem SPIDER_ML e da ferramenta Spider-PM. Esse trabalho inclui análises detalhadas das características da linguagem SPIDER_ML e da ferramenta Spider-PM. Esse trabalho também apresenta uma comparação relacionando a linguagem SPIDER_ML com as principais linguagens de modelagem de processos e uma comparação relacionando a ferramenta Spider-PM com as principais ferramentas para modelagem por meio do SPEM.

5. Aplicabilidade dos Resultados

Sabe-se que o aperfeiçoamento contínuo de um processo de software é um fator que contribuem para a melhoria do software obtido ao final desse processo. Sabe-se, também, que a modelagem de processos de software é um mecanismo capaz de facilitar o aperfeiçoamento do processo de software [Oliveira, 2006]. Assim, os resultados desse projeto devem contribuir diretamente para o aumento da qualidade dos softwares, uma vez que esses resultados visam facilitar a modelagem de processos de software.

Esses resultados devem influenciar inicialmente qualquer organização que visa utilizar a modelagem de processos como mecanismo para apoiar a melhoria dos seus processos de software. Além disso, a Spider-PM foi desenvolvida de modo a estar totalmente compatível com o *Capability Maturity Model Integration for Development* (CMMI-DEV) [SEI, 2006] e com o programa de Melhoria do Processo de Software Brasileiro (MPS.BR) [SOFTEX, 2009]. Dessa forma, todas as organizações que seguem as recomendações do CMMI-DEV ou do MPS.BR podem utilizar a Spider-PM e os demais resultados desse projeto de modo a facilitar o aperfeiçoamento dos seus processos de software.

6. Características Inovadoras

O principal diferencial da ferramenta Spider-PM juntamente com a linguagem SPIDER_ML consiste na utilização das principais práticas para modelagem de processo da indústria de software, como mencionado, essas práticas foram levantadas a partir de pesquisas em referências bibliográficas e com base em observações realizadas durante a implantação de programas de melhoria do processo de software.

A seguir tem-se uma lista com as principais características da ferramenta Spider-PM, isoladamente, essas características podem ser encontradas em outras ferramentas para modelagem de processos de software, no entanto, todo esse conjunto de características presentes em uma única ferramenta também constitui uma inovação da Spider-PM:

- **reutilização das notações do SPEM e da UML:** tal característica, como descrito, permite que muitos profissionais já familiarizados com essas notações possam reaproveitar seus conhecimentos na construção de modelos de processos na ferramenta 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 ferramenta 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 ferramenta 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 o CMMI-DEV e com o MPS.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 ferramenta Spider-PM;

- **executa a partir da Java Standard Runtime Environment 1.6 (JRE 1.6):** a ferramenta Spider-PM pode ser facilmente inserida em diferentes ambientes de desenvolvimento, uma vez que ela pode ser executada em qualquer sistema compatível com essa tecnologia;
- **permite a extensão da semântica dos componentes para modelagem:** todo e qualquer elemento de modelagem da ferramenta 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;
- **publica os processos modelados por meio de páginas da internet:** A ferramenta Spider-PM exporta os modelos de processos como páginas da internet. Assim, os processos de software podem ser divulgados para as partes interessadas e também podem ficar.

A Tabela 1 mostra uma avaliação das principais ferramentas de apoio à modelagem de processos de software que utilizam o SPEM [Barros, 2010]. Para fins de comparação com a ferramenta Spider-PM, resalta-se que a mesma é executada com base na plataforma Java na versão 1.6, além disso, a ferramenta Spider-PM atende a todos os demais critérios presentes na Tabela 1.

Outro diferencial da ferramenta Spider-PM é que ela foi desenvolvida para lidar com os conhecimentos envolvidos nos processos Definição do Processo Organizacional (DFP) e Avaliação e Melhoria do Processo Organizacional (AMP) do MPS.BR. Com relação ao processo DFP, a ferramenta Spider-PM é parcialmente aderente aos resultados esperados DFP1, DFP2, DFP3 e DFP4. Do processo AMP, a ferramenta Spider-PM é parcialmente aderente aos resultados esperados AMP1 e AMP2. Com relação ao DFP1, DFP2, DFP4, AMP1 e AMP2, a ferramenta 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 manterá um registro de todas as modificações realizadas no arquivo XML que armazena o processo de software modelado. Com relação ao DFP2, a ferramenta Spider-PM é parcialmente aderente devido a não realizar uma simulação dos processos modelados, deste modo, a ferramenta Spider-PM não determina automaticamente o desempenho esperado dos processos de softwares. Apesar dessa relação com o MPS.BR, tanto processos de software baseados no MPS.BR quanto não baseados podem ser modelados nessa ferramenta [Barros; Oliveira, 2010b].

Tabela 1. Comparação entre as ferramentas para modelagem por meio do SPEM [Barros, 2010].

	APES 2.5	DSL4SPM	Eclipse Process Framework Composer 1.5	Enterprise Architect 8.0	Microsoft Visio 2010	Objecteering Enterprise Edition 6.1	Objecteering Free Edition 6.1	StarUML 5.0
Plataforma:	Java 1.6	Visual Studio	Linux Windows	Linux Windows Macintosh	Windows	Linux Solaris Windows	Linux Windows	Windows
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
!	Parcialmente
✗	Não
?	Informações não suficientes para definir

7. Conclusões e Perspectivas Futuras

Com base nas pesquisas realizadas nesse projeto, observa-se que a ferramenta Spider- PM sistematiza a construção de modelos de processo de software de maneira mais simplificada, se comparada com as demais soluções existentes, pois ela apresenta um pequeno conjunto de elementos e diagramas. Essa simplificação, aliada com as validações que ocorrem durante a construção dos modelos de processos, e a reutilização de notações UML e do SPEM, contribui para a rápida assimilação da semântica dos elementos da modelagem e entendimento dos modelos construídos.

Dessa forma, a ferramenta Spider-PM facilita o atendimento dos objetivos da modelagem de processos de software pelas organizações, contribuindo positivamente para o aperfeiçoamento, comunicação, gerenciamento e entendimento dos processos de software das organizações.

Além disso, a Spider-PM também é uma ferramenta com interface gráfica inspirada nos modeladores UML mais populares. Portanto, isso facilita ainda mais sua utilização, pois a Spider-PM apresenta um processo de iteração conhecido pela maioria dos profissionais envolvidos no processo de software. A Spider-PM também é uma ferramenta totalmente baseada em tecnologias livres, assim, além dela poder ser utilizada sem nenhuma limitação e sem implicar em nenhum custo, todas as tecnologias necessárias para a execução da mesma também são gratuitas. Essa característica constitui um diferencial da Spider-PM, uma vez que, como constatado por Bendraou *ET al* (2010), soluções proprietárias constituem uma dificuldade para que as organizações possam usufruir dos benefícios da modelagem de processos.

Futuramente, a ferramenta Spider-PM ainda 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 [Oliveira, 2010] e a Spider-PM é uma das ferramentas que fará parte desse ambiente. A ferramenta Spider-PM tem um papel essencial nesse ambiente uma vez que a simulação, execução, definição, avaliação e melhoria dos processos de software serão realizadas a partir dos processos de software modelados na ferramenta Spider-PM.

Outro trabalho futuro consiste no estabelecimento de uma interface de integração da ferramenta Spider-PM com outras ferramentas de apoio ao desenvolvimento de software. A Spider-PM deve ser a ferramenta que estabelecerá a comunicação entre diferentes ferramentas livres de um suíte para apoio à implementação dos processos do MPS.BR que se encontra em desenvolvimento no Projeto SPIDER. Atualmente, a ferramenta Spider-PM permite iniciar a execução das ferramentas presentes nos processos modelados. No entanto, espera-se que, futuramente, seja possível, por exemplo, acessar os dados dessas ferramentas e não somente iniciar a execução das mesmas.

8. Agradecimentos

O desenvolvimento deste trabalho está sendo apoiado financeiramente pelo programa PARD/UFGA.

Referências Bibliográficas

Barros, R. S. (2010) "Spider-PM: Um Ferramental de Apoio para Modelagem de Processos de Software". Trabalho de Conclusão de Curso. Faculdade de Computação – Instituto de Ciências Exatas e Naturais – Universidade Federal do Pará. Belém.

Barros, R. S., Oliveira, S. R. B. (2010a) "SPIDER_ML: Uma Linguagem de Linguagem de Modelagem de Processos de Software". II Escola Regional De Informática. Manaus.

Barros, R. S., Oliveira, S. R. B. (2010b) "Spider-PM: Uma Ferramenta de Apoio à Modelagem de Processos de Software". VIII Encontro Anual De Computação. Catalão.

Bendraou, R. et al. (2010) "A Comparison of Six UML-Based Languages for Software Process Modeling. Software Engineering". IEEE Transactions on, v.36, n.5, p.662- 675. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5593045>>. Acesso: 20 set. 2010.

Conradi, R., Liu, C. (1995) "Process Modelling Languages: One or Many?". EWSPT '95 Proceedings of the 4th European Workshop on Software Process Technology, p. 98-118, Londres.

ISO/IEC - International Organization for Standardization / International Electrotechnical Commission. (2003) "ISO/IEC 15504-2: Information Technology - Process Assessment – Part 2 – Performing an Assessment". Geneve.

Kellner, M. I., Hansen G. A. (1988) " Software Process Modeling". Pittsburgh. Disponível em: <<http://www.sei.cmu.edu/reports/88tr009.pdf>>. Acesso: 18 set. 2010.

Lonchamp, J. (1993) "A Structured Conceptual and Terminological Framework for Software Process Engineering". Second International Conference on the Continuous Software Process Improvement, vp.41-53. Disponível em: <<http://ieeexplore.ieee.org /stamp/stamp.jsp?tp=&arnumber=236823>>. Acesso: 18 set. 2010.

Oliveira, S. R. B. (2006) "Processo de Software: Princípios, Ambiente e Mecanismos de Execução". Exame de Qualificação do Programa de Doutorado. Centro de Informática – Universidade Federal de Pernambuco. Recife.

Oliveira, S. R. B. *et al* (2010) "SPIDER - Um Suite de Ferramentas de Software Livre de Apoio à Implementação do modelo MPS.BR". Anais do VIII Encontro Anual de Computação – ENACOMP 2010, Catalão-GO.

OMG – Object Management Group (2008) "Software & Systems Process Engineering Meta-Model Specification". Disponível em: <<http://www.omg.org/spec/SPEM/2.0/PDF>>. Acesso: 26 mar. 2010.

SEI – Software Engineering Institute. (2006) "CMMI for Development". Pittsburgh. Disponível em: <<http://www.sei.cmu.edu/reports/06tr008.pdf>>. Acesso: 20 set. 2010.

SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2009) "MPS.BR – Melhoria de Processo do Software Brasileiro – Guia Geral". Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_Geral_2009.pdf>. Acesso: 18 set. 2010

Spider-QA: Uma Ferramenta de Software Livre de Apoio à Garantia da Qualidade

Marília Paulo Teles¹, Sandro Ronaldo Bezerra Oliveira¹

¹Programa de Pós-Graduação em Ciência da Computação - Universidade Federal do Pará (UFPA) -
Rua Augusto Corrêa, 01 - Guamá - 91501-970 - Belém - PA - Brasil

mariliapaulo@gmail.com, srbo@ufpa.br

Abstract. *The Software Quality Assurance process is essential to improve both product and development process of an organization. This paper presents the Spider-QA, a project to assist implementing the quality assurance process in an organization, which makes part of a biggest project known as SPIDER developed at Federal University of Pará. The project attends MR-MPS model (Modelo para Melhoria do Processo de Software Brasileiro – Model for Brazilian Software Process Improvement), launched in 2003 by SOFTEX (Associação para Promoção da Excelência do Software Brasileiro - Association for Promoting Brazilian Software Excellence).*

Resumo. *O processo da Garantia da Qualidade de Software é indispensável para a melhoria do produto e do processo de desenvolvimento de uma organização. Desse modo, este artigo apresenta a Spider-QA, um projeto que visa apoiar a implantação do processo de Garantia de Qualidade em uma empresa, que faz parte de um projeto maior chamado SPIDER, desenvolvido na Universidade Federal do Pará. O projeto atende ao modelo MR-MPS (Melhoria do Processo de Software Brasileiro), criado em 2003 pela SOFTEX (Associação para Promoção da Excelência do Software Brasileiro).*

1. Introdução

O mercado de software é um mercado em expansão, que dia após dia se torna mais concorrido e mais rigoroso no grau de complexidade, ou seja, a cada dia os softwares se tornam mais complexos, pois a sua clientela espera soluções mais completas (Bartié, 2002). Nesse contexto as organizações que produzem softwares precisam se reinventar e se destacar. O processo de qualidade de software passa a ser então um diferencial entre as organizações (Rocha, 2001).

Organizações que possuem um processo de qualidade de software são capazes de diminuir o número de defeitos nos seus produtos, o retrabalho e o custo de seus softwares (Koscianski, 2007). Isso acaba por se tornar o diferencial necessário para que as empresas despontem no mercado.

Nesse contexto, surge o SPIDER (www.spider.ufpa.br) (Oliveira, 2010). Um projeto desenvolvido na Universidade Federal do Estado do Pará que visa auxiliar a implantação do modelo MPS.BR – Melhoria do Processo de Software Brasileiro (SOFTEX, 2009a). Esse projeto encontra-se dividido em sub-projetos, cada um para uma área de processo específica do MPS.BR. Este artigo é portanto sobre um dos subprojetos do SPIDER, sobre o sub-projeto de Garantia da Qualidade de Software, conhecido como Spider-QA.

O Spider-QA é um projeto que visa auxiliar na institucionalização do processo de Garantia da Qualidade nas organizações desenvolvedoras de softwares por meio da utilização de ferramentas de software livre. Para isto foi proposto um processo passível de ser implantado nas organizações e também foi desenvolvida uma ferramenta capaz de apoiar o processo proposto e assim facilitar o processo de implantação do processo de Garantia da Qualidade nas organizações.

Este artigo encontra-se estruturado em sete seções. A seção 2 apresenta os objetivos e a justificativa para a realização do projeto. Na seção 3 é apresentada a metodologia de execução do Projeto. A seção 4 mostra os resultados que foram obtidos no decorrer do projeto. A seção 5 exemplifica a aplicabilidade dos resultados obtidos com a execução do Projeto. A seção 6 mostra as características que tornam o projeto inovador. E, finalmente, a seção 7 apresenta as conclusões e as perspectivas futuras para o projeto.

2. Objetivos e Justificativa

O projeto Spider-QA surge com o objetivo de auxiliar a implantação da Gerência de Qualidade de Software do modelo MR-MPS, que é o modelo de referência do MPS.BR, nas organizações com a possibilidade da utilização de uma ferramenta. Diz-se que é com a possibilidade da utilização da ferramenta, pois o uso do processo e da ferramenta são decisões distintas, é possível utilizar somente o processo ou somente a ferramenta.

Além da produção de um processo e de uma ferramenta voltados para a Garantia da Qualidade, o projeto Spider-QA visa a descoberta de empresas parceiras capazes de executar o processo e testar a ferramenta, pois assim, os produtos propostos pelo projeto poderão ser testados, devidamente ajustados e consequentemente aprimorados.

O projeto Spider-QA possui ainda o objetivo mais amplo de auxiliar, inicialmente, as organizações da região Norte a se tornarem mais competitivas no mercado nacional por meio da adoção do processo de Garantia da Qualidade, o que possibilitaria uma redução na quantidade de defeitos e uma melhor aceitação pelo mercado.

Este projeto justifica-se então pela possibilidade de auxiliar as organizações a implantarem o processo de Garantia da Qualidade, e por meio disso auxiliá-las também a produzir *softwares* de qualidade, com processos e produtos de trabalho em conformidade com os processos organizacionais. O que acarretaria para as organizações menos retrabalho e menos defeitos no consumidor final.

Além disto, existe a preocupação em auxiliar o crescimento das organizações da região norte, e assim, propiciar tanto o aprimoramento da mão de obra local quanto o seu estabelecimento na região.

3. Metodologia de Execução

Para o desenvolvimento do Projeto Spider-QA foram utilizados alunos de mestrado, graduação e iniciação científica. Os alunos trabalharam em conjunto, observando as especificidades do processo de Garantia da Qualidade e em seguida propuseram o processo Spider-QA, específico para esta área. Para tanto, foi realizado um mapeamento entre os ativos constantes no principais modelos, normas e padrões que possuem boas práticas relacionadas ao processo de Garantia da Qualidade.

Depois do processo já definido, os alunos trabalharam na definição da ferramenta. Juntos definiram o protótipo da ferramenta, a modelagem de banco de dados e a metodologia de desenvolvimento. Em seguida teve início a construção da ferramenta.

Com a ferramenta já pronta os alunos estabeleceram contatos com uma organização parceira do projeto, a Cobra Tecnologia, e nessa organização efetuaram um estudo de caso sobre a aplicação da ferramenta em um projeto real.

Dessa integração com a organização parceira, além do teste da ferramenta em si, foram obtidas opiniões sobre a ferramenta, essas opiniões foram analisadas, e se tornaram o insumo necessário para permitir as atualizações de melhorias da ferramenta.

Com essa metodologia de desenvolvimento os alunos do projeto Spider-QA puderam experimentar a teoria dentro de um ambiente real de avaliação. Assim, propiciou-se a integração da tripla “ensino, pesquisa e extensão”.

4. Resultados Obtidos

Com o desenvolvimento do projeto Spider-QA alguns resultados relevantes puderam ser obtidos. Como um processo de software, uma ferramenta, artigos, capacitação de recursos humanos e uma dissertação de mestrado.

A execução do projeto começou com o desenvolvimento de um processo relacionado à área de Garantia da Qualidade que visa auxiliar a implantação dessa gerência em uma organização. No processo foram definidas três macro-atividades: Planejamento e Preparação, Execução e Acompanhamento. Essas macro-atividades foram decompostas em outras quatorze atividades, que quando executadas garantem a aderência aos resultados esperados da área da Garantia da Qualidade do Modelo de Referência do MPS.BR (SOFTEX, 2009b).

A ferramenta Spider-QA foi produzida. Esta ferramenta está aderente tanto ao processo proposto quanto ao Modelo de Referência do MPS.BR. É importante observar que a utilização da ferramenta e do processo são decisões distintas para a organização, ou seja, a organização pode decidir por utilizar apenas um dos dois sem prejuízo para a mesma. Buscou-se não associar a utilização da ferramenta à utilização do processo, para que as organizações pudessem utilizar os seus próprios processos de Garantia da Qualidade, e assim não causar maiores impactos para a organização.

A ferramenta desenvolvida está dividida em módulos, é voltada para a *web*, e possui uma interface agradável e intuitiva para facilitar a sua utilização. Ela é um *software* livre, com código aberto e encontra-se disponível para download no site do projeto SPIDER (www.spider.ufpa.br acessando a guia “Downloads”) juntamente com o seu manual de instalação e utilização.

Com o desenvolvimento deste projeto foram publicados dois artigos. O primeiro artigo intitulado “*Spider-Qa: A Systematic Approach For Quality Assurance Process*” foi publicado na Conferência Internacional de Engenharia de Software 2010 (SEA – *Software Engineering and Applications* – 2010) (Teles, 2010). O segundo artigo foi aceito na revista Traços, intitulado “Spider-QA: Uma Solução de Apoio ao Processo de Garantia da Qualidade”, e será publicado na sua edição 27 com data de publicação provável para junho de 2011.

Durante o desenvolvimento do projeto, os alunos tiveram seu interesse pelo modelo MPS.BR despertado e alguns realizaram o curso C1 e a prova P1 promovida pela SOFTEX, obtendo 100% de aprovação, o que pode evidenciar o comprometimento e a dedicação dos alunos do projeto SPIDER.

E por fim, está sendo produzida uma dissertação de mestrado. A dissertação encontra-se em sua fase final de produção e correção. A defesa dessa dissertação está prevista para Maio de 2011.

5. Aplicabilidade dos Resultados

Os resultados obtidos com a execução do projeto Spider-QA foram muitos e das mais variadas naturezas. A aplicabilidade dos mesmos se torna abrangente e considerável na medida em que leva o projeto Spider-QA e também o projeto SPIDER a reconhecimento nacional e internacional.

A produção do processo e da ferramenta garantiu às organizações a verificação da seriedade e da utilidade de uma proposta acadêmica para um problema da indústria de software. Isto mostrou que a determinação em tornar o projeto não somente um estudo, mas, também, uma proposta capaz de ser utilizada em ambientes reais de produção de *software* é uma meta atingida pelo projeto.

Outra aplicabilidade recai na aprovação dos artigos em um evento internacional e em uma revista classificada como qualis B5 pela CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior), o reconhecimento da pesquisa, da aplicabilidade e da seriedade do Projeto Spider-QA feito pela comunidade acadêmica.

Atribuir resultados das áreas acadêmicas e da indústria de software ao projeto Spider-QA foi um verdadeiro desafio dada a singularidade de ambas as áreas. Um desafio que se acredita ter sido alcançado com eficiência pelos alunos e pelo coordenador do projeto.

6. Características Inovadoras

O modelo MPS.BR para qualidade de software é um modelo relativamente novo, que está sendo amplamente explorado pela academia por ser um modelo brasileiro, voltado para a realidade das pequenas e médias empresas brasileiras.

Entretanto, um trabalho com a perspectiva de um processo e uma ferramenta para auxiliar na implantação da gerência da Garantia da Qualidade em organizações brasileiras, que seja totalmente aderente ao MPS.BR e que produza as evidências que podem ser utilizadas durante uma possível avaliação da organização para o MPS.BR, ainda não era conhecida. Portanto, a utilização da ferramenta e do processo podem se tornar um diferencial para as empresas que os adotarem.

Além disso foi efetuado um mapeamento entre o modelo MPS.BR, o modelo CMMI- *Capability Maturity Model Integration*, a norma ISO/IEC 12207, o processo do SWEBOK - *Software Engineering Body of Knowledge* e o processo do PMBOK – *Project Management Body of Knowledge*. Esse mapeamento visou facilitar a adoção da solução proposta pelas organizações. Assim, permite-se que organizações com processos baseados em outros padrões, que não o MPS.BR, possam fazer uso da proposta com menores impactos em seu dia-a-dia.

O mapeamento não é uma pretensão do projeto em atender todos os requisitos desses outros processos, normas ou modelos, mas sim um diferencial proposto para o projeto com o intuito de torná-lo mais abrangente e acessível.

Outro fator diferencial do projeto Spider-QA é a integração da ferramenta proposta com ferramentas de *bugtracking* para que o acompanhamento das nãoconformidades seja efetuado através do uso desse tipo de ferramenta. Assim, propõe-se aproveitar as ferramentas que já são utilizadas na organização e também minimizar o impacto da adoção da solução proposta no projeto Spider-QA.

7. Conclusões e Perspectivas Futuras

O projeto Spider-QA atingiu sua fase final de desenvolvimento. No entanto ainda existem pontos que podem ser futuramente explorados no projeto Spider-QA, como: o aumento do número de ferramentas de *bugtracking* com que a ferramenta é capaz de se comunicar, e o desenvolvimento de um módulo de relatórios capaz de ser customizado pela organização que fizer uso da ferramenta.

Espera-se ainda que o processo Spider-QA possa ser implantado em outras organizações para que assim ele possa ser avaliado em outras realidades e, caso necessário, ele seja aprimorado.

O projeto SPIDER como um todo continuará desenvolvendo soluções automatizadas para a implantação do MPS.BR em pequenas e médias organizações e servindo como referência para novos estudos nas empresas da região Norte que reúnam a trílice “ensino, extensão e pesquisa”.

Referências Bibliográficas

Bartí, A. Garantia da Qualidade de Software. 9^o Ed. Rio de Janeiro: Elsevier, 2002.

Koscianski, André; Soares, Michel dos Santos. Qualidade de Software. 2^o Ed. São Paulo: Novatec Editora, 2007.

Oliveira, S. R. B. *et al* (2010) "SPIDER - Um Suite de Ferramentas de Software Livre de Apoio à Implementação do modelo MPS.BR". Anais do VIII Encontro Anual de Computação – ENACOMP 2010, Catalão-GO.

Rocha, A. R. C., et al. Qualidade de Software - Teoria e Prática. Prentice Hall, 2001.

SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2009a) "MPS.BR - Guia Geral".

SOFTEX – Associação para Promoção da Excelência do Software Brasileiro (2009b) "MR-MPS – Modelo de Referência".

Teles, P. M.; Oliveira, S. R. B. "Spider-Qa: A Systematic Approach For Quality Assurance Process". SEA – *Software Engineering and Applications*. Califórnia. 2010.

Utilizando *Grounded Theory* para Compreender a Aceitação de uma Técnica de Elicitação de Requisitos

Amadeu Anderlin Neto¹, Cristina Araújo¹, Tayana Conte¹

¹Departamento de Ciência da Computação Universidade Federal do Amazonas (UFAM) – Manaus, AM - Brasil

{neto.amadeu, crisoara, tayana}@dcc.ufam.edu.br

Abstract. *The requirements elicitation phase is critical to the software project success. If the requirements are not well defined, the entire project development cycle can be compromised. Therefore, it is necessary to use a requirements elicitation technique that is suitable for a particular situation.*

However, few analysts make use of different elicitation techniques. This project proposes the use of procedures of Grounded Theory in the qualitative analysis of the acceptance of a particular requirements elicitation technique.

Preliminary results, obtained from an exploratory research in a software development company, indicate the main factors when accepting and using a new elicitation technique.

Resumo. *A fase de elicitação de requisitos é fundamental para o sucesso do projeto de software. Se os requisitos não forem bem definidos, todo o ciclo de desenvolvimento do projeto pode ser comprometido. Por isso, faz-se necessário utilizar uma técnica de elicitação de requisitos que seja adequada a uma determinada situação. No entanto, são poucos os analistas que fazem uso de diferentes técnicas de elicitação. Este projeto propõe a utilização de procedimentos da Grounded Theory na análise qualitativa de aceitação de uma determinada técnica de elicitação de requisitos. Resultados iniciais, obtidos a partir de uma pesquisa exploratória em uma empresa, indicam os fatores mais críticos na aceitação e utilização de uma nova técnica de elicitação.*

1. Introdução

A Engenharia de Requisitos (ER) abrange as atividades de elicitação, análise, especificação e validação de requisitos (Kotonya e Sommerville, 1998). De acordo com Dieste *et al.* (2008), a elicitação de requisitos é uma das atividades mais complexas no processo de desenvolvimento de *software*.

Pensando em minimizar tais problemas, surgiram as técnicas de elicitação de requisitos. As técnicas existentes são utilizadas em determinados cenários, ou, até mesmo, adaptadas a eles. Apesar disso, utiliza-se entrevistas na maioria dos casos (Carrizo *et al.*, 2008).

Com este cenário, faz-se necessário investigar a aceitação de técnicas de elicitação em diferentes contextos. Quais os fatores que fazem com que um analista utilize uma determinada técnica de elicitação?

Neste projeto, é apresentado um estudo de caso sobre a aceitação de uma determinada técnica de elicitação de requisitos, no contexto de uma pequena empresa de desenvolvimento de *software*. Trata-se de um estudo qualitativo, de caráter exploratório, que objetiva investigar a aceitação de uma determinada técnica de elicitação por parte de analistas que adotavam, principalmente, entrevistas. Os dados coletados no estudo foram analisados utilizando procedimentos da *Grounded Theory* (GT).

O restante deste artigo está estruturado da seguinte forma. A seção 2 mostra a justificativa e os objetivos deste projeto. A seção 3 expõe a metodologia utilizada para sua execução. A seção 4 apresenta os fatores identificados no estudo. A seção 5 exhibe a aplicabilidade dos resultados. A seção 6 cita as características inovadoras desta pesquisa. Por fim, a seção 7 conclui o artigo.

2. Objetivos e Justificativa

Este projeto tem como objetivo geral obter indícios de fatores que influenciam um analista a adotar uma determinada técnica de elicitação de requisitos. A Tabela 1 apresenta o objetivo através do paradigma GQM (*Goal-Questions-Metrics*), que sugere que o objetivo de um estudo experimental em Engenharia de *Software* seja elaborado de forma estruturada (Basili e Rombach, 1988).

Tabela 1. Objetivo do estudo de caso segundo o paradigma GQM.

Analisar	a aceitação de uma técnica de elicitação de requisitos
Com o propósito de	caracterizar
Em relação à	identificação de fatores que impactam na aceitação e utilização da técnica
Do ponto de vista	dos analistas
No contexto de	uma empresa de desenvolvimento de software

Com o estudo realizado, quatro contribuições importantes para a área de ER são apresentadas: (1) relato da condução do estudo de caso; (2) resultados da análise qualitativa dos dados sobre os fatores que levam à aceitação de uma técnica de elicitação de requisitos; (3) disseminação do método de pesquisa qualitativa junto à comunidade de pesquisadores em ER; e (4) uma contribuição muito importante do presente projeto é a proposição de utilização dos procedimentos de GT para a análise qualitativa dos dados coletados no estudo.

3. Metodologia de execução

Para o presente projeto, foi realizado um estudo de caso, onde os dados coletados foram analisados qualitativamente através dos procedimentos da GT.

O método de pesquisa GT (Teoria Fundamentada em Dados) é definido como a teoria derivada de dados sistematicamente coletados e analisados (Strauss e Corbin, 1998). A GT se preocupa em gerar uma teoria substantiva com alto poder explicativo que seja capaz de explicar como e por que ocorrem diferentes formas de comportamento e como podem se modificar no decorrer do tempo. A análise dos dados é feita através de codificação que, segundo Douglas (2003), é o resultado de fazer questionamentos e dar respostas provisórias sobre categorias e suas relações. A codificação divide-se em três fases: aberta, axial e seletiva.

Para o estudo de caso, foi escolhida apenas uma técnica de elicitação para evitar viés no aprendizado. Foram analisadas quais técnicas os analistas da empresa não conheciam e, então, a técnica baseada em *mockups* foi selecionada. Esta técnica, proposta por Unger e Chandler (2009), baseia-se em protótipos visuais (*mockups*) desenhados utilizando *wireframes*, o que ajuda a dar maior clareza e direcionamentos aos analistas no processo de elicitação. Unger e Kane (2009) afirmam que prototipagem é o ato de criar e testar toda ou parte da funcionalidade de uma aplicação ou *site* com os usuários.

Os participantes do estudo foram dois analistas da empresa investigada, sendo um pleno (oito anos de experiência) e um júnior (dois anos de experiência). Além deles, participaram um pesquisador especialista na técnica de elicitação baseada em *mockups* e um pesquisador-observador.

O pesquisador especialista na técnica fez uma apresentação na empresa dos analistas, ensinando a técnica. Posteriormente, foi realizada uma entrevista com cada um dos analistas, individualmente, com questões que tratavam sobre o entendimento e aceitação técnica baseada em *mockups*. Foi dado um prazo de utilização da técnica de duas semanas, onde o pesquisador-observador coletou dados de utilização e aceitação da técnica. Após o período de observação, foi realizada outra entrevista com os mesmos analistas, individualmente, com questões relacionadas à utilização da técnica.

Todas as perguntas do questionário eram de caráter subjetivo, onde os entrevistados fornecem sua própria opinião. Com base nisso, foram realizadas duas entrevistas com cada analista, tratando de questões sobre três perspectivas: entendimento, aceitação e utilização da técnica baseada em *mockups*. A Figura 1 mostra parte do questionário utilizado nas entrevistas.

I – Questões sobre o Entendimento da Técnica: I.1) Como você entendeu a técnica? I.2) O que mais chamou sua atenção na técnica? I.3) Qual(is) a(s) dificuldade(s)/facilidade(s) em aprender a técnica? I.4) Você ficou com alguma dúvida sobre como utilizar a técnica?
II – Questões sobre Aceitação da Técnica: II.1) Do seu ponto de vista, quais as principais facilidades/ dificuldades da técnica? ...

Figura 1. Parte do questionário utilizado nas entrevistas.

Após a finalização de todas as entrevistas, foi feita a análise dos dados de forma qualitativa, através dos procedimentos da GT. Na próxima seção serão mostrados os resultados obtidos.

4. Resultados Obtidos

4.1. Análise Qualitativa dos Dados

Para iniciar a análise qualitativa dos dados, foi eleito um questionário como fonte inicial de exploração (Bandeira-de-Mello e Cunha, 2003). Utilizando o software Atlas.ti¹⁷, foi iniciada a fase de codificação aberta, conforme Figura 2. Os códigos gerados eram relacionados a citações do questionário que foi respondido pelo entrevistado (as entrevistas foram transcritas). Dois pesquisadores fizeram a análise inicial e, posteriormente, a revisaram. Após esta revisão, foi iniciada a codificação do outro questionário, seguindo os mesmos procedimentos iniciais, ou seja, analisando os dados e criando códigos a partir deles.

¹⁷ 1 Atlas.ti – The Knowledge Workbench, Scientific Software Development – <http://www.atlasti.com>

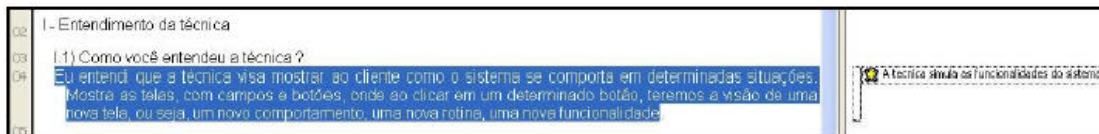


Figura 2. Exemplo de um código relacionado com a citação no texto na etapa de *codificação aberta*.

Finalizada a codificação aberta, os códigos gerados foram separados em grupos de acordo com suas propriedades. Ao iniciar a etapa de codificação axial, os pesquisadores notaram que os códigos gerados durante a codificação aberta poderiam ser melhorados, de modo que o relacionamento entre eles fosse melhor compreendido.

A partir disso, foi feita uma revisão completa na codificação aberta e, ao final, foram gerados trinta e dois códigos.

Terminada a codificação axial, foram identificadas quatro categorias principais: “*Adaptação da técnica à realidade da empresa*”, “*Problemas de utilização da técnica de entrevistas*”, “*Facilidades da utilização da técnica*” e “*Dificuldades da utilização da técnica*” (as duas últimas referem-se a técnica baseada em *mockups*).

Os códigos da categoria “*Adaptação da técnica à realidade da empresa*” (Figura 3) mostram que a técnica baseada em *mockups* foi aprendida e utilizada. Também é possível observar que a técnica foi adaptada e adotada pela empresa, sendo utilizada em conjunto com a técnica de entrevistas, que era utilizada anteriormente.

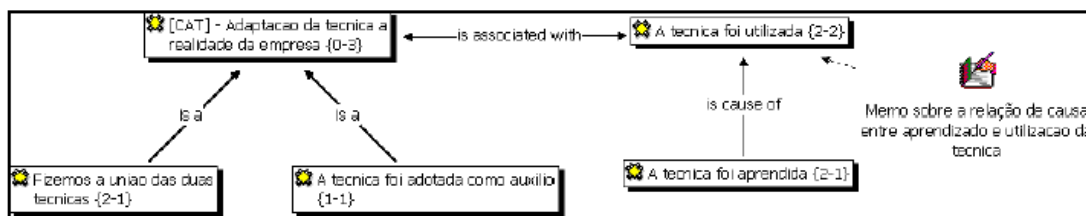


Figura 3. Representação gráfica das associações relacionadas à categoria “*Adaptação da técnica à realidade da empresa*”.

Essa adaptação foi percebida pelo pesquisador-observador e se deu da seguinte forma: (1) os analistas continuaram utilizando a técnica de entrevistas como primeiro instrumento de elicitação de requisitos; (2) após análise e organização dos requisitos obtidos com as entrevistas, os analistas desenvolviam telas que ilustravam como determinada funcionalidade se comportaria no sistema e as apresentavam aos clientes; (3) o cliente solicitava modificações que eram rapidamente atendidas; e (4) novas telas eram desenvolvidas e exibidas aos clientes até que chegassem a um consenso.

Através da análise qualitativa, foram identificadas motivações para utilização da técnica baseada em *mockups*: (1) “*o processo de construção de sites é lento e difícil*” e (2) “*existem mudanças que afetam todo o projeto*” (esses dois códigos foram retirados das entrevistas dos analistas). Os pesquisadores interpretaram que, no caso do primeiro código, o processo de desenvolvimento que o analista costumava seguir gerava algumas dificuldades no decorrer do projeto e que, utilizando uma nova técnica de elicitação, essas dificuldades poderiam ser minimizadas. No caso do segundo código, as mudanças que o cliente solicitava faziam com que o projeto sofresse alterações de cronograma, custo e esforço. Tais códigos foram associados à categoria “*Problemas de utilização da*

técnica de entrevistas” (Figura 4).

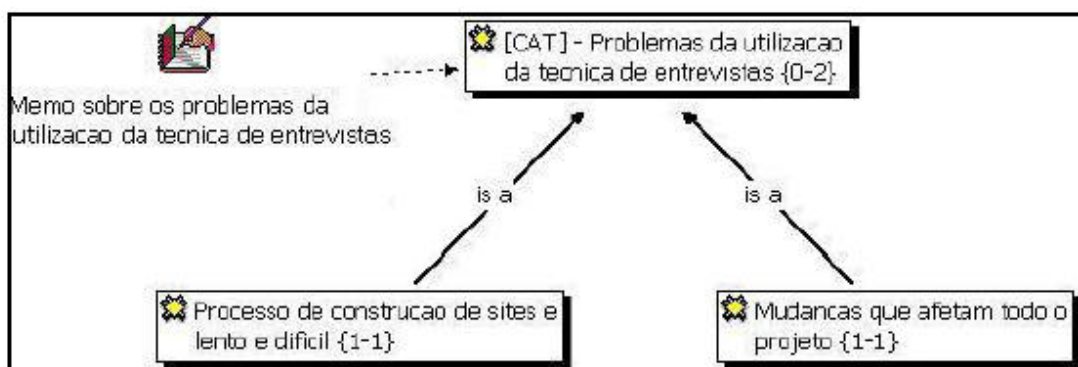


Figura 4. Associações da categoria “*Problemas de utilização da técnica de entrevistas*”.

Como facilidades da técnica baseada em *mockups*, foram identificados três aspectos: “Aspectos de facilidade relacionados à técnica”, “Aspectos de facilidade relacionados ao projeto” e “Aspectos de facilidade relacionados aos clientes”. Estas três subcategorias estão relacionadas diretamente a categoria principal chamada “Facilidades da utilização da técnica”. A Figura 5 mostra parte desta categoria.

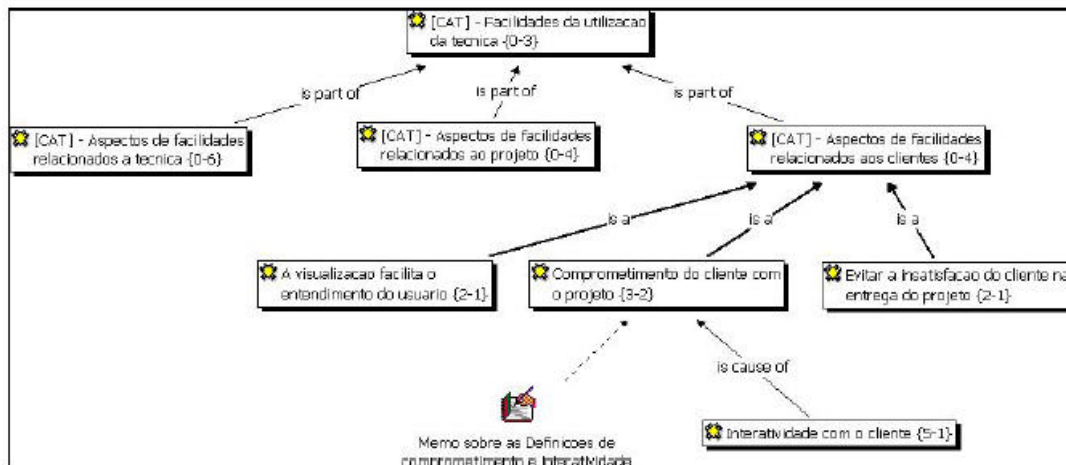


Figura 5. Representação gráfica das associações relacionadas à parte da categoria “Facilidades da utilização da técnica”.

Finalmente, os entrevistados citaram possíveis dificuldades que podem vir a ocorrer, não tendo acontecido durante o estudo. Foram identificados dois aspectos relacionados à categoria “Dificuldades de utilização da técnica” (Figura 6): “Falta de disponibilidade do cliente” e “Dificuldade em elaborar os wireframes”.

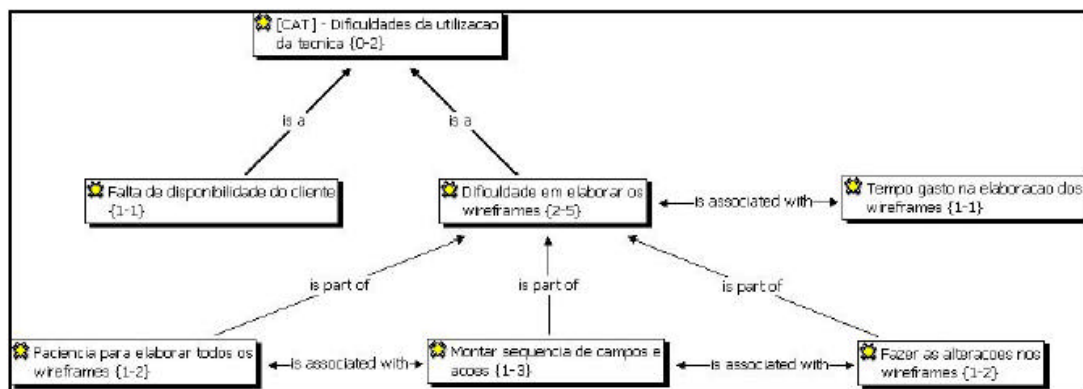


Figura 6. Representação gráfica das associações relacionadas à categoria “Dificuldades da utilização da técnica”.

No estudo de caso apresentado não foi definida a categoria central que explica o fator de aceitação da técnica de elicitação de requisitos baseada em *mockups*. Isso se deve ao fato de ter sido realizada apenas uma coleta e análise dos dados, o que não permite validar as propriedades das categorias geradas. Segundo Bandeira-de-Mello e Cunha (2006), uma das principais características da GT é a circularidade entre as fases de coleta e análise dos dados.

Os resultados obtidos neste projeto permitem a identificação preliminar de fatores que os analistas julgaram relevantes para a aceitação da técnica. Um ponto muito importante para a utilização da técnica foi a possibilidade de adaptá-la à empresa.

4.2. Artigo publicado

O artigo “Utilizando *Grounded Theory* para Compreender a Aceitação de uma Técnica de Elicitação de Requisitos” (Anderlin Neto et al. 2010) foi apresentado no VI Workshop um Olhar Sociotécnico sobre a Engenharia de Software (WOSES 2010), realizado em Belém (PA).

4.3. Outros resultados: Recurso Humano Capacitado

Trabalho de conclusão de curso de Amadeu Anderlin Neto: Bacharelado em Ciência da Computação. Universidade Federal do Amazonas – UFAM.

5. Aplicabilidade dos resultados

Embora seja muito importante apontar quais as melhores técnicas de elicitação de requisitos, igualmente importante é entender o por que, existindo tantas técnicas, as entrevistas são utilizadas na maioria das vezes.

Através deste estudo, foi possível obter fatores preliminares que indicam por qual motivo um analista adota uma determinada técnica de elicitação.

Com este estudo, busca-se contribuir para o aumento do conhecimento sobre aceitação de técnicas de elicitação de requisitos em âmbito nacional.

Através de novos estudos de caso, com novas coletas e análises dos dados, pretende-se atingir a saturação teórica, ou seja, quando não houver mais ganho de conhecimento através das entrevistas, uma teoria substantiva estará elaborada. Esta teoria nos fornecerá os fatores que levam um analista a adotar ou não uma determinada técnica de elicitação de requisitos.

6. Características inovadoras

Este estudo apresenta uma nova forma de fazer um estudo sobre aceitação de técnicas de elicitação, propondo a utilização de análise qualitativa através de procedimentos da GT.

7. Conclusões e trabalhos futuros

No presente projeto foram apresentados os resultados iniciais de uma pesquisa exploratória sobre fatores que influenciam na aceitação de uma técnica de elicitação de requisitos.

O estudo de caso realizado permitiu que fosse capturada a percepção dos analistas em relação aos fatores que os levaram a utilizar a técnica baseada em *mockups*.

Através da análise qualitativa utilizando os procedimentos de GT, foi possível propor uma categorização inicial que poderá ser alterada quando novos estudos forem realizados.

Mafrá e Travassos (2006) afirmam que uma desvantagem do estudo de caso é a dificuldade em se generalizar os resultados obtidos. Portanto, não é possível generalizar os resultados deste projeto, pois foi feita apenas uma coleta e análise de dados. Em outro cenário, novos fatores podem ser identificados.

Leitão (2009) afirma que pesquisas qualitativas definem um foco reduzido de abordagem de um problema para poder investigá-lo com a necessária profundidade.

Shull *et al.* (2004) dizem que nenhum estudo sobre determinada tecnologia pode ser considerado definitivo. Portanto, espera-se que em trabalhos futuros novos estudos sejam realizados com um cenário diferenciado.

Agradecimentos

Agradecemos à organização e aos analistas que participaram deste estudo. Agradecemos também o apoio financeiro da FAPESP e do CNPq por meio dos projetos 553164/2005-8 e 554071/2006-1.

Referências

- Bandeira-de-Mello, R., Cunha, C. (2003). *Operacionalizando o Método da Grounded Theory nas Pesquisas em Estratégia: Técnicas e Procedimentos de Análise com Apoio do Software Atlas/ti*. Encontro de Estudos em Estratégia. Curitiba, Brasil.
- Bandeira-de-Mello, R., Cunha, C. (2006). *Grounded Theory*. In: Godoi, C. K., Bandeira-de-Mello, R., Silva, A. B. d. (eds), *Pesquisa Qualitativa em Estudos Organizacionais: Paradigmas, Estratégias e Métodos*, Capítulo 8, São Paulo, Saraiva.
- Basili, V., Rombach, H. (1988). *The Tame Project: Towards Improvement-oriented Software Environments*. IEEE Transactions on Software Engineering 14(6): 758-773.
- Carrizo, D., Dieste, O., Juristo, N. (2008). *Study of Elicitation Techniques Adequacy*. 11º Workshop on Requirements Engineering (WER 2008), pp. 104-114.
- Dieste, O., Lopez, M., Ramos, F. (2008). *Updating a Systematic Review about Selection of Software Requirements Elicitation Techniques*. 11º Workshop on Requirements Engineering (WER 2008), pp. 96-103.
- Douglas, D. (2003). *Grounded Theories of Management: a Methodological Review*. Management Research News, v. 26, n. 5, pp. 44-52.
- Kotonya, G., Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*. John Wiley and Sons.
- Leitão, C. (2009). *Métodos Qualitativos de Pesquisa Científica*. Computação Brasil: Interação Humano-Computador no Brasil, pág. 22-23.
- Mafrá, S., Travassos, G. H. (2006). *Estudos Primários e Secundários Apoiando a Busca por Evidência em Engenharia de Software*. Rio de Janeiro, COPPE/UFRJ.

Shull, F., Mendonça, M., Basili, V., Carver, J., Maldonado, J., Fabbri, S., Travassos, G., Ferreira, M. (2004). *Knowledge-sharing Issues in Experimental Software Engineering*. Empirical Software Engineering, v. 9, ed. 1-2, março.

Strauss, A., Corbin, J. (1998). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Thousand Oaks: Sage Publications.

Unger, R., Chandler, C. (2009). *A Project Guide to UX Design: For User Experience Designers in the Field or in the Making*, Capítulo 11, New Riders.

Unger, R., Kane, J. (2009). *A Project Guide to UX Design: For User Experience Designers in the Field or in the Making*, Capítulo 12, New Riders.

Prêmio Dorgival Brandão Júnior da Qualidade e Produtividade em Software

Prêmio Dorgival Brandão Júnior da Qualidade e Produtividade em Software

Vencedores Anuais

Encerradas as etapas de cada Ciclo Anual de Projetos do PBQP Software, com a efetivação do processo de avaliação dos projetos concluídos durante o período vigente, os resultados são divulgados amplamente apontando os indicados para premiação.

A seguir, a relação completa dos projetos já laureados, com seus objetivos e dados básicos sobre a coordenação, entidades e responsáveis.

Vencedores do Ciclo 2010

1º Lugar:

Curso de Residência em Arquitetura de Software: Uma Parceria UFPA, UNAMA e CESUPA com AmazonCorp, CTIC-UFPA e PRODEPA

Entidades Coordenadoras: 1Laboratório de Engenharia de Software - Universidade Federal do Pará (UFPA)2, Instituto Federal do Pará (IFPA)3

Responsáveis: Rodrigo Quites Reis1, Carla Alessandra Lima Reis1, Claudio Roberto de Lima Martins2, Marcos Venícios Araújo3

E-mail: quites@ufpa.br , clima@ufpa.br, cmartins@ifpa.edu.br, maraujo@cesupa.br

Objetivo: Formar profissionais/especialistas com forte embasamento teórico e experiência prática como residentes na área de Arquitetura de Software.

2º Lugar:

Uma Investigação sobre os Fatores Críticos de Sucesso em Iniciativas de Melhoria de Processos de Software

Entidade Coordenadora: Universidade Federal do Rio de Janeiro/ Coordenação de Programa de Pós-Graduação em Engenharia - UFRJ/COPPE

Responsáveis: Mariano Angel Montoni e Ana Regina Rocha

E-mail: {mmontoni,darocha}@cos.ufrj.br

Objetivo: O objetivo deste projeto é desenvolver uma abordagem para condução de iniciativas de melhoria de processos de software com o propósito de permitir que organizações de consultoria em melhoria de processos de software estabeleçam estratégias adequadas para conduzir iniciativas de melhoria, bem como permitir a execução e acompanhamento dessas estratégias.

3º Lugar:

Agile Brazil 2010 - Conferência Brasileira sobre Métodos Ágeis de Desenvolvimento de Software

Entidade Coordenadora: Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS

Responsável: Rafael Prikladnicki

E-mail: rafaelp@pucrs.br

Objetivo: Realização da Conferência Brasileira sobre Métodos Ágeis de Desenvolvimento de Software (Agile Brazil 2010). A Agile Brazil é uma conferência nacional sem fins lucrativos organizada por representantes das principais comunidades ágeis brasileiras. O evento tem como objetivo promover a comunicação e a colaboração entre seus integrantes visando à disseminação coordenada da cultura Ágil por todo o país.

Comissão de Avaliação do Grupo de Projetos do PBQP Software Ciclo 2010

- ? Ana Cervigni Guerra - CTI
- ? Ana Liddy Cenni de Castro Magalhães - QualityFocus
- ? André Henrique de Siqueira - BCB
- ? André Villas-Boas - CPqD
- ? Carlos Lombardi - Consultor
- ? Clênio Figueiredo Salviano - CTI
- ? Cristiano Lehrer - MS
- ? Cristina Ângela Filipak Machado - CELEPAR
- ? Edmeia Leonor Pereira de Andrade - EMBRAPA

- ? Edna Pereira Fernandes - Ministério da Fazenda/SRF
- ? Francisco José Silveira de Vasconcellos - Consultor
- ? Jucele França Vasconcellos - Consultora
- ? Luiz Carlos de Almeida Oliveira - CELEPAR
- ? Maria de Fátima Mattiello Francisco - INPE
- ? Regina Maria Thienne Colombo - CTI
- ? Sarah Kohan - FCAV
- ? Salvador Alves de Melo Junior - Consultor
- ? Sonia Thereza Maintinguer - CPqD
- ? Teresa Maria de Medeiros Maciel - UFRPE
- ? Teresinha Moraes Falabella de Castro - Eletronorte

Vencedores do Ciclo 2009

1º Lugar:

Processo de Contratação de Software e Serviços Correlatos para Entes Governamentais

Entidades Coordenadoras: Tribunal de Contas da União (TCU); Empresa Brasileira de Pesquisa Agropecuária (Embrapa); Universidade de Brasília (UnB). - Faculdade FGA/Campus Gama

Responsáveis: Cláudio Silva da Cruz, Edméia Leonor Pereira de Andrade, Rejane Maria da Costa Figueiredo

E-mail: cscruz@cscruz.org, edmeia.andrade@embrapa.br, rejanecosta@unb.br

Objetivo: Propor uma versão do MPS.BR-Guia de Aquisição para entes da Administração Pública Federal, compatibilizando o processo de aquisição de software e serviços correlatos com a legislação pública brasileira para contratações de serviços de tecnologia da informação (TI) e a respectiva jurisprudência.

2º Lugar:

FireScrum: Ferramenta Open-Source

Entidades Coordenadoras: Universidade Federal Rural de Pernambuco - UFRPE; Universidade Federal de Pernambuco - UFPE; Centro de Estudos e Sistemas Avançados do Recife – CESAR

Responsáveis: Eric Cavalcanti, Teresa M. de Medeiros Maciel e Jones Albuquerque

E-mail: ecavalcanti@gmail.com; tmmaciel@gmail e jas@gmail.com

Objetivo: Definição, especificação e desenvolvimento de uma ferramenta, FireScrum, open source proposta para servir de suporte ao Scrum em equipes remotas. O objetivo principal da ferramenta é proporcionar o uso do Scrum por equipes remotas evitando a perda de histórico, propiciando o levantamento de métricas sem demandar esforço excessivo do time e, principalmente, facilitando sua adoção sem burocratizar o processo.

3º Lugar:

WDP-RT: Uma Técnica de Leitura para Inspeção de Usabilidade de Aplicações Web

Entidade Coordenadora: Universidade Federal do Amazonas - UFAM

Responsáveis: Tayana Conte e Marcos Antonio da Silva Gomes

E-mail: tayanaconte@gmail.com e marcos.sgomes@gmail.com

Objetivo: Desenvolver uma técnica de leitura para inspeção de software Web, através da extensão da técnica WDP (Web Design Perspectives-based Inspection Technique) (CONTE et al, 2007). Propor melhorias ao método, através de estudos experimentais. Referências: Conte T., Massolar, J., Mendes, E., Travassos, G. (2007) Web Usability Inspection Technique Based on Design Perspectives. SBES 2007, João Pessoa, PB, Brasil.

Comissão de Avaliação do Grupo de Projetos do PBQP Software Ciclo 2009

- ? Ana Cervigni Guerra - CTI
- ? Ana Liddy Cenni de Castro Magalhães - QualityFocus
- ? Alexandre Marcos Lins de Vasconcelos - UFPE
- ? André Villas-Boas - CPqD
- ? Carlos Lombardi - Consultor
- ? Cristiano Lehrer - ICODES

- ? Cristina Ângela Filipak Machado - CELEPAR
- ? Edna Pereira Fernandes - Ministério da Fazenda/SRF
- ? Francisco José Silveira de Vasconcellos - SAE/PR
- ? Jucele França de A. Vasconcellos - UFRJ/COPPE
- ? Maria de Fátima Mattiello Francisco - INPE
- ? Regina Maria Thienne Colombo - CTI
- ? Samy Lima Assi - FabriQ
- ? Sarah Kohan - FCAV
- ? Sonia Thereza Maintinguer - CPqD

Vencedores do Ciclo 2008

1º Lugar:

“FAPS - Ferramenta de Suporte à Avaliação de Processos de Software Alinhada com os Modelos CMMI e MPS.BR”

Entidades Coordenadoras: Universidade do Vale do Itajaí (UNIVALI) e Incremental Tecnologia em Informática

Responsáveis: Marcelo Thiry, C. G. Gresse von Wangenheim, Alessandra Zoucas, Leonardo Reis Tristão, José Luiz da Silveira

E-mail: {marcello.thiry, gresse, alessandra.zoucas, leutristao, joseluizsilveira}@gmail.com

Objetivo: Este projeto tem por objetivo desenvolver uma ferramenta de software baseada nos métodos de avaliação SCAMPI e MA-MPS, que auxilie a equipe avaliadora durante o processo de avaliação (Gerenciamento, Contextualização e Execução), suportando as atividades da avaliação. O processo de avaliação é intelectualmente complexo e requer alta experiência dos avaliadores especialmente no mapeamento dos processos e níveis de capacidade, na identificação dos riscos e sugestões de melhoria. Assim, o objetivo deste trabalho não é automatizar completamente o processo de avaliação, mas prover um suporte para a gerência dos documentos relacionados e semi-automatizar procedimentos quando possível.

2º Lugar:

“Um Framework para Avaliação da Implantação do MR MPS em Grupos Cooperados de Empresas”

Entidades Coordenadoras: Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS

Responsável: Rafael Prikladnicki

E-mail: rafael@inf.pucrs.br

Objetivo: O foco deste projeto é criação de um framework de avaliação da implantação do Modelo de Negócio Cooperado (MNC-MPS.BR) em grupos de empresas, de forma a melhorar a execução de projetos cooperados em um contexto nacional, permitindo avaliações sistematizadas e comparativas, algo que não tem sido feito atualmente.

3º Lugar:

“FROISPI - Framework Return on Investment of Software Process Improvement”

Entidades Coordenadoras: Centro de Informática da UFPE e Fundação Centro de Análise, Pesquisa e Inovação Tecnológica - Instituto de Ensino Superior (FUCAPI-CESF)

Responsáveis: Paulino W. P. Viana, Alexandre M. L. de Vasconcelos

E-mail: wagner_palheta@fucapi.br

Objetivo: Propor um Framework constituído por fases, baseado nos conceitos de Business Process Management, Metodologia ROI e DMAIC, baseando-se nos indicadores utilizados por David Rico em ROI of SPI e uma seleção de medições utilizadas para Melhorias de Processo de Software

Comissão de Avaliação do Grupo de Projetos do PBQP Software Ciclo 2008

- ? André Villas-Boas - CPqD
- ? Carlos Lombardi - Consultor
- ? Cristiano Lehrer - ICODES

- ? Cristina Ângela Filipak Machado - CELEPAR
- ? Edna Pereira Fernandes - Ministério da Fazenda/SRF
- ? Gisele Rodrigues de Mesquita Ferreira - Banco Central
- ? José Eduardo Albuquerque Medeiros - MCT/SEPIN
- ? Jucele França de A. Vasconcellos - UFRJ/COPPE
- ? Luiz Carlos de Almeida Oliveira – CELEPAR
- ? Maria de Fátima Mattiello Francisco - INPE
- ? Salvador Alves de Melo Junior - Consultor
- ? Samy Lima Assi - FabriQ
- ? Sonia Thereza Maintinguer - CPqD
- ? Teresa Maria de Medeiros Maciel - C.E.S.A.R
- ? Teresinha Moraes Falabella de Castro - Eletronorte

Vencedores do Ciclo 2007

Dos 60 projetos integrantes desse Ciclo foram relatados 54% dos projetos, sendo que 22 foram concluídos (37%) e encaminhados para avaliação final com seus respectivos artigos.

Este processo de avaliação foi finalizado em Brasília/DF, nos dias 14 e 15 de maio de 2008, durante o Workshop da Qualidade e Produtividade em Software - WQPS.

A seguir, o resultado da classificação final:

1º Lugar:

“Plataforma de Tecnologia da Informação e Comunicação de Santa Catarina - PLATIC”

Entidades Coordenadoras: Instituto Euvaldo Lodi de Santa Catarina - IEL/SC e pela Universidade Regional de Blumenau - FURB

Responsáveis: Carlos Eduardo Negrão Bizzotto, Eliza Coral e Valéria Arriero Pereira

E-mail: carlos_bizzotto@uol.com.br, elizaiel@yahoo.com.br, varriero@gmail.com

Objetivo: Desenvolver e disponibilizar um conjunto de ferramentas que melhorem a qualidade e a produtividade, ampliando a competitividade das empresas. Essa competitividade será obtida por meio da melhoria do processo de desenvolvimento, dos produtos e serviços, da gestão do negócio, além da capacitação de pessoas. O projeto integra a academia, as empresas e o governo, garantindo que os conhecimentos gerados sejam aplicados nas empresas, contribuindo para que elas possam competir no mercado mundial.

2º Lugar:

“IACS - Identificação Automática de Componentes de Software”

Entidades Coordenadoras: Laboratório de Inovação– DigitalAssets / Ci&T e Universidade Estadual de Campinas - UNICAMP.

Responsáveis: Marcílio Oliveira, José Vahl, Kleber Bacili.

E-mail: marcilio.oliveira@digitalassets.com.br; jose.vahl@digitalassets.com.br; kleber.bacili@digitalassets.com.br.

Objetivo: O foco central do projeto consiste em montar um mecanismo capaz de reconhecer, em parques de aplicações já desenvolvidas, grupos de artefatos que componham componentes. Visando, sobretudo, a extração destes componentes para catalogação e reuso. Está sendo implementada uma ferramenta que utiliza os mecanismos que identifiquem automaticamente o agrupamento de artefatos relacionados sugerindo ativos de software (componentes, serviços etc.) candidatos à reutilização.

3º Lugar:

“Avaliação de Riscos Aplicada à Qualidade em Desenvolvimento de Software”

Entidades Coordenadoras: Módulo Security e pela Prime Up Soluções em TI

Responsáveis: Alberto Bastos, Gustavo Carvalho, Leandro Daflon, Rafael Espinha e João Condack.

E-mail: abastos@modulo.com.br, gustavo.carvalho@primeup.com.br, daflon@primeup.com.br, rafael.espinha@primeup.com.br

Objetivo: Este projeto gera uma solução para auxiliar empresas contratantes e desenvolvedoras de software, a avaliar, de modo automático e padronizado, pessoas, fornecedores e processos com base nas melhores práticas de engenharia de software (CMMI e MPS.Br). O software contribuirá com a difusão e gestão de conceitos de Qualidade tendo como base Regulamentações e Normas técnicas, otimizando investimentos e reduzindo custos na realização de avaliação de projetos de desenvolvimento de software.

Comissão de Avaliação do Grupo de Projetos do PBQP-Software Ciclo 2007

- ? Alexandre Marcos Lins de Vasconcelos - UFPE
- ? Ana Cervigni Guerra - CenPRA
- ? Ana Liddy Cenni de Castro Magalhães - Consultor
- ? André Henrique de Siqueira - CEF
- ? André Villas-Boas - CPqD
- ? Carlos Henrique R. Cardoso - INATEL
- ? Carlos Lombardi - Consultor
- ? Carlos Mathias Mota Vargas - Consultor
- ? Clênio Figueiredo Salviano - CenPRA
- ? Cristiano Lehrer - ICODES
- ? Cristina Ângela Filipak Machado - CELEPAR
- ? Denise Maria Rodrigues Carneiro - UNIFOR
- ? Edmeia Leonor Pereira de Andrade - Ministério da Agricultura
- ? Edna Pereira Fernandes - Ministério da Fazenda/SRF
- ? José Eduardo Albuquerque Medeiros - MCT/SEPIN
- ? Jucele França de A. Vasconcellos - UFRJ/COPPE
- ? Karina Rodrigues Pereira - C.E.S.A.R
- ? Luiz Carlos de Almeida Oliveira – CELEPAR
- ? Maria de Fátima Mattiello Francisco - INPE
- ? Regina Thienne Colombo - CenPRA
- ? Renata Endriss - SOFTEX/Recife
- ? Salvador Alves de Melo Junior - Consultor
- ? Samy Lima Assi - FabriQ
- ? Sonia Thereza Maintinguer - CPqD
- ? Teresa Maria de Medeiros Maciel - C.E.S.A.R
- ? Teresinha Morais Falabella de Castro - Eletronorte

Vencedores do Ciclo 2006

Dos 99 projetos integrantes desse Ciclo foram relatados 95% dos projetos, sendo que 47 foram concluídos (48%) e encaminhados para avaliação final.

O processo de avaliação deste Ciclo foi finalizado em Brasília/DF, nos dias 14 e 15 de maio de 2007, durante o Workshop da Qualidade e Produtividade em Software - WQPS, obtendo-se a seguinte classificação:

1º Lugar:

“MPS.BR - Melhoria de Processo do Software Brasileiro”

Entidade Coordenadora: Associação para Promoção da Excelência do Software Brasileiro - SOFTEX

Responsáveis: Kival Chaves Weber e Eratóstenes Araújo

E-mail: kival.weber}@nac.softex.br, era@nac.softex.br

Objetivo: Criação de um novo modelo para melhoria de processo de software - composto de um Modelo de Referência (MRMPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS), a ser implantado em um número significativo de empresas de software no Brasil, a um custo acessível (MNC - modelo de negócio cooperado entre várias empresas).

2º Lugar:

“Laboratório de Engenharia de Software e Inteligência Artificial: Construção do ambiente WebAPSEE”

Entidade Coordenadora: Laboratório de Engenharia de Software da Universidade Federal do Pará - UFPA

Responsáveis: Carla Alessandra Lima Reis e Rodrigo Quitês Reis

E-mail: clima@ufpa.br, quites@computer.org

Objetivo: Consolidar pesquisa em Tecnologia do Processo de Software, em especial a experiência na construção do ambiente WebAPSEE, para gerência de processos de software, procurando avançar no estado da arte dessa tecnologia através da integração com técnicas de inteligência artificial.

3º Lugar: (empate)

“RCCS - Rede de Compartilhamento de Componentes de Software”

Entidades Coordenadoras: Laboratório de Inovação em Software - Ci&T Labs / UNICAMP e Instituto de Computação - Universidade Estadual de Campinas

Responsáveis: Kleber Bacili, Ricardo Anido, Marcílio Oliveira

E-mail: krbacili@cit.com.br, ranido@ic.unicamp.br, marcilio@cit.com.br

Objetivo: O principal objetivo deste projeto é a criação de uma rede pública de distribuição e compartilhamento de componentes de software através de uma arquitetura baseada em serviços e da utilização dos padrões Web Services.

“Qualidade do Software Embarcado em Aplicações Espaciais”

Entidades Coordenadoras: Instituto Nacional de Pesquisas Espaciais - INPE e DBA Engenharia de Sistemas Ltda

Responsáveis: Fátima Mattiello Francisco e Ricardo Costa

E-mail: fatima@dss.inpe.br, rcosta@dba.com.br

Objetivo: Transferir para a indústria brasileira do setor de software os conhecimentos adquiridos no INPE no desenvolvimento de software crítico para a área espacial, capacitando a indústria no uso de ambientes e técnicas de validação adotadas na integração e testes dos software embarcados em cargas úteis de missões de satélites científicos.

Comissão de Avaliação do Grupo de Projetos do PBQP-Software Ciclo 2006

- ? Alexandre Marcos Lins de Vasconcelos - UFPE
- ? Ana Cervigni Guerra - CenPRA
- ? Ana Liddy Cenni de Castro Magalhães - Consultor
- ? André Henrique de Siqueira - CEF
- ? André Villas-Boas - CPqD
- ? Carlos Henrique R. Cardoso - INATEL
- ? Carlos Lombardi - Consultor
- ? Carlos Mathias Mota Vargas - Consultor
- ? Clênio Figueiredo Salviano - CenPRA
- ? Cristiano Lehrer - ICODES
- ? Cristina Ângela Filipak Machado - CELEPAR
- ? Denise Maria Rodrigues Carneiro - UNIFOR
- ? Edmeia Leonor Pereira de Andrade - Ministério da Agricultura
- ? Edna Pereira Fernandes - Ministério da Fazenda/SRF
- ? Francisco José Silveira de Vasconcellos - Marinha do Brasil
- ? José Salustiano Fagundes - HIRIX
- ? Jucele França de A. Vasconcellos - UFRJ/COPPE
- ? Karina Rodrigues Pereira - C.E.S.A.R
- ? Luiz Carlos de Almeida Oliveira - CELEPAR
- ? Marcelo Alves de Barros - UFPB
- ? Regina Thienne Colombo - CenPRA
- ? Renata Endriss - SOFTEX/Recife
- ? Salvador Alves de Melo Junior - Consultor
- ? Samy Lima Assi - FabriQ
- ? Sonia Thereza Maintinguer - CPqD
- ? Teresa Maria de Medeiros Maciel - C.E.S.A.R
- ? Teresinha Moraes Falabella de Castro - Eletronorte

Vencedores do Ciclo 2005

Dos 142 projetos integrantes desse Ciclo foram relatados 92% dos projetos, sendo que 50 foram concluídos (35%) e encaminhados para avaliação final 49 projetos cujos artigos foram apresentados.

Este processo de avaliação foi finalizado em Brasília/DF, nos dias 27 e 28 de abril de 2006, durante o Workshop da Qualidade e Produtividade em Software - WQPS.

A seguir, o resultado da classificação final:

1º Lugar:

“Residência em Software”

Entidades Coordenadoras: Quali Software Processes, Universidade Federal de Pernambuco - UFPE e Motorola Industrial Ltda

Responsáveis: Augusto Cezar Alves Sampaio e José Mário Lima

E-mail: acas@cin.ufpe.br, augusto@qualiti.com, jose.lima@motorola.com

Objetivo: Implantar um modelo inovador de formação em Engenharia de Software seguindo a filosofia da Residência Médica. Na Residência em Software o papel dos hospitais é desempenhado pelas fábricas de software e uma instituição de ensino oferece a formação teórica.

O modelo proposto pode ser utilizado tanto como uma formação complementar a cursos de graduação em computação e áreas afins quanto para reciclar profissionais de mercado nas novas tecnologias, conceitos e paradigmas da engenharia de software.

2º Lugar:

“Revista ProQualiti: Qualidade na Produção de Software”

Entidades Coordenadoras: Universidade Federal de Lavras - UFLA e SWQuality Consultoria e Sistemas Ltda

Responsáveis: Ana Cristina Rouiller e Geovane Nogueira Lima

E-mail: acr@comp.ufla.br, geovane@swquality.com.br

Objetivo: Criar uma edição semestral da revista InfoComp especializada em Qualidade de Software. Esta revista, publicada em meio impresso pela Editora da UFLA, deve apresentar a priori dois tipos de trabalhos: artigos científicos e relatos de experiência. É possível a inclusão de artigos de cunho didático, tais como tutoriais e revisões bibliográficas.

3º Lugar:

“PRO2PI: Perfis de Capacidade de Processo para Melhoria de Processo”

Entidades Coordenadoras: Centro de Pesquisas Renato Archer - CenPRA, Universidade Estadual de Campinas/Faculdade de Engenharia Elétrica e de Computação - Unicamp/FEE

Responsáveis: Clênio Figueiredo Salviano e Mario Jino

E-mail: clenio.salviano@cenpra.gov.br, jino@dca.fee.unicamp.br

Objetivo: Consolidar uma metodologia para orientar a melhoria de processo de software, com referência a múltiplos modelos de capacidade de processo, baseada no conceito de perfis de capacidade de processo, que sejam, dinâmicos e específicos. Esta metodologia vem sendo desenvolvida e aplicada nos últimos seis anos e foi recentemente nomeada PRO2PI. Os modelos utilizados têm sido: ISO/IEC 15504, SW-CMM, CMMI-SE/SW, MR-MPS e outros.

Comissão de Avaliação do Grupo de Projetos do PBQP-Software Ciclo 2005

- ? Alexandre Marcos Lins de Vasconcelos - UFPE
- ? Ana Cervigni Guerra - CenPRA
- ? Ana Liddy Cenni de Castro Magalhães - Consultor
- ? André Henrique de Siqueira - CEF
- ? André Villas-Boas - CPqD
- ? Carlos Henrique R. Cardoso - INATEL
- ? Carlos Lombardi - Consultor
- ? Carlos Mathias Mota Vargas - Consultor
- ? Clênio Figueiredo Salviano - CenPRA
- ? Cristiano Lehrer - ICODES
- ? Cristina Ângela Filipak Machado - CELEPAR
- ? Denise Maria Rodrigues Carneiro - UNIFOR
- ? Edmeia Leonor Pereira de Andrade - Ministério da Agricultura
- ? Edna Pereira Fernandes - Ministério da Fazenda/SRF
- ? Francisco José Silveira de Vasconcellos - Marinha do Brasil
- ? José Salustiano Fagundes - SUCESU/DF
- ? Jucele França de A. Vasconcellos - UFRJ/COPPE
- ? Karina Rodrigues Pereira - C.E.S.A.R
- ? Luiz Carlos de Almeida Oliveira - CELEPAR
- ? Paulo Sérgio Bomfim – MCT/SPOA/CGTI
- ? Regina Thienne Colombo - CenPRA
- ? Renata Endriss - SOFTEX/Recife

- ? Salvador Alves de Melo Junior – MCT/SEPIN
- ? Samy Lima Assi - FabriQ
- ? Teresa Maria de Medeiros Maciel - C.E.S.A.R
- ? Teresinha Moraes Falabella de Castro - Eletronorte

Vencedores do Ciclo 2004

Dos 75 projetos integrantes desse Ciclo foram relatados 100% dos projetos, sendo que 42 foram concluídos (56%) e encaminhados para avaliação final 41 projetos cujos artigos foram apresentados.

Este processo de avaliação foi finalizado em Brasília/DF, nos dias 28 e 29 de abril de 2005, durante o Workshop da Qualidade e Produtividade em Software - WQPS.

A seguir, o resultado da classificação final:

1º Lugar:

“Documentação Essencial: Um Enfoque na Documentação Necessária para Manutenção de Software”

Entidade Coordenadora: Universidade Católica de Brasília - UCB

Responsáveis: Sérgio Cozzetti Bertoldi de Souza, Nicolas Anquetil e Káthia Marçal de Oliveira

E-mail: sergio.cozzetti@triall.com.br, anquetil@ucb.br, kathia@ucb.br

Objetivo: Propor um método de desenvolvimento de software com documentação essencial, necessária para manutenção, e que não comprometa a entrega do software para o cliente.

2º Lugar:

"Consolidação da Metodologia para Avaliação de Processos de Software de MPes Baseada na Norma ISO/IEC 15504 (SPICE)"

Entidades Coordenadoras: Universidade do Vale do Itajaí - UNIVALI e Centro de Pesquisas Renato Archer - CenPRA

Responsáveis: Christiane Gresse von Wangenheim e Clenio F. Salviano

E-mail: gresse@sj.univali.br, clenio.salviano@cenpra.gov.br

Objetivo: Consolidar uma metodologia de avaliação de processos de software com foco na ISO/IEC 15504 (SPICE) para o contexto de micro e pequenas empresas (MPes) de software.

3º Lugar (empate):

"Livro: Aquisição de Produtos e Serviços de Software"

Entidade Coordenadora: Centro de Pesquisas Renato Archer - CenPRA

Responsáveis: Ana Cervigni Guerra e Ângela Maria Alves

E-mail: ana.guerra@cenpra.gov.br, angela.alves@cenpra.gov.br

Objetivo: Elaborar publicação para apresentação dos problemas clássicos e tendências para processos de aquisição de produtos e serviços de software. Apresenta também um conjunto de atividades fundamentais para as empresas que necessitam contratar, para os contratantes, clientes e compradores de produtos e serviços de software.

3º Lugar (empate):

"Qualidade de Software na Região Norte"

Entidade Coordenadora: Centro de Incubação e Desenvolvimento Empresarial - CIDE

Responsável: Francisco José Silveira de Vasconcellos

E-mail: vasconcellos@netmentor.info

Objetivo: Estimular a divulgação de indicadores setoriais da qualidade e produtividade no setor de software como instrumentos de conscientização e motivação.

De acordo com procedimentos previamente estabelecidos pelo Grupo de Projetos, no Ciclo 2004 a exemplo do Ciclo 2003, adicionalmente aos melhores projetos premiados foram reconhecidos os melhores projetos por Categoria, resultando no reconhecimento na categoria **Articulação Institucional** para:

“O AmazonSoft como Núcleo SOTEX do Amazonas”

Coordenação: Centro de Análise, Pesquisa e Inovação Tecnológica – FUCAPI, Antonio Luiz da Silva Maués

Comissão de Avaliação do Grupo de Projetos do PBQP-Software Ciclo 2004

- ? Alexandre Marcos Lins de Vasconcelos - QualiTi
- ? Ana Cervigni Guerra - CenPRA

- ? André Henrique de Siqueira - CEF
- ? André Villas-Boas - CPqD
- ? Carlos Henrique R. Cardoso - INATEL
- ? Carlos Lombardi - Consultor
- ? Carlos Mathias Mota Vargas - CBM
- ? Clênio Figueiredo Salviano - CenPRA
- ? Cristiano Lehrer - IESGO
- ? Cristina Ângela Filipak Machado - CELEPAR
- ? Denise Maria Rodrigues Carneiro - INSOFT
- ? Edmeia Leonor Pereira de Andrade - EMBRAPA
- ? Edna Pereira Fernandes - Ministério da Fazenda/SRF
- ? Francisco José Silveira de Vasconcellos - Marinha do Brasil
- ? Giovano Marcos Mazetto - Consultor
- ? José Carlos De Luca - ASSESPRO
- ? José Salustiano Fagundes - SUCEsu/DF
- ? Jucele França de A. Vasconcellos - FUCAPI
- ? Káthia Marçal de Oliveira - UCB
- ? Luiz Carlos de Almeida Oliveira - CELEPAR
- ? Marcos Nobre - SOFTEX
- ? Mauro de Mesquita Spinola - USP/POLI
- ? Paulo Sérgio Bomfim - MCT/CGMI
- ? Regina Thienne Colombo - CenPRA
- ? Renata Endriss - C.E.S.A.R
- ? Salvador Alves de Melo Junior - MCT/SEPIN
- ? Samy Lima Assi - FabriQ
- ? Teresa Maria de Medeiros Maciel - C.E.S.A.R
- ? Teresinha Morais Falabella de Castro - Eletronorte

Vencedores do Ciclo 2003

Dos 82 projetos integrantes desse Ciclo foram relatados 99% dos projetos, sendo que 26 foram concluídos (32%).

O processo de avaliação deste Ciclo foi finalizado em Brasília/DF, nos dias 13 e 14 de maio de 2004, durante o Workshop da Qualidade e Produtividade em Software - WQPS, obtendo-se a seguinte classificação:

1º Lugar:

“Adaptação do MEDE-PROS para o Processo de Pré-Qualificação do PNAFM”

Entidade Coordenadora: Centro de Pesquisas Renato Archer - CenPRA

Responsáveis: Ana Cervigni Guerra e Sônia Thereza Maintinguer

E-mail: ana.guerra@cenpra.gov.br, soniam@cpqd.com.br

Objetivo: Elaborar uma proposta genérica de um método de avaliação especialista para produtos de software baseado na especificação de requisitos presente em editais e no MEDE -PROS. O programa PNAFM foi considerado como motivador da proposta.

2º Lugar:

“Curso de Especialização Lato-Sensu à Distância em Melhoria de Processo de Software”

Entidade Coordenadora: Universidade Federal de Lavras - UFLA

Responsável: Ana Cristina Rouiller

E-mail: acr@comp.ufla.br

Objetivo: Aprofundar o conhecimento de graduados em Ciência da Computação (ou áreas afins) em Qualidade de Software, com especial enfoque na melhoria dos processos e produtos de software; formar recursos humanos de forma a atender às atuais exigências de qualificação do mercado de trabalho; disseminar os conceitos e aplicabilidade da garantia da qualidade de software para todo o país, estreitando o relacionamento entre indústria de software e centros de pesquisa e normatização; e auxiliar a indústria de software nacional, através de uma pesquisa de levantamento dos processos de qualidade da indústria nacional.

3º Lugar:

“CORDIS-FBC Integrado a um Sistema de Gerência do Conhecimento Cardioknowledge”

Entidades Coordenadoras: Coordenação de Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro - UFRJ/COPPE e Fundação Bahiana de Cardiologia - FBC

Responsável: Álvaro Rabelo Jr., Ana Regina Rocha e Karina Villela

E-mail: arabelojr@terra.com.br, darocha@centroin.com.br, kvillela@uol.com.br

Objetivo: Construir um ambiente de desenvolvimento de software orientado a uma organização específica. O ambiente CORDIS-FBC está portanto orientado ao domínio de Cardiologia, é baseado nos processos de software especificamente definidos para a Fundação Bahiana de Cardiologia e utiliza o seu conhecimento organizacional em desenvolvimento de software, por incorporar princípios de gestão do conhecimento. CORDIS-FBC será integrado ao ambiente de gerência do conhecimento CardioKnowledge.

De acordo com procedimentos previamente estabelecidos pelo Grupo de Projetos, no Ciclo 2003, adicionalmente aos melhores projetos premiados foram reconhecidos os melhores projetos por Categoria, resultando no reconhecimento na categoria **Serviços Tecnológicos** para:

“Elementos Fundamentais para a Melhoria da Qualidade de Software nas Organizações de TI”

Entidade Coordenadora: Centro de Pesquisas Renato Archer - CenPRA

Responsáveis: Ana Cervigni Guerra e Eduardo Paulo de Souza

E-mail: ana.guerra@cenpra.gov.br, eduardos@cpqd.com.br

Vencedores do Ciclo 2002

Dos 68 projetos integrantes desse Ciclo foram relatados 94% dos projetos, sendo que 36 foram concluídos (53%).

O processo de avaliação deste Ciclo foi finalizado em Brasília/DF, nos dias 07 a 09 de maio de 2003, durante o Workshop da Qualidade e Produtividade em Software - WQPS, obtendo-se a seguinte classificação:

1º Lugar:

“Simpósio Brasileiro de Qualidade de Software”

Entidades Coordenadoras: Universidade Federal do Espírito Santo - UFES e Universidade Católica de Brasília - UCB

Responsáveis: Ricardo de Almeida Falbo, Kátia Marçal de Oliveira

E-mail: falbo@inf.ufes.br, kathia@ucb.br

Objetivo: Realizar do 1º Simpósio Brasileiro de Qualidade de Software (SBQS). O SBQS é a consagração do Workshop de Qualidade de Software realizado nos últimos oito anos no Brasil associado ao Simpósio Brasileiro de Engenharia de Software.

2º Lugar:

“Workshop on Software Quality”

Entidades Coordenadoras: Coordenação de Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro - UFRJ/COPPE e Universidade Católica de Brasília - UCB

Responsáveis: Kátia Marçal de Oliveira, Ana Regina Rocha, Kival Chaves Weber

E-mail: kathia@ucb.br, darocha@cos.ufrj.br, kival@polo.com.br

Objetivo: Realizar o Workshop de Qualidade de Software integrado ao International Conference on Software Engineering. O Workshop foi realizado em Orlando - USA no dia 25 de maio de 2002 e contou com a presença de pesquisadores de renome internacional na área de qualidade de software.

3º Lugar:

“SIMPROS 2002: IV Simpósio Internacional de Melhoria de Processo de Software”

Entidades Coordenadoras: Centro de Pesquisas Renato Archer - CenPRA, Faculdade SENAC de Ciências Exatas e Tecnologia e Centro de Estudos e Sistemas Avançados do Recife - C.E.S.A.R

Responsável: Clenio Figueiredo Salviano, Ozeas V. Santana Filho e Teresa Maciel

E-mail: clenio.salviano@cenpra.gov.br, ozeas@sp.senac.br, teresa@cesar.org.br

Objetivo: Realizar o SIMPROS 2002 para disseminar as principais tendências para melhoria de processo de software no cenário nacional e internacional, dando continuidade aos cinco eventos já realizados com sucesso nos anos de 1995, 1996, 1998, 1999, 2000 e 2001. As três primeiras edições do evento tiveram o nome de Semana de Engenharia de Software e a partir da quarta edição foi adotado o nome SIMPROS. O SIMPROS 2002 foi organizado de forma conjunta pelo CenPRA, Faculdade SENAC de Ciências Exatas e Tecnologia e C.E.S.A.R.

Destaques "hors-concours" - Ciclo 2002:

“Qualidade e Produtividade no Setor de Software Brasileiro – 2001”

Entidade Coordenadora: Ministério da Ciência e Tecnologia/Secretaria de Política de Informática - MCT/SEPIN

Responsáveis: Célia Joseli do Nascimento e Diva da Silva Marinho

“Produtividade Sistêmica no Setor de Software Brasileiro – 2001”

Entidades Coordenadoras: Ministério da Ciência e Tecnologia/Secretaria de Política de Informática - MCT/SEPIN e Instituto Brasileiro da Qualidade e Produtividade do Paraná - IBQP-PR

Responsáveis: Célia Joseli do Nascimento e Cristiane de Almeida

“Normalização de Software no Brasil em 2002”

Entidade Coordenadora: Associação Brasileira de Normas Técnicas / Subcomitê de Software - ABNT/CB-21

Responsável: Kival Chaves Weber

Vencedores do Ciclo 2001

Dos 103 projetos integrantes desse Ciclo, foram relatados 72 projetos (70%), sendo que 28 foram concluídos (27%).

O processo de avaliação deste Ciclo foi finalizado em Brasília/DF, nos dias 11 a 12 de março de 2002, durante o Workshop da Qualidade e Produtividade em Software - WQPS, obtendo-se a seguinte classificação:

1º Lugar:

“Metodologias Qualiti - Aumentando a Competitividade da Indústria Nacional de Software”

Entidade Coordenadora: Qualiti Software Processes (Unidade de Negócios do Centro de Estudos e Sistemas Avançados do Recife - C.E.S.A.R)

Responsáveis: Hermano Perrelli de Moura e Alexandre Marcos Lins de Vasconcelos

E-mail: hermano@qualiti.com.br

Objetivo: Fornecer suporte ao desenvolvimento de uma solução de automação comercial (o P2K) que permitirá ao varejo responder aos desafios de um ambiente de negócios, cada vez mais exigente e competitivo. O projeto foi considerado relevante por permitir a capacitação de uma fábrica de software da Comércio Soluções Inteligentes (CSI) com cerca de 40 pessoas, apoiar o desenvolvimento de um software de grande porte totalmente OO usando uma metodologia, e pela sua contribuição para a concepção e criação da Qualiti (uma Unidade de Negócios do C.E.S.A.R).

2º Lugar:

“Qualidade em Serviços apoiada por Sistemas de Informação”

Entidades Coordenadoras: Coordenação de Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro - UFRJ/COPPE e Fundação Bahiana de Cardiologia - FBC

Responsável: Alvaro Rabelo Jr. e Ana Regina da Rocha

E-mail: arabelo@ufba.br, darocha@cos.ufrj.br

3º Lugar:

“Normalização de Software no Brasil”

Entidade Coordenadora: Associação Brasileira de Normas Técnicas/ Subcomitê de Software/Comissão de Estudos “Avaliação de Processos de Software” - ABNT/CB-21/SC-10 - CE-21:101.04

Responsável: Kival Chaves Weber

E-mail: abnt_sw@pr.gov.br

Vencedores do Ciclo 2000

Dos 86 projetos integrantes desse Ciclo, foram relatados 79 projetos (92%), sendo que 35 foram concluídos (41%).

O processo de avaliação deste Ciclo foi finalizado em Brasília/DF, nos dias 05 e 06 de março de 2001, durante o Workshop da Qualidade e Produtividade em Software - WQPS, obtendo-se a seguinte classificação:

1º Lugar:

“Livro Qualidade de Software: Teoria e Prática”

Entidades Coordenadoras: Coordenação de Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro - UFRJ/COPPE e Universidade de São Paulo - USP

Responsáveis: Ana Regina Rocha, José Carlos Maldonado, Kival Weber

Objetivo: Publicar um livro sobre Qualidade de Software com vários autores e editado por: Ana Regina Rocha, José Maldonado e Kival Weber. Foram convidados para autores de capítulos do livro pesquisadores de Universidades e Centros de Pesquisa bem como profissionais de empresas do país que atuam na área. O livro teve seus capítulos organizados de forma a atender aos temas da ISO 12207.

2º Lugar:

“Qualidade e Produtividade no Setor de Software Brasileiro – 1999”

Entidade Coordenadora: Ministério da Ciência e Tecnologia/Secretaria de Política de Informática - MCT/SEPIN

Responsáveis: Célia Joseli do Nascimento / Diva da Silva Marinho

3º Lugar:

“Qualidade de Software Médico”

Entidades Coordenadoras: Coordenação de Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro - UFRJ/COPPE e Fundação Bahiana de Cardiologia - FBC

Responsáveis: Dr. Alvaro Rabelo Jr. e Ana Regina Rocha

Vencedores do Ciclo 1999

Dos 79 projetos integrantes desse Ciclo, foram relatados 50 projetos (63%), sendo que 36 foram concluídos (46%).

O processo de avaliação deste Ciclo foi finalizado em Brasília/DF, nos dias 27 e 28 de março de 2000, durante o Workshop da Qualidade e Produtividade em Software - WQPS, obtendo-se a seguinte classificação:

1º Lugar:

“ISSES’99 4th IEEE – International Software Engineering Standards Symposium”

Entidade Coordenadora: Centro Internacional de Tecnologia de Software - CITS, Curitiba/PR

Responsável: Roberto Antônio Rodrigues de Almeida

Objetivo: Propiciar ao público conhecimentos sobre as melhores práticas, ferramentas e padrões de engenharia de software utilizadas no mundo. Conquistou o Prêmio Dorgival Brandão Junior da Qualidade e Produtividade de 1999.

2º Lugar:

“Programa Paraná Classe Mundial em Software”

Entidade Coordenadora: Centro Internacional de Tecnologia de Software - CITS, Curitiba/PR

Responsável: Roberto Antônio Rodrigues de Almeida

3º Lugar:

“Criação de Núcleos de Qualidade de Software”

Entidade Coordenadora: Fundação Centro Tecnológico para Informática - CTI, hoje CenPRA, Campinas/SP

Responsável: Romildo Monte

Vencedores do Ciclo 1998

Dos 84 projetos integrantes desse Ciclo, 70 (83%) foram relatados, sendo que 41 (49%) foram concluídos.

1º Lugar:

“Mestrado em Design de Produtos de Informática com Área de Concentração em Design de Software”

Entidades Coordenadoras: Departamento de Sistemas e Computação da Universidade Federal da Paraíba - UFPB e Núcleo SOFTEX de Campina Grande - CGSOFT

Responsável: Marcelo Alves de Barros

2º Lugar:

“Criação de Núcleos de Qualidade de Software”

Entidade Coordenadora: Fundação Centro Tecnológico para Informática - CTI, hoje CenPRA, Campinas/SP

Responsável: Romildo Monte

3º Lugar:

“Conscientização e Motivação de Jovens Empreendedores de Software para Melhoria da Qualidade e Produtividade”

Entidade Coordenadora: Sociedade Brasileira para Promoção da Exportação de Software - SOFTEX, Campinas/SP

Responsável: Eratóstenes Edson Ramalho de Araújo

Vencedores do Ciclo 1997

Dos 57 projetos integrantes desse Ciclo, 36 foram concluídos (63%), com a seguinte classificação final:

1º Lugar:

“Normalização de Software no Brasil”

Entidade Coordenadora: Associação Brasileira de Normas Técnicas/ Subcomitê de Software - ABNT/CB-21/SC-21:10, Curitiba/PR

Responsável: Kival Chaves Weber

2º Lugar:

“Conscientização e Motivação de Jovens Empreendedores de Software para a Melhoria da Qualidade e Produtividade”

Entidade Coordenadora: Sociedade Brasileira para Promoção da Exportação de Software - SOFTEX, Campinas/SP

Responsável: Eratóstenes Edson Ramalho de Araújo

3º Lugar:

“Avaliação de Melhoria da Qualidade de Software”

Entidade Coordenadora: Companhia de Processamento de Dados do Paraná CELEPAR, hoje Companhia de Informática do Paraná, Curitiba/PR

Responsável: Cristina Filipak

Vencedores do Ciclo 1996

Dos 42 projetos integrantes desse Ciclo, 25 projetos (60%) foram realizados e relatados durante WQPS Brasília realizado em 03 e 04 de março de 1997.

1º Lugar:

“Qualidade no Setor de Software Brasileiro”

Entidade Coordenadora: Ministério da Ciência e Tecnologia/Secretaria de Política de Informática - MCT/SEPIN, Brasília/DF

Responsáveis: Célia Joseli do Nascimento, Diva da Silva Marinho, Bernardo Estellita Lins

2º Lugar:

“Normalização de Software no Brasil”

Entidade Coordenadora: Associação Brasileira de Normas Técnicas/ Subcomitê de Software - ABNT/CB-21/SC-21:10, Curitiba/PR

Responsável: Kival Chaves Weber

3º Lugar:

“Testes de Software para Melhoria da Qualidade”

Entidades Coordenadoras: Centro de Pesquisa e Desenvolvimento em Telecomunicações da Telebrás - CPqD, Universidade Estadual de Campinas - Unicamp e Universidade de São Paulo - USP São Carlos

Responsável: André Villas-Boas

Vencedores do Ciclo 1995

Dos 37 projetos integrantes desse Ciclo, 30 projetos (81%) foram realizados.

1º Lugar:

“Rumo à ISO 9000, parte II: A Busca da Certificação”

Entidade Coordenadora: Centro Internacional de Tecnologia de Software - CITS, Curitiba/PR

Responsável: Kival Chaves Weber

Como resultado deste projeto, no primeiro semestre de 1996, três empresas paranaenses de software (AUSLAND, HOTSOFT e POLO) foram certificadas ISO 9001/TickIT e uma (KERNEL) foi certificada ISO 9002/TickIT, após avaliação da empresa certificadora inglesa BVQI (Bureau Veritas Quality International).

2º Lugar:

“Melhoria da Qualidade do Processo de Marketing de Software nas Empresas Brasileiras de Software”

Entidade Coordenadora: Sociedade Brasileira para Promoção da Exportação de Software - SOFTEX, Campinas/SP

Responsável: Sérgio O. A. Andrade

Destaque:

“Normalização de Software no Brasil”

Entidade Coordenadora: Associação Brasileira de Normas Técnicas/ Subcomitê de Software - ABNT/CB-21/SC-21:10, Curitiba/PR

Responsável: Kival Chaves Weber

Vencedores do Ciclo 1994

Dos 24 projetos integrantes desse Ciclo, 11 projetos (46%) foram realizados.

1º Lugar:

“Avaliação de Software”

Entidade Coordenadora: Fundação Centro Tecnológico para Informática - CTI, hoje CenPRA, Campinas/SP

Objetivo: Gerar e disseminar procedimentos para avaliação da qualidade de software, a partir do estabelecimento de um conjunto evolutivo de critérios.

2º Lugar:

“Rumo à ISO 9000-3”

Entidade Coordenadora: Centro Internacional de Tecnologia de Software - CITS, Curitiba/PR

Objetivo: Desenvolver projeto-piloto de implantação de sistema da qualidade em dez empresas paranaenses que desenvolvem software, com base na Norma Internacional ISO 9000-3

Destaques:

- ? Primeira edição do livro “Qualidade e Produtividade em Software”, Centro Internacional de Tecnologia de Software - CITS, Curitiba/PR
- ? Consolidação do Subcomitê de Software, Associação Brasileira de Normas Técnicas/ Subcomitê de Software - ABNT/CB-21/SC-21:10, Curitiba/PR
- ? Instalação de laboratórios para desenvolvimento e difusão de métodos, ferramentas e procedimentos avançados de Engenharia de Software em treze cidades, Sociedade Brasileira para Promoção da Exportação de Software - SOFTEX, Campinas/SP