Ministério da Ciência e Tecnologia
Secretaria de Política de Informática

PBQP Software

Programa Brasileiro da Qualidade e Produtividade em Software

Projetos Ciclos 2008 e 2009



1ª edição
Janeiro/2010

Presidente da República
Luiz Inácio Lula da Silva

Ministro da Ciência e Tecnologia
Sergio Machado Rezende

Secretário de Política de Informática
Augusto Cesar Gadelha Vieira

Diretora de Políticas e Programas Setoriais em Tecnologias da Informação
Marylin Peixoto da Silva Nogueira

Coordenador Geral de Serviços e Programas de Computador
Antenor Cesar Vanderlei Corrêa

Compilação, organização e revisão
Diva da Silva Marinho

Apoio Técnico
Euripedes José de Sousa

Ministério da Ciência e Tecnologia
Secretaria de Política de Informática

Revista do Programa Brasileiro da Qualidade e Produtividade em Software

1ª edição
Maio/2010



Editorial

Estamos diante de mais uma publicação que o Ministério da Ciência e Tecnologia – MCT, por intermédio da Secretaria de Política de Informática – SEPIN, realiza no tema 'qualidade' no setor de Software e Serviços de Tecnologia da Informação. Trata-se da primeira edição da Revista do Programa Brasileiro da Qualidade e Produtividade em Software – PBQP-Software, que reúne os artigos relativos aos projetos concluídos apresentados em 2008 e 2009. Tais artigos foram submetidos à avaliação do PBQP-Software no âmbito dos ciclos anuais de projetos, cujo objetivo é engajar instituições e empresas do setor software e serviços de tecnologia da informação na promoção da melhoria de processos, produtos e serviços correlatos. Aos três melhores projetos de cada ciclo são concedidos o Prêmio Dorgival Brandão Junior da Qualidade e Produtividade em Software.

Esse tema tem sido objeto de ações sistemáticas das políticas públicas para a área de Tecnologia da Informação – TI desenvolvidas no Brasil nas últimas décadas, dentre as quais se destacam as medidas tanto no âmbito do Plano de Ação de Ciência Tecnologia e Inovação – PACTI, de novembro de 2007, quanto da Política de Desenvolvimento da Produção – PDP, de maio de 2008, que visam estimular a adoção de melhores práticas de engenharia de software pelas empresas brasileiras como um dos principais instrumentos para elevar a competitividade e a capacidade produtiva do setor.

O PBQP-Software é um programa mobilizador, que procura estimular a adoção de normas, métodos, técnicas e ferramentas da qualidade e da engenharia de software, com vista a promover a melhoria da qualidade dos processos, produtos e serviços de software brasileiros e, assim, tornar as empresas nacionais mais capacitadas a competir no mercado globalizado.

Os 24 artigos reunidos nessa primeira edição da revista demonstram a capacidade e a pujança da comunidade que se dedica a esse tema no Brasil. Esperamos, também, que representem um estímulo para que novas contribuições sejam aportadas aos próximos ciclos de projetos que serão promovidos pelo PBQP-Software.

Antenor Corrêa

Coordenador-Geral de Serviços e Programas de Computar/SEPIN-MCT

Sumário

Artigos dos Projetos 2008

Incentivando a Disseminação e o Desenvolvimento de Núcleos SPINs (Software and Systems ProcessImprovement Network) pelo Brasil
Ana Liddy Cenni de Castro Magalhães, Laudecy Fabiani de Oliveira Alves,César de Ávila Rodrigues _ 15

Projeto Cooperado MPS.BR - ACATE 2007/2008
Demetrius Ribeiro Lima, Marcello Thiry, Christiane G. von Wangenheim, Alessandra Zoucas, Elisete Robazza _____ 21

Grupo de Melhoria de Processos de Software do CTIC-UFPA
Carla Alessandra Lima Reis, Antonio J. Gomes Abelém, Rodrigo Quites Reis_____ 27

Procedimento de Avaliação de Qualidade de Serviços
Carlos Constantino Moreira Nassur _____ 32

Um Framework para Avaliação da Implantação do MR MPS em Grupos Cooperados de Empresas
Rafael Prikladnicki _____ 42

Estratégia para a Implementação do Modelo CMMI ou MPS-BR na DATAPREV.
Rosana Fernandes Osório _____ 47

FAPS - Ferramenta de Suporte à Avaliação de Processos de Software Alinhada com os Modelos CMMI e MPS.BR
Marcello Thiry, C. G. Gresse von Wangenheim, Alessandra Zoucas, Leonardo Reis Tristão, José Luiz da Silveira _____ 54

Ambiente da Qualidade de Produto de Software
Regina Thienne, Ana Guerra, Marbilia Sérgio, Márcia Pimenta _____ 59

Aplicação da Teoria das Restrições na Gestão de Desenvolvimento de Software
Lauro César Araujo, Américo Ferreira da Silva Neto_____ 66

Modelo de Processo de Aquisição de Software e Serviços Correlatos da FUNCEF
Nirian Martins S. Santos, Luciana Roldan de Almeida, Lania V. A. Lacorte _____ 70

FROISPI - Framework Return on Investment of Software Process Improvement
Paulino W. P. Viana, Alexandre M. L. de Vasconcelos _____ 75

Gestão Quantitativa de Pessoas em Desenvolvimento Distribuído de Software
Rafael Prikladnicki _____ 80

Artigos dos Projetos 2009

Processo de Contratação de Software e Serviços Correlatos para Entes Governamentais
Cláudio Silva da Cruz, Edméia Leonor Pereira de Andrade, Rejane Maria da Costa Figueiredo ____ 87

PCP-DEF - Método para Definição de Perfis Alvo para Melhoria de Processo em MPEs de Software
Cristiano Schwening, Marcello Thiry _____ 94

Um Modelo de Capacidade para Desenvolvimento Distribuído de Software em Ambientes de Internal Offshoring
Rafael Prikladnicki _____ 100

Construção de um Portal de Conhecimentos de Engenharia de Requisitos
Judith Pavón , Erika do Nascimento , Sidney Viana _____ 106

FireScrum: Ferramenta Open-Source
Eric Cavalcanti, Teresa M. de Medeiros Maciel, Jones Albuquerque _____ 110

Implantação de Gerência do Conhecimento no CTIC-UFPA com o Ambiente WebAPSEE
Jadielly F. Oliveira, Gabriela F. Andrade, Lorena C. Tavares, Antonio J. G. Abelém, Rodrigo Quites Reis, Carla A. Lima Reis_____ 116

Padrões e Processos para Arquitetura de Software do Governo do Estado do Pará: uma Parceria PRODEPA e UFPA

Cláudio R. de L. Martins, Adailton M. Lima, Carla A. Lima Reis, Rodrigo Quites Reis _____ 122

Representação do Conhecimento Gerado no Processo de Desenvolvimento de Software através de Storytelling

Leia F. Assis, Carlos A M Pietrobon _____ 129

Sensedia Dashboard

Rafael Navarro, José Vahl, Marcílio Oliveira _____ 138

Uma Ferramenta para Ter Conhecimento sobre Riscos

Felipe Muller Diehl, Marcelo Hideki Yamaguti _____ 148

Verificação dos Processos de um Modelo e de Desenvolvimento com o MPS.BR

Priscila Souza _____ 152

WDP-RT: Uma Técnica de Leitura para Inspeção de Usabilidade de Aplicações Web

Marcos Gomes, Tayana Conte _____ 159

Prêmio Dorgival Brandão Júnior _____ 167

Artigos dos Projetos Concluídos Ciclo 2008

Incentivando a Disseminação e o Desenvolvimento de Núcleos SPINs (*Software and Systems ProcessImprovement Network*) pelo Brasil

Ana Liddy Cenni de Castro Magalhães1, Laudecy Fabiani de Oliveira Alves 1, César de Ávila Rodrigues1

1SPIN-BH – www.spin-bh.com.br – Belo Horizonte – MG – Brasil

analiddy@gmail.com, laudecy.fabiani@gmail.com, cesar.avila@gmail.com

Abstract. *One of the main difficulties reported by the coordinators of SPINs –nodes of the Software and Systems Process Improvement Network –, as well asby people interested in establishing new ones on their region has been the lackof information and the poor available structure. This paper reports theprocess framework that has been developed, formalized and disseminated bySPIN-BH, aiming at favoring the creation of new SPINs, supporting emergentSPINs on its establishment and encouraging the exchange of ideas,information and mutual support among all Brazilian SPINs.*

Resumo. *Uma das principais dificuldades relatadas pelos coordenadores de SPINs – núcleos que compõem a rede SPIN (Software and Systems Process Improvement Network) –, bem como pelos interessados em criá-los em sua região tem sido a falta de informação e de estrutura disponível. Este artigo relata o trabalho de estruturação de processos desenvolvido, formalizado e que está sendo disseminado pelo SPIN-BH, visando facilitar a criação de novos SPINs, apoiar SPINs emergentes em sua estruturação e incentivar a troca de experiências, informação e suporte mútuo entre SPINs brasileiros.*

1. Introdução

A progressiva atenuação de fronteiras políticas, tarifárias e comerciais, decorrentes da globalização, bem como a constante entrada de novas organizações, acirrou a disputa pelo mercado de software e aumentou a expectativa por melhores produtos.

Paralelamente, o surgimento da internet facilitou a disseminação de comunidades de prática, que buscam a obtenção de um benefício coletivo pelo compartilhamento de experiências, habilidades e conhecimentos, possibilitando potencializar ações. Com o objetivo de melhorar a qualidade dos sistemas de defesa norteamericanos dependentes de software, foi criado nos EUA o SEI (*Software Engineering Institute*), responsável pelo desenvolvimento de diversos modelos que se tornaram referência mundial, entre eles o CMMI. Lançando mão de comunidades de prática e visando promover a disseminação de seus modelos e o aperfeiçoamento das práticas de engenharia e processos de software pelo mundo, o SEI incentivou a formação e a propagação da rede SPIN (*Software and Systems Process Improvement Network*), constituída de diversos núcleos SPINs – nós desta rede espalhados pelo mundo.

Em busca de uma maior participação no mercado, tanto brasileiro quanto global, a melhoria de processos e o fornecimento de produtos de alta qualidade têm sido, cada vez mais, objetivo estratégico das empresas brasileiras de software. Despontam iniciativas para capacitação em engenharia e qualidade de software, bem como para a promoção internacional do software brasileiro. Apesar de sua relevância no cenário atual, a qualidade de software é um tema recente e ainda pouco explorado nos currículos

universitários brasileiros. Visando capacitar profissionais para atuarem nesta área, cursos específicos de pós-graduação *Lato* e *Stricto Sensu* foram criados, bem como consultorias especializadas passaram a fornecer treinamentos, geralmente embutidos em contratos maiores de implantação de programas de melhoria. Paralelamente a estas iniciativas, as comunidades de prática – como os grupos de discussão e os núcleos SPINs – começaram a fornecer um suporte complementar, como fórum para troca de idéias e experiências reais de organizações, em geral sem custo para os interessados.

Procurando despertar e promover iniciativas locais de melhoria de processos e de produtos de software em organizações brasileiras, o movimento SPIN chegou ao Brasil em 1997, com a formação do primeiro núcleo SPIN brasileiro – o SPIN São Paulo [SPIN-SP 2008]. Decorridos mais de dez anos, surgiram mais 17 núcleos SPINs espalhados por 12 estados brasileiros [LatinSPIN 2006]. Boa parte destes, porém,

encontra-se ainda em fase de estruturação.

O SPIN-BH foi criado como uma das atividades do projeto "Minas - Rede de Qualidade", submetido ao Ciclo de Projetos 2003 do PBQP-Software [Faria 2004]. Com o aprendizado obtido ao longo dos anos, tornou-se necessário reestruturá-lo, de onde surgiu a idéia de formalizar um processo e compartilhar experiências com outros SPINs.

Este artigo relata o trabalho de re-estruturação interna e de formalização de processos que foi desenvolvido pelo SPIN-BH, visando melhor atender a sua comunidade. Esta Introdução apresentou uma contextualização e a motivação para o desenvolvimento deste projeto. A Seção 2 apresenta os objetivos gerais e a justificativa para sua realização. A Seção 3 explica como o trabalho foi desenvolvido, abordando aspectos relevantes de

formação e estrutura do SPIN-BH. O conjunto de processos definidos pelo SPIN-BH é apresentado na Seção 4. A seguir, as Seções 5 e 6 abordam a aplicabilidade dos resultados obtidos e características inovadoras deste projeto. Finalmente, a Seção 7 tece algumas conclusões sobre o projeto e apresenta propostas para trabalhos futuros.

2. Objetivos e Justificativa

Em reunião dos coordenadores de SPINs Latino-Americanos realizada durante a Conferência SEPG LA (*Latin American Software Engineering Process Groups Conference*) em 2006 [LatinSPIN 2006], uma das principais dificuldades relatadas foi o informalismo e a falta de estrutura para o funcionamento de um SPIN. Os núcleos SPINs incentivam a melhoria de processos e precisam, portanto, dar o exemplo de sua correta aplicação. Poucos são os SPINs bem estruturados e em constante atuação.

Muitos acabam desistindo ou "hibernando" antes mesmo de se estruturarem. Neste sentido, este projeto possui dois objetivos principais:

- Desenvolver e disponibilizar processos que facilitem a propagação e a estruturação de SPINs pelo Brasil, fortalecendo a rede existente, apoiando a disseminação da cultura SPIN e estimulando o desenvolvimento de software com qualidade;
- Aplicar a melhoria no próprio SPIN-BH, para atuar de forma mais dinâmica e profissional, aprimorar a realização de seus eventos (palestras, painéis, grupos de estudo e treinamentos), organizar o conhecimento adquirido e os dados existentes.

3. Metodologia de Execução

A estratégia de desenvolvimento do projeto incluiu as seguintes etapas: levantamento de requisitos para a definição do que deveria estar contemplado em um processo para organização e operação de SPINs e para a realização de eventos, realizada por meio de *brainstorm* entre os então coordenadores do SPIN-BH; definição inicial do processo, incluindo descrições, diagramas e *templates*, utilizando recursos básicos de edição de texto e diagramação; aplicação desta definição na organização de eventos pilotos, seguida de ajustes na definição inicial; formalização do processo utilizando uma ferramenta para criação e descrição de processos de desenvolvimento de software customizados – o *Eclipse Process Framework Composer*; disponibilização do processo resultante no site do SPIN-BH para toda a comunidade; divulgação para os demais SPINs brasileiros.

4. Resultados Obtidos

Para melhor compreensão do processo resultante, faz-se necessário um entendimento mais amplo de como surge e funciona um núcleo SPIN, seguida de explicações gerais sobre o SPIN-BH. Após este entendimento, é apresentada uma visão geral do processo, seguida de um detalhamento dos principais componentes do processo.

4.1. A Estrutura da Rede SPIN

A rede SPIN (*Software and Systems Process Improvement Network*) surgiu em 1988 como uma iniciativa voluntária vinculada ao SEI (*Software Engineering Institute*). Sua idéia era criar um mecanismo pelo qual profissionais da área, interagindo com os SEPGs (*Software and Systems Engineering Process Groups*), pudessem trocar experiências, compartilhar conhecimentos e fornecer suporte mútuo [SEI 2008]. A propagação desta iniciativa fomentou a inovação na prática da engenharia de software pelo mundo, gerando núcleos SPINs espalhados pelos cinco continentes [SEI 2008a]. Seus coordenadores geralmente se reúnem nas Conferências SEPG e em suas derivações, como a *Latin American SEPG* (SEPG LA), *European SEPG* e *SEPG Australia*, todas co-patrocinadas pelo SEI.

Cada núcleo da rede SPIN integra profissionais de uma região interessados em promover o aperfeiçoamento das práticas de engenharia de software e sistemas e trocar experiências. Eles promovem reuniões periódicas e abertas a todos os interessados na melhoria de processos em geral. Os núcleos SPINs são completamente independentes e autônomos: o SEI não exerce nenhum controle sobre eles, porém possibilita e incentiva sua comunicação e integração. Alguns SPINs operam com o trabalho voluntário de pessoas, outros possuem o apoio de alguma organização patrocinadora. Para se manter, alguns cobram taxas simbólicas de associação ou de participação em eventos. Em função de seu porte e da comunidade regional, a estrutura do SPIN pode variar de um ou dois coordenadores, até a existência de comitês ligados a questões específicas comoeducação, treinamento, métricas e qualidade de software [SEI 2008b].

A maioria dos SPINs utiliza a internet e seus serviços como principal meio de divulgação do núcleo e de seus encontros. A freqüência de reuniões varia em função da demanda e estrutura existentes. Nestas reuniões são realizadas apresentações técnicas de especialistas e pessoas envolvidas em áreas relacionadas a processos de software, que geralmente apresentam informações práticas baseadas em experiências do mundo real, propiciando discussões sobre as melhores práticas, estudos de casos e lições aprendidas.

4.2. A Estrutura do SPIN-BH

O SPIN-BH é formado por profissionais da indústria, comércio, governo e academia ligados à melhoria do processo de software, situados na região metropolitana de Belo Horizonte/MG. Opera com tempo e recursos voluntários, sem nenhuma organização patrocinadora. Esta foi uma opção de seus coordenadores, como forma de prover um fórum para troca livre e aberta de experiências e idéias, sem envolver interesses comerciais e de marketing. A associação ao núcleo, bem como a participação nos eventos é gratuita e aberta a toda a comunidade. Seus eventos discutem a melhoria de processos em geral, atendendo aos anseios da comunidade local e à disseminação do uso dos modelos MPS.BR e CMMI e de outras normas relacionadas à qualidade de software, entre elas ISO 9001, ISO/IEC 12207, ISO/IEC15504, além de metodologias e técnicas relacionadas a gerenciamento, desenvolvimento e teste de software. Dentre as atividades desenvolvidas, destacam-se a organização de eventos como palestras, painéis e treinamentos, além da divulgação de informações e novidades sobre o desenvolvimento prático e pesquisas relacionadas à melhoria de processos de software.

4.3. O Processo Desenvolvido pelo SPIN-BH

O processo apresentado a seguir foi inicialmente criado para uso pelo próprio SPIN-BH, em busca de um funcionamento mais dinâmico e profissional, visando melhor organizar o conhecimento adquirido e os dados existentes, bem como aprimorar a realização de seus eventos. Percebendo a melhoria obtida, surgiu a idéia de disponibilizar o resultado para outros SPINs, de forma a facilitar a propagação e a estruturação de SPINs pelo Brasil, fortalecer a rede existente, apoiar a disseminação da cultura nesta área e estimular o desenvolvimento da qualidade e produtividade do software brasileiro.

4.3.1. Visão Geral do Processo

Apresentado na Figura 1, o processo definido tem como objetivo mostrar o caminho a ser percorrido para a criação, estruturação, controle e manutenção de um SPIN, bem como para a realização de seus eventos. O primeiro passo é a criação propriamente dita do SPIN, em que será necessário realizar seu registro junto ao SEI. Após a criação, deve-se estruturar o SPIN, para que o núcleo possa atuar de forma mais efetiva. Esta estruturação engloba a definição de regras de funcionamento, papéis e responsabilidades pela execução dos trabalhos, bem como a criação de *templates* para os controles a serem realizados. Com o núcleo SPIN estabelecido e estruturado, devem-se planejar as atividades que serão desenvolvidas para a comunidade em um determinado período. A partir deste planejamento, cada evento deve ser preparado, executado e concluído, e as demais atividades acompanhadas. Além disso, tudo o que foi criado na estruturação do SPIN deve ser mantido atualizado. Todos estes sub-processos foram detalhados em tarefas, descritas por seu propósito, papéis envolvidos, entradas, saídas e passos. Para a execução destas tarefas foram definidos os seguintes papéis:

Coordenadores Geral, de Captação, de Conteúdo, de Infraestrutura e de Eventos; Organizador do Evento e Colaborador(es). Mais de um papel pode ser acumulado por uma mesma pessoa, o que significa que a estrutura de um núcleo SPIN não precisa ser grande. A prática tem mostrado que, a menos dos Colaboradores (que são vários), é preferível que um mesmo papel não seja executado por mais de uma pessoa.

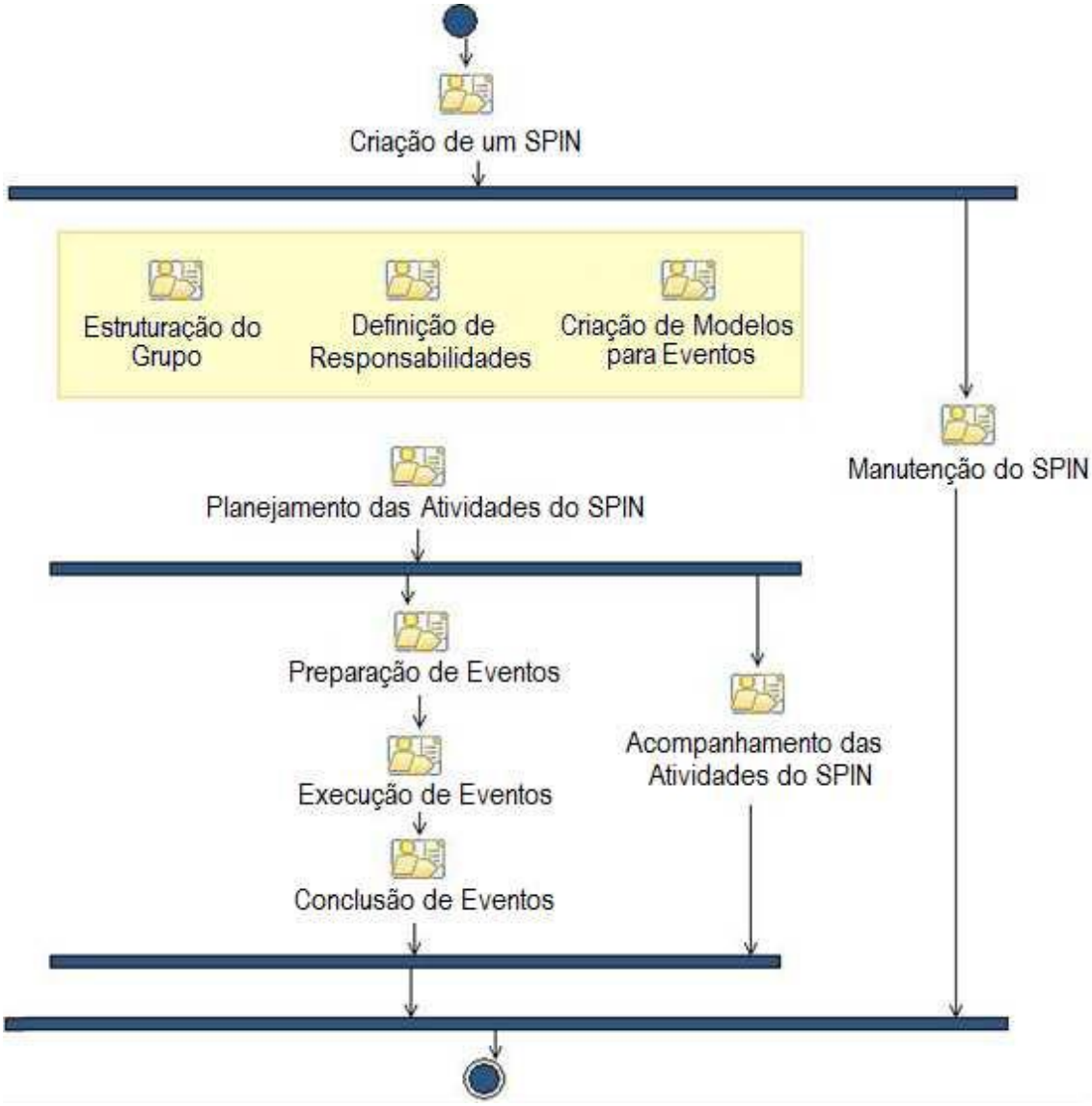


Figura 1. Visão macro do processo desenvolvido

4.3.2. Descrição Geral dos Sub-Processos

Criação de um SPIN

Para a criação de um núcleo SPIN, é necessário que exista um grupo de pessoas interessadas em assuntos relacionados à melhoria de processos na região e dispostas a assumir a tarefa de montar e conduzir o núcleo adiante. Estas pessoas devem reunir-se para discutir e definir a missão do núcleo (o porquê dele estar sendo criado), bem como outras questões básicas sobre seu funcionamento, entre elas: se ele será tratado de forma voluntária e informal, ou regido por regras e financeiramente sustentado; como e por quem será administrado; como serão captados os recursos e fundos para os encontros; qual a freqüência dos eventos; como ocorrerá a divulgação de seus encontros e realizações; que papéis e responsabilidades deverão existir no núcleo; estratégias gerais para sua estruturação, divulgação e sustentação. O núcleo pode manter-se mais informal ou definir um regimento interno, tornando-se mais formal. Após estas definições iniciais, o núcleo deve realizar contato para sua oficialização junto ao SEI.

Estruturação Geral do SPIN

Mesmo que possua uma abordagem mais informal, torna-se necessária uma estruturação mínima das atividades, papéis e responsabilidades para que o núcleo funcione de maneira efetiva a partir das ações dos voluntários. Um Grupo de Coordenadores deverá dar andamento às atividades do SPIN. Sugere-se a criação de uma lista de discussão para o SPIN, como forma de disseminar a melhoria de processos e interagir com a comunidade. Se possível, também deve ser criado um *web site*, o que facilita divulgá-lo, disponibilizar material e automatizar atividades, como inscrições em eventos. O núcleo deve criar também um repositório para manter toda a sua documentação gerencial, com informações sobre sugestão de temas a serem abordados, possíveis locais, palestrantes e patrocinadores para os eventos. Deve ainda elaborar modelos diversos, como: convite, orientações e agradecimento a palestrantes; solicitação de patrocínio; documentação de eventos (listas de inscritos/ presença/ verificação, fichas de avaliação e histórico).

Planejamento das Atividades do SPIN

As atividades do SPIN são contínuas e dependem do trabalho voluntário dos envolvidos.

Assim, é recomendável que haja um planejamento dessas atividades para que os eventos possam acontecer a contento. Este planejamento deve prever a realização de eventos ao longo de um período (ex: anual). O trabalho envolvido na realização de um evento pode ser estruturado como um projeto, contendo atividades, responsáveis, entregas e datas previstas. Cadastros de apoio, contendo sugestão de temas a serem abordados, possíveis locais, palestrantes e patrocinadores, ajudam na elaboração deste planejamento macro de eventos, a ser confirmado e detalhado posteriormente para cada evento.

Realização de Eventos

A realização de eventos é dividida em três etapas distintas: preparação do evento (préevento), incluindo a confirmação de patrocinador, palestrantes, local e programação, bem como a divulgação e controle de inscrições; a realização do evento propriamente dita, que engloba as providências de última hora e execução conforme a programação estabelecida; o fechamento do evento (pós-evento), que formaliza os agradecimentos a patrocinadores e palestrantes, disponibiliza material para interessados, avalia seu resultado e registra sugestões e lições aprendidas. O uso de modelos de comunicação e listas de verificação possibilita garantir um padrão para os eventos, bem como lembrar detalhes necessários para a sua realização.

Acompanhamento das Atividades do SPIN

Além da realização de eventos, o planejamento realizado deve ser acompanhado de forma macro, incluindo também o acompanhamento da atuação dos coordenadores, tratamento de problemas e questões ocorridas, coleta de lições aprendidas relacionadas a organização de eventos, palestrantes, patrocinadores e locais.

Manutenção do SPIN

Deve-se manter uma estrutura suficiente para o funcionamento adequado do núcleo.

Para isso, é necessário pelo menos: administrar a lista de discussão, moderando associados e mensagens, bem como postando notícias relevantes da área; administrar o *web-site* (se existente), atualizando material e dados de eventos, bem como habilitando/ desabilitando inscrições *on-line* para eventos; manter o repositório e os controles existentes; rever a adequação dos modelos desenvolvidos. Quando aplicável, controlar recursos financeiros, participar de pesquisas e eventos do SEI e representar o SPIN em outros eventos da área.

4.3.3. Principais Produtos de Trabalho Gerados

O processo descreve todos os produtos de trabalho gerados, entre os quais destacam-se:

- Regimento Interno – possui as principais definições sobre a constituição do núcleo SPIN, entre elas: missão, estrutura básica, papéis e responsabilidades, estratégias gerais de sustentação, divulgação e administração do núcleo;
- Listas de Discussão da Coordenação e da Comunidade – possibilitam gerenciar a troca de informações entre coordenadores, bem como divulgar eventos e comunicados, discutir temas relevantes e compartilhar experiências, disseminando a cultura de melhoria de processos entre os participantes;
- *Web-site* – possibilita marcar a presença do grupo na internet e permite incluir funcionalidades adicionais, que facilitam a organização de eventos;
- Material de apoio – cadastros (coordenadores/ colaboradores/ locais/ palestrantes/ patrocinadores), modelos de documentos (apresentação padrão do núcleo SPIN, agenda, convite, e-mails para patrocinadores e palestrantes) e controles diversos (financeiro, organização de evento, histórico de eventos), que visam manter uma estrutura de funcionamento adequada, propiciando uma melhor gerência;
- Repositório – objetiva armazenar e controlar o acesso aos documentos gerenciais e demais materiais do SPIN. Ele pode ser criado no *web-site* ou na lista de discussão;
- Plano de Trabalho Periódico – contém a programação básica de eventos para um determinado período. Define tipos de evento, temas específicos, datas previstas e uma programação preliminar para cada evento.
- Listas de Verificação – contêm a relação de cuidados a serem tomados durante a preparação, execução e conclusão de eventos.
- Ficha de Avaliação dos Eventos – fornece um *feedback* dos participantes e possibilita capturar novas oportunidades de patrocinadores, palestras, palestrantes e locais, além de temas de interesse, críticas e sugestões, visando à melhoria contínua.

5. Aplicabilidade dos Resultados

Os núcleos SPINs são difusores da cultura em melhoria de processos em geral. Melhor estruturá-los implica em um movimento ainda maior em prol da qualidade do software brasileiro. Nos principais centros produtores de software do Brasil temos SPINs, portanto o impacto poderá ser grande: inúmeras organizações que participam de seus eventos poderão beneficiar-se de núcleos SPINs mais atuantes e melhor estruturados.

Os SPINs possuem abrangência multiinstitucional na região em que atuam, e seus dirigentes são voluntários. Com a disponibilização e divulgação do processo gerado para todos os SPINs, pretende-se conseguir um alcance nacional, uma vez que existem atualmente no Brasil 18 SPINs distribuídos em 12 estados, sendo a maioria ainda emergente. Existe também a possibilidade de sua disseminação irrestrita para os SPINs no exterior, desde que o material gerado seja traduzido para outros idiomas.

6. Características Inovadoras

Não se tem conhecimento de nenhuma iniciativa no Brasil cujo objetivo principal tenha sido apoiar o surgimento e a consolidação de SPINs. Atuando desde agosto/2003, o SPIN-BH já passou por várias fases e situações comuns a outros SPINs, tendo mostrado anteriormente detalhes de seu funcionamento no SIMPROS 2005 [Magalhães 2005].

Fora esta apresentação, realizada pelo próprio SPIN-BH, não se tem notícia de que algum SPIN tenha desenvolvido, publicado ou disponibilizado um trabalho semelhante.

7. Conclusão e Perspectivas Futuras

A melhoria de processos de software e sistemas é uma área relativamente nova, que tem crescido rapidamente. Diversas são as organizações que buscam profissionais e informação para iniciar um programa de melhoria. O apoio de comunidades de prática neste recente mercado, como os núcleos SPINs, tem possibilitado às organizações uma interação rica e constante, tanto de forma presencial quanto à distância, com uma relação custo-benefício extremamente favorável.

Este projeto se propôs a realizar um trabalho de re-estruturação interna do SPINBH e de formalização de processos, visando melhor atender a sua comunidade. A divulgação e a disponibilização de seus resultados, que ainda está em andamento, têm como objetivo facilitar a criação de novos SPINs, apoiar a estruturação dos SPINs emergentes e incentivar a troca de experiências com SPINs estruturados, repassando para estes a experiência adquirida. É sabido que cada SPIN tem uma maneira própria de atuação e que existem SPINs mais estruturados que o SPIN-BH, porém seu intuito é trazer à tona a discussão e compartilhar a experiência vivenciada até o momento.

Os autores gostariam de agradecer aos colegas Nicole Delucca Linhares Rodrigues, Cintya Campos Corgosinho Suzuki e Junilson Pereira Souza pelo inestimável apoio durante a realização deste trabalho.

8. Referências Bibliográficas

[Faria 2004] Faria, F. (2004) “Minas – Rede da Qualidade” http://www.mct.gov.br/ upd_blob/3329.pdf

[LatinSPIN 2006] Latin SPIN (2006) “SEPG-LA BoF”. http://www.latinspin.org/docs/ bofreport2006.pdf

[Magalhães 2005] Magalhães, A. L., Parreiras, F., Faria, F. (2005) “SPIN BH: Desafios e Contribuições na consolidação de uma comunidade de prática para melhoria de processos de software” http://www.simpros.com.br/simpros2004/Apresentacoes _PDF/Apresentacoes/Comunidade_Melhoria_Processo_Software.pdf

[SEI 2008] Software Engineering Institute (2007) “SPIN Brochure” http://www.sei.cmu. edu/ collaborating/spins/spin.pdf

[SEI 2008a] Software Engineering Institute (2007) “Software and Systems Process Improvement Networks (SPINs)” http:// www.sei.cmu.edu/collaborating/spins/

[SEI 2008b] Software Engineering Institute (2007) “Frequently Asked Questions” http://www.sei.cmu.edu/collaborating/spins/faq.html

[SPIN-SP 2008] SPIN São Paulo (2008) web-site http://www.spinsp.org.br

Projeto Cooperado MPS.BR - ACATE 2007/2008

Demetrius Ribeiro Lima[1], Marcello Thiry [2 3], Christiane G. von Wangenheim[2 3], Alessandra Zoucas[2 3] , Elisete Robazza[1]

[1]Associação Catarinense de Empresas de Tecnologia (ACATE)
Rua Lauro Linhares, 589 -3° Andar – Trindade
CEP 88036-001 - Florianópolis - SC

[2] Incremental Tecnologia em Informática Ltda.
Caixa Postal 7009 – CEP 88025-970 – Florianópolis, SC Brasil

[3]Laboratório de Qualidade e Produtividade de Software (LQPS)
Mestrado em Computação Aplicada (MCA)
Universidade do Vale do Itajaí (UNIVALI) – Campus São José
Rod. SC 407 km 4, s/n – 88.122-000 – São José – SC – Brasil

{dir.software, softpolis}@acate.com.br,
{marcello.thiry, gresse, alessandra.zoucas}@gmail.com

Resumo. Com o objetivo implementar ações de melhoria de processo de software alinhadas ao nível G de maturidade do MPS.BR, cinco empresas associadas a ACATE participaram do Projeto Cooperado MPS.BR – ACATE que foi executado pela II-MPS.BR Incremental Tecnologia. Este artigo apresenta as atividades realizadas no Projeto Cooperado MPS.BR – ACATE, descrevendo seu contexto, bem como objetivos, justificativa, metodologia de execução, resultados obtidos, aplicabilidade dos resultados, conclusões e perspectivas futuras.

1. Introdução

A partir dos resultados da pesquisa do Ministério de Ciência e Tecnologia (MCT) realizou uma pesquisa comparando indicadores de produtividade sistêmica das Micro e Pequenas Empresas (MPEs) de software em relação aos de médias e grandes empresas [MCT, 2001], pôde-se verificar que o desempenho das MPEs fica muito aquém do desempenho das médias e grandes empresas. A escassez de recursos, tanto humano quanto financeiro, e a imaturidade dos processos são características inerentes à maioria das MPEs. Portanto, tais características, somado à falta de conhecimento da existência de modelos de referência, minimizam as possibilidades das MPEs desenvolverem um produto com qualidade e produtividade. Isto pode comprometer seu desenvolvimento, sua competitividade e até mesmo sua sobrevivência.

É por meio da garantia da qualidade do processo produtivo que tipicamente se atinge a melhoria da qualidade do produto final [Maldonado et. Al. 2001]. Desta forma, as empresas brasileiras de desenvolvimento de software estão cada vez mais envolvidas na melhoria dos processos de software, buscando alcançar os padrões internacionais de qualidade e vantagens competitivas no mercado interno e externo.

O interesse das empresas da ACATE em participar deste projeto surgiu a partir da demanda dos clientes que vêem cobrando, por exemplo, melhoria da qualidade dos produtos e cumprimento de prazos de entrega. Outro motivo relevante é a conscientização das empresas tanto em relação à melhoria dos processos de desenvolvimento de software, quanto em relação ao bem estar da equipe de trabalho, buscando diminuir a necessidade de realizar horas extras, bem como minimizar a desmotivação da equipe em função de projetos caóticos.

A oportunidade de participar de um projeto cooperado de melhoria de processo de software, que ainda tenha 50% do seu custo financiado, é uma contribuição relevante para alavancar o setor de software brasileiro, em particular do pólo de Florianópolis.

2. Objetivos e Justificativa

O objetivo do Projeto Cooperado MPS.BR – ACATE foi implementar um programa de melhoria de processo de software alinhado ao modelo MPS.BR [SOFTEX 2006] em um grupo composto por cinco micro, pequenas e médias empresas da Grande Florianópolis, permitindo que estas empresas pudessem ser avaliadas com sucesso no respectivo modelo. O projeto foi coordenado e executado pela Instituição Implementadora MPS.BR Incremental Tecnologia e teve como IOGE (Instituição Organizadora de Grupo de Empresas), a ACATE (Associação Catarinense de Empresas de Tecnologia). A Incremental Tecnologia atua em parceria com LQPS (Laboratório de Qualidade e Produtividade de Software) da UNIVALI (Universidade do Vale do Itajaí), compartilhando pesquisadores e consultores na área de melhoria e avaliação de processo de software.

O projeto buscou também a formação de um grupo de profissionais com conhecimento em modelos, métodos e processos em cada empresa participante, estruturando uma equipe de melhoria responsável pelas ações de melhoria. A Tabela 1 descreve algumas das características destas empresas.

Tabela 1. Características das organizações

	Organização 1	***Organização 2***	***Organização 3***	***Organização 4***	***Organização 5***
Porte	Micro	Pequena	Pequena	Média	Média
Fundação	2004	1992	2001	1990	1990
Área de atuação	Desenvolvimento de plataformas eletrônicas embarcadas.	Desenvolvimento de sistema de pregão eletrônico para mercado empresarial privado e público	Desenvolvimento de sistemas para gestão da informação, gestão do conhecimento e e-learning	Desenvolvimento de sistemas para iformatização de Tribunais, Ministério Público e Procuradorias	Desenvolvimento de sistemas para Construção Civil, Transporte e Obras
Sócios	3	3	1	1	1
Funcionários	2	16	9	95	98
Principais metas	-Expandir a faixa no mercado -honrar prazos estabelecidos - Reduzir tempo de desnvolvimento de sistemas	-Melhorar satisfação dos clientes. - Melhorar satisfação dos cfuncionários -Obter certificaçào CMMI ML2	-Melhorar produtividade -Reduzir prazos de desenvolvimento - Obter certificaçào CMMI ML2	-Expandir a faixa no mercado -Melhorar produtividade e satisfação dos clientes	-Expandir a faixa no mercado -Melhorar produtividade e satisfação dos clientes
Principais Clientes	Telecomunicações, Setor de construção civil, Energia, Comércio	Setor de comércio eletrônico	Setores comerciais e educacionais.	Setor Judiciário, Ministério Público e Procuradorias.	Setor de construção civil, Transporte e Obras
Certificação	Não possui	Profissionais certificados (PMP, Websphere, Lotus Notes, Java)	Não possui	ISO 9000	ISO 9000

A base para a melhoria foi o Modelo de Referência MPS.BR, nível G (parcialmente gerenciado), e as regras definidas conforme COMUNICADO SOFTEX MPS.BR 20/2005, atualizado em 16 de dezembro de 2005. A partir da oficialização do projeto em 30 de abril de 2007, a Equipe de Implementação traçou a estratégia de implementação e iniciou a elaboração do planejamento do projeto

3. Metodologia de Execução

A Incremental vem aprimorando sua metodologia de implementação de melhoria de processo. A metodologia de implementação que a Incremental emprega no projeto cooperado ACATE é composta por sete fases distintas:

- Fase 1.Conscientização
- Fase 2.Diagnóstico do Processo de Software da Empresa
- Fase 3.Planejamento das Melhorias
- Fase 4.Treinamento
- Fase 5.Execução do Plano de Melhorias
- Fase 6.Avaliação Preliminar
- Fase 7.Avaliação Oficial

A Fase 1-Conscientização, iniciou com uma reunião de *kick off* onde as empresas participantes puderam conhecer a proposta do projeto, a equipe da II-MPS.BR Incremental, bem como a metodologia que seria aplicada para a sua execução. Para esta reunião compareceram de um a dois colaboradores de cada empresa participante do projeto.

Durante a Fase 2-Diagnóstico do Processo de Software da Empresa, as empresas participantes tiveram seus processos de software analisados em alto-nível por meio de avaliações de contextualização [Gresse Von Wangenheim et. al. 2005]. As avaliações de contextualização possibilitaram que fossem identificadas características relevantes de cada empresa, tais como tamanho da equipe de desenvolvimento, metas de negócio, estratégias de gerenciamento, execução das áreas de processo referente ao nível G de maturidade do MPS.BR, ferramentas, metodologias e técnicas utilizadas. Estas avaliações também permitiram mapear as diferenças entre a situação atual dos processos de cada empresa e os resultados esperados do nível G de maturidade do MPS.BR. Para isto, primeiramente coletamos informações para estabelecer uma *baseline* inicial da execução dos processos de desenvolvimento de software nas empresas e em seguida comparamos esta *baseline* atual com perfil de capacidade alvo, que é onde se pretende chegar ao final da implementação das melhorias. Esta avaliação foi utilizada para demonstrar, logo de início para as empresas, a evolução da melhoria de processos para as empresas, ou seja, o quanto de benefício elas estariam obtendo com a implementação de melhoria de processos de software.

A avaliação foi realizada individualmente em cada empresa participante do projeto. A análise foi feita por meio de questionários e entrevistas com os responsáveis envolvidos no processo de software (incluindo gerentes, desenvolvedores, etc.). Como resultado do diagnóstico, foram elaborados relatórios individuais para cada empresa.

Esta fase proporcionou a identificação das características e demandas próprias de cada empresa como, por exemplo, clientes públicos, clientes privados, desenvolvimento de software embarcado, produzir software de prateleira ou sob encomenda, etc. Estas diferenças levam às empresas a fazer uso de tecnologias e ferramentas diferentes.

Portanto, o objetivo da Fase 3-Planejamento das Melhorias, foi identificar uma forma de alinhar a execução das fases seguintes de forma cooperada. Para isto planejamos as atividades que poderiam ser executadas de forma cooperada como, por exemplo, a execução dos 3 treinamentos de 8 horas de duração cada um, e planejamos atividades individuais que, por questões de confidencialidade, não poderiam ser executadas de forma cooperada. Desta forma, as atividades individuais foram planejadas para acontecerem em paralelo nas cinco empresas. As atividades individuais incluem os 3 simulados de avaliação de processos e visitas técnicas. O primeiro simulado aconteceu em abril de 2007, o próximo está planejado para acontecer em novembro de 2007 e o último em março de 2008. As visitas técnicas quinzenais e individuais para cada empresa, foram planejadas para ocorrerem com 2 horas de duração. Inicialmente esta duração das visitas técnicas, com intervalo de duas semanas, parece pouco tempo aos olhos das empresas participantes. Porém, cada visita técnica gera uma grande quantidade de esforço em atividades de melhoria que a empresa deve executar no período entre as visitas técnicas.

O objetivo da Fase 4-Treinamento é capacitar os profissionais das empresas participantes para que eles possam identificar e avaliar metodologias de desenvolvimento e manutenção de software que sejam adequadas aos processos do nível G de maturidade do MPS.BR. Os treinamentos de Visão Geral do MPS.BR, Gerência de Projetos e Gerência de Requisitos, foram planejados e executados de forma concentrada, logo no início do projeto. Tivemos pontos positivos utilizando esta abordagem, pois todos os participantes dos treinamentos já foram capacitados na parte teórica desde o início do projeto. Entretanto, enfrentamos problemas em relação à alta rotatividade das equipes das empresas. Como as empresas são de micro e pequeno porte, o número de colaboradores por empresa é muito pequeno. Se um destes colaboradores que participou dos treinamentos se desliga da empresa, o impacto na execução do projeto cooperado é significativo.

A partir das fases da metodologia descritas à cima, ações de melhoria foram identificadas e planejadas. A seguir está descrito como estas ações foram implementadas em projetos piloto, acompanhadas e revisadas, para então serem aplicadas em projeto representativo da unidade organizacional cujos processos serão oficialmente avaliados.

O objetivo da Fase 5-Execução do Plano de Melhorias, é acompanhar as empresas participantes do projeto para a implantação e controle dos resultados esperados pelo nível G de maturidade do MPS.BR. A maioria das empresas participantes do projeto cooperado ACATE começaram a implementação das melhorias a partir do nível 0 de maturidade. Entretanto, há empresas que executavam alguns processos internos, mas estes processos não estavam estabelecidos, apresentavam muitas variações, não estavam completamente registrados ou amplamente disponíveis aos interessados. Portanto, a primeira atividade realizada após o início desta fase, foi a modelagem dos processos atualmente adotados em cada uma das empresas. Para tal, utilizou-se uma abordagem colaborativa, e visual para a modelagem do processo de software, onde são utilizados símbolos específicos que facilitam a visualização e o entendimento sobre o processo [Thiry et. Al. 2006]. Destas seções participaram os implementadores da II-MPS.BR Incremental e dos responsáveis de cada processo discutido. A evolução da modelagem dos processos de software foi registrada por meio de fotos digitais (vide Figura 1). Posteriormente o processo modelado foi validado e documentado pelos próprios responsáveis dos processos discutidos.

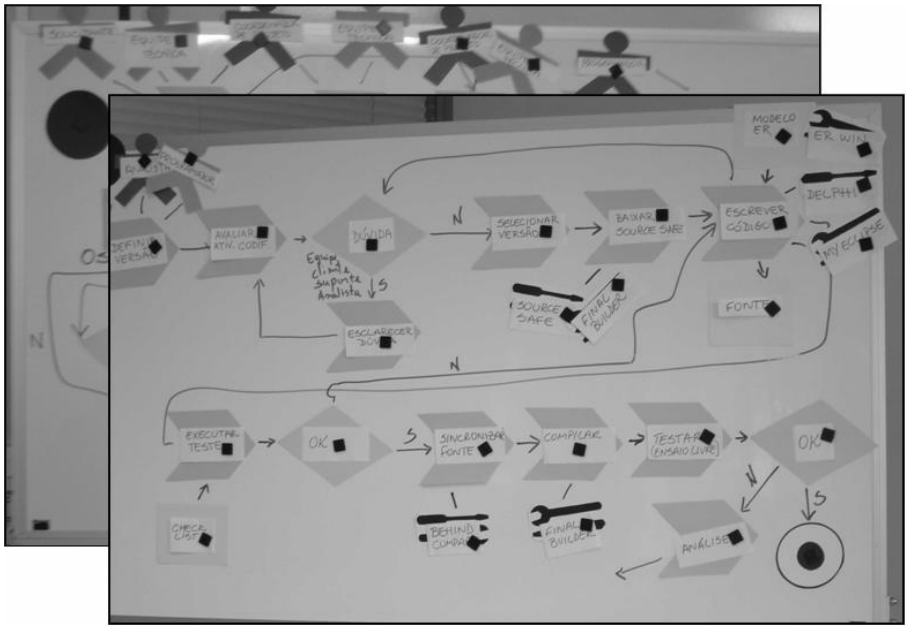


Figura 1. Exemplos de modelagem do processo de software

Visando pormenorizar os processos modelados e validados nas etapas anteriores, foi feita uma descrição detalhada dos mesmos com o auxilio de um template de detalhamento de processos. Este template foi elaborado pela equipe de implementadores de melhoria de processos da Incremental em parceria com o LQPS e disponibilizado para as empresas participantes utilizarem. O template apresenta campos que devem ser preenchidos com detalhes do processo documentado, de forma que seja possível um colaborador sem experiência no processo realizá-lo apenas lendo as informações contidas neste documento.

As atividades da Fase 6-Avaliação Preliminar consistem em uma avaliação geral do processo de desenvolvimento da organização logo após as atividades previstas no Plano de Melhorias terem sido concluídas. O objetivo desta atividade é permitir uma visão da aderência do processo em relação às metas estabelecidas no Plano de Melhoria. Com base nos resultados desta avaliação a empresa avaliará a situação, definindo ações de ajuste para atendimento às metas. Ainda nesta fase a organização deverá estabilizar seu processo de software de forma que possa estar apta para passar pela avaliação oficial conforme o Guia de Avaliação do MPS.BR.

Após a Avaliação Preliminar e dos eventuais ajustes necessários ao processo de desenvolvimento, é iniciada a Fase 7-Avaliação Oficial, onde a organização deverá submeter-se à avaliação oficial conforme Guia de Avaliação do MPS.BR.

4. Resultados Obtidos

O Projeto Cooperado MPS.BR – ACATE foi concluído em julho de 2008 após doze meses de execução e três meses de avaliações. Os três treinamentos planejados para o projeto, Visão Geral do MPS.BR, GPR - Gerência de Projetos e GRE - Gerência de Requisitos, foram realizados por implementadores e avaliadores MPS.BR da Incremental Tecnologia, com média de participação de quatro colaboradores por empresa. Todas as empresas foram contextualizadas, tiveram os processos referentes ao o nível G de maturidade do MPS.BR modelados, detalhados e revisados, e iniciaram projetos representativos da unidade organizacional, cujos processos foram oficialmente avaliados por uma Instituição Avaliadora MPS.BR (IA-MPS.BR).

O apoio financeiro concedido pela Sociedade SOFTEX para a implementação e avaliação das empresas em relação a níveis G e F de maturidade, motivou a decisão de participação das empresas no projeto cooperado. Observamos que isto permitiu o desenvolvimento de uma cultura de melhoria nas empresas participantes fazendo com que elas busquem manter as melhorias já implementadas e se interessarem em posteriormente dar continuidade nas ações de melhoria, como partir para a implementação de níveis de maturidade mais altos.

Um exemplo disto ocorreu no terceiro mês de execução do projeto cooperado ACATE, quando uma das cinco empresas participantes demonstrou interesse em implementar o nível F de maturidade em vez do nível G. Além da pressão do mercado em relação à melhoria da qualidade dos produtos consumidos, o interesse na implementação de um nível de maturidade mais alto veio da própria equipe de trabalho ao notar as melhorias alcançadas durante a execução do projeto cooperado. A Instituição Implementadora percebeu que havia condições da unidade organizacional, onde estava sendo feita a implementação, partir para a implementação do nível F. Portanto, esta empresa recebeu o apoio financeiro concedido pela Sociedade SOFTEX para a implementação e avaliação do nível G de maturidade e a própria empresa cobriu a diferença financeira pertinente à decisão de implementar e avaliar sua unidade organizacional em relação ao nível F de maturidade.

Outro exemplo dos efeitos da melhoria de processo em outra empresa participante do projeto cooperado foi o fechamento de uma parceria internacional, em função da empresa estar participando de um programa de melhoria de processos. Um dos critérios para a parceria ser aprovada era a empresa ter e usar, um processo de gerência de projetos.

Por motivos internos, uma das empresas que buscavam o nível G de maturidade passou por um período em que gerou menos resultados do que havia sido planejado e embora tenha alcançado 50% em relação ao planejado para o Projeto Cooperado ACATE, não conseguiu dar continuidade na utilização do processo durante a execução de seus projetos. Por este motivo a empresa optou por encerrar as atividades planejadas para o Projeto Cooperado ACATE. Portanto, apenas quatro das cinco empresas participantes foram oficialmente avaliadas pela Instituição Avaliadora MPS.BR (IA-MPS.BR) Estratégia. Três destas empresas atingiram o nível G de maturidade e uma empresa alcançou o nível F de maturidade do modelo MPS.BR.

Outro resultado obtido foi o excelente resultado da aplicação do método implementação de melhoria de processos de software alinhado ao modelo MPS.BR desenvolvido pela II-MPS Incremental Tecnologia em parceria com o LQPS Laboratório de Qualidade e Produtividade de Software da UNIVALI Universidade de Vale do Itajaí nas empresas participantes do projeto.

5. Aplicabilidade dos Resultados

A metodologia desenvolvida de implementação de melhoria de processos de software alinhada ao MPS.BR em sua versão atual, é adequada para grupos cooperados próximos da equipe de implementação. Entretanto, para grupos distribuídos, a proposta de trabalharmos com visitas técnicas quinzenais precisaria ser revista, uma vez que o custo de viagens e estadias aumentaria consideravelmente o custo total do projeto. Neste sentido, estão sendo avaliadas alternativas, como workshops mais concentrados, além da utilização de recursos para reuniões remotas.

Uma das dificuldades latentes está na pouca flexibilidade para o planejamento de eventos coletivos. Não apenas pela dificuldade de estabelecer uma agenda comum, mas também pela questão de manter a confidencialidade controlada. Visando manter a confidencialidade dos dados de cada empresa, atualmente, somente a fase de treinamento é feita de forma cooperada. Todas as demais atividades são individualizadas.

Embora as empresas apresentem muitas vezes problemas semelhantes, a forma de resolver é particular para cada empresa, pois cada empresa tem peculiaridades que não permitem uma mesma solução para problemas similares. Porém, é possível agrupar algumas soluções pelas características das empresas, oferecendo um catalisador para futuras tomadas de decisão.

A eficiência das visitas técnicas depende do esforço de melhoria realizado pelas empresas durante o intervalo entre as visitas técnicas. Esta é uma questão importante que causa uma variação na produtividade do grupo.

6. Características Inovadoras

O Projeto Cooperado MPS.BR – ACATE foi a primeira experiência da ACATE em formar grupos de empresas para projeto de implementação e avaliação de melhoria de processos de software.

O método implementação de melhoria de processos de software alinhado ao modelo MPS.BR desenvolvido pela II-MPS Incremental foi aplicado com sucesso nas empresas participantes do projeto. Este método se mostrou adequado à realidade tanto de empresas de pequeno quanto de grande porte, bem como empresas de diferentes linhas de segmento como fábricas de software e empresas que desenvolvem software embarcado.

7. Conclusão e Perspectivas Futuras

O financiamento concedido pela Sociedade SOFTEX para a implementação e avaliação das empresas em relação a níveis G e F de maturidade, motivou a formação do grupo cooperado de empresas do projeto cooperado ACATE.

A aprovação e execução deste projeto favoreceu o desenvolvimento de uma cultura de melhoria nas empresas participantes e hoje, mesmo com o encerramento do projeto, as empresas participantes procuram manter as melhorias já implementadas e se interessarem em dar continuidade nas ações de melhoria, como partir para a implementação de níveis de maturidade mais altos. O sucesso do primeiro Projeto Cooperado repercutiu favoravelmente para a formação de novos grupos de empresas da ACATE que também almejam participar de projetos cooperados para a melhoria de seus processos de software.

A IOGE ACATE pretende submeter novos projetos de implementação de melhoria de processos para a Sociedade SOFTEX avaliar e aprovar apoio financeiro de novos grupos cooperados de empresas catarinenses para a implementação e avaliação de melhoria de processo de software alinhado tanto a níveis iniciais como os níveis G e F quanto a níveis mais altos como os níveis C, D ou E do modelo MPS.BR.

Referências Bibliográficas

Gresse Von Wangenheim, C., Pickler, K., Thiry, M., Zoucas, A., e Salviano, C. (2005) “Aplicando Avaliações de Contextualização em Processos de Software Alinhados ao CMMISE/SW”. In: SIMPROS 2005, Simpósio Internacional de Melhoria de Processo de Software, São Paulo, Brasil, 2005.

Maldonado, J.C; Rocha, A.R.C.; Weber, K.C. (2001) “Qualidade de software: teoria e prática”. São Paulo: Prentice Hall. ISBN 8587918540.

MCT Ministério da Ciência e Tecnologia. (2001) “Qualidade e produtividade no setor de software brasileiro: Resultados da pesquisa 2001”. Disponível em http://www.mct.gov.br/index.php/content/view/4816.html. Acessado em 29/08/2007.

SOFTEX (2006). Guia Geral do MPS.BR - Melhoria de Processo do Software Brasileiro (Versão 1.1). Abril, 2006.

Thiry, M., Gresse Von Wangenheim, C., Zoucas, A. e Pickler, K. (2006) “Uma Abordagem para a Modelagem Colaborativa de Processos de Software em Micro e Pequenas Empresas”. In: V SBQS 2006, Simpósio Brasileiro de Qualidade de Software, Vila Velha, Brasil, 2006.

Grupo de Melhoria de Processos de Software do CTIC-UFPA

Carla Alessandra Lima Reis1, Antonio J. Gomes Abelém2, Rodrigo Quites Reis1

1Laboratório de Engenharia de Software (LABES) 2Instituto de Ciências Exatas e Naturais Centro de Tecnologia da Informação e Comunicação – CTIC-UFPA Universidade Federal do Pará (UFPA)
http://www.labes.ufpa.br

clima@ufpa.br, abelem@ufpa.br quites@ufpa.br

Abstract. *This paper presents the results achieved by the CTIC-UFPA Software Process Improvement Group which was created to combine the studies of a research group on software engineering with the practical experience of the unit devoted to information technology management at the Federal University of Pará in Belém, Brazil. One of the important results was the positive assessment in MPS.BR model considering G level on December, 2008. With this result, CTIC is the first University Information Technology unit in Brazil that has achieved this maturity level. The project aimed to potentialize human resource development and the dissemination of Software Quality Area.*

Resumo. *Este artigo relata os resultados alcançados pelo grupo de Melhoria de Processos de Software do CTIC-UFPA constituído com o objetivo de combinar os estudos de um laboratório de pesquisa com a prática da unidade de gestão de tecnologia da informação da Universidade Federal do Pará. Um dos principais resultados desse projeto foi a avaliação positiva no nível G do MPS.BR em dezembro de 2008. Com isso, o CTIC tornou-se o primeiro setor de TI ligado a Universidade a ser avaliado no MPS.BR. O grupo de melhoria buscou potencializar a formação de recursos humanos atrelados ao projeto e a disseminação da área de Qualidade de Software.*

1. Introdução

O CTIC - CENTRO DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO da Universidade Federal do Pará (UFPA) é o setor responsável pelo desenvolvimento de software da Instituição. Considerando a busca por qualidade de produtos e serviços oferecidos, foi proposta a criação de um grupo de melhoria de processos, coordenado por pesquisadores da área que atuam na UFPA, e que tem como integrantes os funcionários do CTIC e os alunos de graduação e pós-graduação da UFPA. Essa iniciativa teve como foco integrar o LABES-LABORATÓRIO DE ENGENHARIA DE SOFTWARE ao CTIC para auxiliar o CTIC a aumentar sua maturidade em software. Como resultado, foram criadas oportunidades para aumento da interação entre as equipes, maior aprendizado prático dos conhecimentos sobre melhoria de qualidade de software, além de possibilidade de realização de vários estudos de caso acadêmicos, visando produção científica. Em particular, a interação entre os grupos contribuiu para avaliar o desempenho de metodologia e ferramental tecnológico desenvolvidos internamente na UFPA na forma de projetos de pesquisa e dissertações de mestrado, os quais se mostraram úteis para acelerar o processo de implantação de qualidade nos processos de software da organização e foram dignos de nota na avaliação formal para o Nível G do MPS.BR conduzida em dezembro de 2008. A UFPA é atualmente a maior universidade federal do país em número de alunos matriculados. São mais de 32 mil alunos matriculados em nível de graduação em uma instituição que se espalha por nove campi instalados em todas as regiões do estado do Pará, com 188 cursos de graduação ativos. O CTIC, portanto, tem a responsabilidade de fornecer os mecanismos e ferramentas de tecnologia da informação para a Universidade, sendo responsável por operar os pontos de presença de Internet e manter o registro eletrônico de todas as atividades acadêmicas da instituição. Para isso conta com aproximadamente 40 funcionários, sendo que existem 17 colaboradores atuando no setor que desenvolve software. Dentre estes, muitos são bolsistas graduandos da própria UFPA. O projeto de implementação de Melhoria do Processo de Software no CTIC-UFPA (denominado MPS.CTIC) teve início em Março de 2007. As atividades foram inicialmente voltadas para a melhoria da organização e seus processos em geral e não diretamente para um nível em particular do MPS.BR. Alguns projetos de software piloto foram iniciados ainda em 2007. Porém, foi no ano de 2008 que o projeto se consolidou e obteve maior êxito com a Avaliação Oficial MPS.BR nível G. Além disso, o projeto como um todo resultou em várias contribuições importantes para toda a organização, não se restringindo à unidade de desenvolvimento de software, tais como: 1) formação e aperfeiçoamento de recursos humanos em Gestão de Projetos, 2) implantação de processos organizacionais de atendimento ao usuário, 3) homologação de ferramentas de software usadas nos vários setores da empresa, 4) melhora no suporte da área de redes e capacitação. Nas seções a seguir deste artigo são apresentados os resultados alcançados pelo Grupo MPS.CTIC no ano de 2008.

2. Objetivos e Justificativa

A melhoria de processos de software em empresas brasileiras tem ocorrido cada vez mais em consonância com o modelo de Melhoria do Processo de Software Brasileiro (MPS.BR). Com mais de 120 empresas avaliadas desde a sua criação em 2003, o MPS.BR tem cumprido as metas estabelecidas e contribuído com a disseminação da área de qualidade de software e com o uso de ferramentas especializadas. Segundo

Travassos e Kalinowski (2008), os resultados de desempenho de organizações que adotaram o modelo MPS.BR indicam que estas empresas alcançaram maior satisfação dos seus clientes, maior produtividade, capacidade de desenvolver projetos maiores e satisfação com o modelo MPS. Contudo, um número muito pequeno de empresas adotou o modelo na região Norte do Brasil.

Como a iniciativa de melhoria da qualidade de seus processos e produtos, o CTIC-UFPA, em conjunto com o LABES decidiram implantar melhorias alinhadas às perspectivas de aumento de maturidade do modelo MPS.BR. Além de melhorar os processos do CTIC-UFPA, um objetivo importante estava na utilização do ambiente WebAPSEE (Lima Reis e Reis 2007), desenvolvida desde 2004 pelo LABES-UFPA, como ferramenta de apoio a gestão de processos de software1. Adicionalmente, foi estabelecido como objetivo aumentar a quantidade de profissionais aptos a trabalharem com a implementação de MPS.BR e fomentar o desenvolvimento de massa crítica na área na região Norte do país. É importante observar a notória carência da região na área caracterizada pela existência de apenas duas empresas com avaliações publicadas no MPS.BR, e uma empresa avaliada no CMMI2. Portanto, este projeto representou uma oportunidade única para intercâmbio de competências entre um grupo de pesquisa e a unidade gestora de tecnologia da informação de uma importante instituição de ensino superior do país.

3. Metodologia de Execução

Em síntese, a metodologia adotada contemplou os seguintes passos principais:

1. Conscientização das pessoas envolvidas com desenvolvimento de software no CTIC acerca da necessidade da implantação de melhoria de processos através de palestras e treinamentos.

2. Levantamento sobre os processos utilizados no CTIC, buscando conhecer os problemas e também as soluções e boas práticas já adotadas. Foi utilizado como guia de avaliação os disponíveis no modelo MPS.BR e também o guia para avaliações CMMI (Bush e Dunaway, 2005).

3. Proposta de um processo de desenvolvimento padrão que contemplasse as características principais do processo atual da organização e que esteja de acordo com os processos do nível pretendido.

4. Treinamento nos processos do nível G (Gerência de Requisitos e Gerência de Projetos) e no Processo padrão.

5. Implantação de uma ferramenta de gestão de processos e escolha de uma ferramenta para apoiar a gerência de requisitos. Quanto a ferramenta de gestão de processos, foi adotado o ambiente WebAPSEE desenvolvido pelo LABES-UFPA.

6. Instanciação do processo nos projetos pilotos. Os consultores acompanham o projeto piloto, orientando o gerente e outros envolvidos como o processo deve ser executado (*mentoring*).

7. Etapas de verificação, validação do processo, podendo retornar a fases anteriores visando melhoria do processo proposto.

8. Institucionalização do processo, com divulgação dos resultados, treinamentos finais, disponibilização de ferramentas, dentre outras tarefas. [1]

Convém observar que, por questões das restrições de espaço para este relatório, esta é uma simplificação dos procedimentos metodológicos realmente adotados. Um dos diferenciais da proposta se refere a dois aspectos em particular: 1) a busca por construir um modelo de processo realmente alinhado com as características da organização-cliente; 2) a evolução do WebAPSEE para atender as demandas da empresa, buscando alinhar os relatórios produzidos pela ferramenta às evidências necessárias para os processos do MPS.BR. A metodologia adotada no projeto evoluiu para uma metodologia de escopo mais genérico para implantação de melhoria de processos específica para o Ambiente WebAPSEE a qual é descrita com maior detalhe em uma dissertação de Mestrado a ser defendida em março de 2009 (Rocha, 2009). Além disso, apesar do esforço de melhoria ter sido iniciado em março de 2007, a alta rotatividade de gerentes alocados em projetos de software trouxe dificuldades e fez com que a implantação recomeçasse em abril de 2008 com nova equipe e novos projetos.

4. Resultados Obtidos

4.1. Resultados para o CTIC

Os principais resultados obtidos com o esforço de Melhoria de Processos do CTIC-UFPA são listados a seguir:

- **Conscientização**: O processo de melhoria permitiu que os colaboradores do CTIC pudessem observar e entender as necessidades da organização e auxiliou na reflexão sobre as suas próprias necessidades perante o processo. Portanto, foi possível observar o processo de entendimento da necessidade de mudança em relação aos processos do CTIC. Todos foram envolvidos e mesmo setores não diretamente ligados a

[1] 1 O projeto que a produziu foi classificado com o segundo lugar do Prêmio Dorgival Brandão Junior de Produtividade e Qualidade de Software em 2007. 2 No momento da redação deste relatório – fevereiro/2009.

desenvolvimento de software – por exemplo, as áreas de suporte ao usuário e gerência de redes - participaram dos eventos e demandaram consultoria para padronização dos seus processos. Cabe observar também que a convivência constante e o intercâmbio de idéias entre alunos de diferentes níveis (graduação, mestrado e doutorado) do LABES-UFPA, funcionários do CTIC (bolsistas e efetivos), professores, e desenvolvedores do WebAPSEE propiciaram um ambiente de troca de conhecimento e colaboração para resolução dos problemas. No total, participaram do projeto 4 alunos de Iniciação Científica, 7 alunos de Mestrado, e 1 aluno de Doutorado, sendo que destes, 6 possuem formação do Softex como Implementadores do MPS.BR (curso de formação e aprovação na prova);

- **Capacitação**: Vários cursos foram oferecidos à equipe do CTIC e a todos os interessados da UFPA, tais como: 1) Gerência de Requisitos, 2) Definição de Processos de Software, 3) Planejamento de Projetos, 4) Modelagem e Execução de Processos com WEBAPSEE. Além destes, também foi oferecido o Curso oficial de Introdução ao MPS.BR, coordenado localmente pelo CTIC em Belém. Além dos cursos presenciais citados, houve cursos na forma de mentoring sobre 1) Gerência de Projetos, 2) Gerência de Requisitos, e 3) Análise e Projeto orientado a objetos com UML. Todos os cursos e atividades de mentoring foram ministrados pela equipe do LABES-UFPA (alunos e professores). Adicionalmente, 4 colaboradores do CTIC formaram-se Especialistas em Gerência de Projetos em curso oferecido na UFPA (http://egps.ufpa.br) ;
- **Definição do Processo Padrão do CTIC:** O processo de software foi proposto em 2007, testado e melhorado a partir de dois projetos piloto e em 2008 foi institucionalizado nos projetos em execução. Além disso, processos para outras áreas foram do CTIC definidos e implantados, tais como: suporte, capacitação, homologação de ferramentas e atendimento ao usuário. Para apoiar os processos, foram também propostos templates de documentos que guiam as atividades a serem executadas;
- **Sucesso na avaliação oficial,** concluída em 15/12/08, realizada pela SWQuality (Instituição Avaliadora credenciada SOFTEX). A conclusão da avaliação é que a organização atende aos critérios do nível G – Parcialmente Gerenciado do modelo de referência MR-MPS. A equipe de avaliação foi formada por: Ana Liddy Cenni de Castro Magalhães - avaliadora líder e Alexandre Marcos Lins de Vasconcelos – avaliador adjunto, da Instituição Avaliadora (IA) SWQuality Consultoria e Sistemas Ltda; Alline de Melo Lemos e Anderson Jorge Serra da Costa - representante do CTIC/UFPA na equipe de avaliação.
- **Consciência sobre seus pontos fracos:** Um excelente resultado para a Direção do CTIC e da UFPA é o conhecimento sobre seus processos, suas lições aprendidas, seus atuais problemas. Tanto pela avaliação oficial quanto pela atual forma de trabalhar, o CTIC tem maior consciência do seu desempenho e do que precisa fazer para melhorar.

4.2. Artigos publicados

Dois artigos científicos foram publicados em 2008 e um foi submetido em 2009 como Relato de Experiência. Em (Rocha, 2008) é apresentada a metodologia de implementação de MPS.BR adotada no CTIC e desenvolvida especificamente para ser apoiada pelo ambiente WebAPSEE. Por sua vez, Sales (2008a) apresenta uma extensão do Ambiente WebAPSEE para incrementar a Gerência de Configuração de Software com base na experiência e demandas do processo no CTIC. Ambos os trabalhos derivam de dissertações de mestrado com defesas previstas para o início de 2009.

4.3. Recursos humanos capacitados

O grupo MPS.CTIC inspirou diversas iniciativas de formação de recursos humanos bem sucedidas, tais como:

- **Trabalhos de Conclusão de Curso:** 3 TCCs do curso de Bacharelado em Ciência da Computação da UFPA foram orientados no período e contribuíram diretamente para a realização do projeto. São eles:
    - Apoio na Geração de Plano de Projeto no Ambiente WebAPSEE (Lopes, 2008). Módulo desenvolvido para geração automática do plano de projeto em arquivo de editor de texto a partir de informações do sistema.
    - Gerência de Requisitos Integrada à Gerência De Projetos no Ambiente WebAPSEE (Sales, 2008b). Estudo sobre avaliação e adoção de ferramentas para gestão de requisitos com base na experiência dos projetos conduzidos no CTIC.
    - Auditoria e Contabilidade de Gerência de Configuração de Artefatos de Software no Ambiente WebAPSEE (Silva, 2008). Trabalho voltado ao aperfeiçoamento do suporte fornecido pelo WebAPSEE para atender demandas mais elevadas de controle de artefatos.
- **Dissertações de Mestrado:** Além das dissertações concluídas citadas na seção 4.2, outras estão em andamento com aplicação no CTIC:
    - Marcelo Almeida Silva. Uma Abordagem para a Gerência de Pessoas em Projetos de Desenvolvimento de Software.
    - Jadielly Oliveira. Gestão de conhecimento integrada a Gerencia de Processos de Software.

- Gabriela Fonseca Andrade. Estudo Experimental sobre a Implantação de Gerência de Conhecimento em Organizações de Software.

- **Especialistas em Gerência de Projetos de Software:** Participaram do curso de Especialização lato sensu em Gerência de Projetos de Software - também promovido pelo LABES-UFPA - quatro colaboradores do CTIC. Seus trabalhos de conclusão foram alinhados ao esforço da melhoria de processo. São eles:
  - o Figueiredo, Elaine; Nogueira, Carlos. Um Estudo sobre Rastreabilidade de Requisitos e sua Influência na Gerência de Projetos de Software.
  - o Câmara, Valéria. Implantação de Sistemas de Informação com Foco na Integração dos Usuários.
  - o Rubeni, Thiago. Estudo sobre a Implementação da Análise de Pontos por Função no Centro de Tecnologia da Informação e Comunicação da Universidade Federal do Pará.
- **Formação Oficial MPS.BR:** No ano de 2008 a maioria (onze, no total) dos integrantes do LABES envolvidos com o grupo MPS.CTIC obteve sucesso na formação oficial de Implementador MPS.BR. Neste contexto, surgiu uma spin off do LABES - denominada QR Consultoria - que está localizada na Incubadora de Empresas da UFPA e que se tornou a primeira instituição implementadora do MPS.BR credenciada pela SOFTEX no Norte do país.

4.4 Outros Resultados

Um esforço paralelo de melhoria da ferramenta WebAPSEE, aliado ao *feedback* obtido com seu uso na prática em uma empresa que visou avaliação do MPS.BR, permitiu que ao final de 2008 fosse entregue uma versão mais estável e consolidada da ferramenta. O WebAPSEE pôde ser testado e melhorado com grande intensidade para apoiar processos de software visando os níveis de maturidade. Houve diversas melhorias no ambiente e foi possível perceber sua utilidade no processo de Gerência de Projetos do MPS.BR. Na avaliação oficial houve destaque para essa ferramenta, conforme citado pela Avaliadora Líder Ana Liddy: *"O CTIC estava muito bem preparado para a avaliação. Além de possuir diversos profissionais com bom conhecimento em engenharia de software e melhoria de processos, a implementação de Gerência de Projetos e de Gerência de Requisitos foi realizada com esmero,* ***utilizando um ambiente de desenvolvimento de software centrado em processo bem interessante****. Parabéns a todos os envolvidos por esta grande conquista!"* (grifo nosso).

5. Aplicabilidade dos Resultados

Os resultados obtidos com a execução do projeto não podem ser tomados apenas como resultados de uma implementação de melhoria de processos. Antes do projeto não havia equipe de implementação oficial MPS.BR no Norte do país e por isso nenhuma organização havia obtido sucesso em avaliação de MPS.BR com consultoria de grupo local. Vários aspectos importantes do projeto podem ser citados para demonstrar a abrangência e os impactos positivos gerados:

- A associação dessa iniciativa com projetos de pesquisa em andamento potencializou a formação de recursos humanos. Foram envolvidos desde estudantes de graduação até doutorandos e está disponível um espaço para experimentação, o que pode melhorar a qualidade científica do que se produz no grupo de pesquisa. Trata-se de uma evolução significativa visto que, embora localizado no mesmo campus, os grupos de pesquisa estiveram sempre distantes da prática do CTIC.

- Os processos implantados no CTIC não visaram tão somente o nível G, mas a estruturação do CTIC como um todo. Foi detectada uma característica importante do CTIC que aumentaria os riscos do projeto: a alta rotatividade dos seus bolsistas e funcionários. Essa condição traz diversos problemas advindos da falta de gerência de conhecimento. A implantação do ambiente WebAPSEE, que centraliza os processos e artefatos em um repositório eletrônico, tende a auxiliar na transferência de conhecimento entre as equipes.

- O LABES passou a executar projetos de desenvolvimento de software para entidades externas à UFPA e com isso adotou a mesma metodologia sendo aplicada no CTIC. Os integrantes do LABES que não estavam diretamente envolvidos com o projeto do CTIC estão tendo a oportunidade de seguir um processo padrão e buscar melhoria desse processo e do produto.

- Ressalta-se a disseminação da importância da melhoria dos processos, que pode ser percebida pela significativa participação da comunidade local em eventos da área, como por exemplo o EQPS - Encontro de Qualidade e Produtividade de Software, ocorrido em Belém em novembro de 2008, no qual se inscreveram mais de 200 pessoas. Além disso, 5 empresas locais interessaram-se em participar do grupo de empresas para implantação do MPS.BR no início de 2009.

- Pode-se dizer que o projeto contribuiu para a consolidação do MPS.BR na região. Além do aumento no número de empresas avaliadas (de 1 para 2) na Região, ainda houve o credenciamento de uma Instituição Implementadora, apta a atender as demais empresas da Região. Ambas as conquistas, apesar de discretas perante resultados de outras regiões, foram realizadas por um único grupo que atuou no projeto MPS.CTIC.

6. Características Inovadoras

A principal inovação do projeto está na articulação e na consolidação de um grupo através da experiência de implementação de melhoria de processo em instituição de ensino. A execução do projeto foi feita por pesquisadores da área (alunos e professores) e aplicando ferramentas resultantes da pesquisa (ambiente WebAPSEE), sem haver nenhuma experiência local anterior nesses moldes. Essa estratégia permitiu extrair dessa experiência novos modelos para implantação de melhoria da qualidade de software e ferramentas adequadas para essa finalidade.

7. Conclusão e Perspectivas Futuras

O projeto de implementação de Melhoria do Processo de Software no CTIC-UFPA (MPS.CTIC) iniciou em Março de 2007. As atividades foram voltadas para a melhoria da organização e seus processos em geral e não diretamente para um nível em particular do MPS.BR. Alguns projetos piloto foram iniciados ainda em 2007. Em junho de 2008, com a consolidação do processo de desenvolvimento sendo implantado, foram iniciados os projetos que seriam avaliados. A avaliação ocorreu nos meses de novembro e dezembro de 2008 e teve resultado positivo. O projeto como um todo resultou em várias contribuições importantes para a organização que não ficaram restritas à área de desenvolvimento de sistemas de software. Por fim, se deve ressaltar que a cooperação entre uma unidade de pesquisa e outra de produção dentro de uma Universidade alcançou resultados importantes e há a perspectiva de avançar nesta cooperação em busca de níveis mais altos de maturidade.

8. Referências Bibliográficas

Bush, M.; Dunaway, D. CMMI Assessments: Motivating Positive Change. Addison Wesley, 2005.

Lima Reis, Carla A.; Reis, Rodrigo Q. (2007) “Laboratório de Engenharia de Software e Inteligência Artificial: Construção do ambiente WebAPSEE”. In: ProQuality (UFLA), v. 3, p. 43-48.

LOPES, PATRICIA MATIAS. APOIO NA GERAÇÃO DE PLANO DE PROJETO NO AMBIENTE WEBAPSEE. 2008. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Universidade Federal do Pará.

ROCHA, Vanderlene Covre ; REIS, C. A. L. ; FAVERO, Eloi Luiz . Metodologia para Implementação do MPS.BR Utilizando o Ambiente WebAPSEE. In: Simpósio Brasileiro de Qualidade de Software, 2008, Florianopolis. Simpósio Brasileiro de Qualidade de Software. Campinas : SBC, 2008. v. 1. p. 171-186

ROCHA, Vanderlene C. Metodologia para Implementação do MPS.BR Utilizando o Ambiente WebAPSEE. 2009. (Dissertação de Mestrado, a ser defendida) Programa de Pós-Graduação em Engenharia Elétrica. Universidade Federal do Pará. Belém.

SALES, Ernani de Oliveira ; REIS, C. A. L. ; REIS, Rodrigo Quites . Apoio à Gerência de Configuração de Artefatos em um Ambiente de desenvolvimento de Software. In: Simpósio Brasileiro de Engenharia de Software, 2008, Campinas. Simpósio Brasileiro de Engenharia de Software. PORTO ALEGRE : SBC, 2008.

SALES, MURILO FARIAS. GERÊNCIA DE REQUISITOS INTEGRADA À GERÊNCIA DE PROJETOS NO AMBIENTE WEBAPSEE. 2008. Trabalho de Conclusão de Curso. (Ciência da Computação) - Universidade Federal do Pará.

SILVA, ALISON RICARDO SANTOS. Auditoria e Contabilidade de Gerência de Configuração de Artefatos de Software no Ambiente WebAPSEE. 2008. Trabalho de Conclusão de Curso. (Ciência da Computação) - Universidade Federal do Pará.

Travassos, G.; Kalinowski, M. (2008) “iMPS: resultados de desempenho de organizações que adotaram o modelo MPS”, Relatório Técnico. Campinas, SP.

Procedimento de Avaliação de Qualidade de Serviços

Carlos Constantino Moreira Nassur

Centro de Informática (CENIN) da Câmara dos Deputados Praça dos Três Poderes, Palácio do Congresso Nacional, Anexo I, 24º andar, Sala 2401 70160-900 – Brasília –DF – Brasil

carlos.nassur@camara.gov.br

Abstract. *This paper describes the report of Project 2.09, submitted by Camara dos Deputados in to Brazilian Quality and Productivity in Software Program 2008 cycle. The text describes the problem context, the goals that should be achieved, its justification for the developed work, the methodology adopted and the results achieved. Besides, it's attached the procedure for evaluating the quality of development and maintenance services for information systems that emerged from the project.*

Resumo. *Este artigo apresenta o relato do projeto 2.09, submetido pela Câmara dos Deputados ao ciclo 2008 do "Programa Brasileiro de Qualidade e Produtividade em Software". O texto descreve o contexto do problema, os objetivos e justificativas para o desenvolvimento do trabalho, a metodologia de execução e os resultados alcançados. Paralelamente, está anexado o procedimento para avaliação da qualidade dos serviços de desenvolvimento e manutenção de sistemas de informação, que foi elaborado a partir da execução do projeto.*

1 Introdução

O Centro de Informática é o órgão da estrutura administrativa da Câmara dos Deputados com a missão institucional de exercer as atividades normativas e executivas relacionadas à informática [1]. Com isto, torna disponível à Casa do Povo Brasileiro os necessários serviços de Tecnologia da Informação com alta qualidade, sempre calcados em bases técnicas e administrativas adequadas. Para cumprir suas atribuições, o CENIN adquire, desenvolve e adapta soluções de Tecnologia da Informação destinadas a apoiar as atividades administrativas e legislativas da Câmara dos Deputados. Não obstante o fato de possuir um quadro de pessoal próprio, com elevada qualificação técnica e capacidade de trabalho, o Centro de Informática enfrenta larga dificuldade para atender a determinadas demandas para o desenvolvimento de aplicações, especialmente quando as necessidades decorrem de solicitações sazonais ou extemporâneas cujo esforço para consecução não pode ser previamente dimensionado.

Algumas dessas demandas são primordialmente caracterizadas por requisitos imprevisíveis e/ou prazos de execução regimentalmente estabelecidos. Existem inúmeros casos ilustrativos de solicitações com tais características, como, por exemplo, o desenvolvimento de soluções de *software* para suporte ao processo de elaboração da Lei de Diretrizes Orçamentárias (LDO); ou mesmo para o Plano Plurianual (PPA). A cada exercício fiscal, o Poder Executivo submete ao Congresso Nacional o Projeto de Lei em determinada configuração. Em alguns casos, o formato recebido inviabiliza a reutilização das soluções de *software* previamente construídas; em outros, obriga a completa reconstrução dos sistemas de suporte para seu correto funcionamento. Os esforços para a elaboração e/ou adequação dos aplicativos devem, entretanto, obedecer a prazos e datas de entrega inegociáveis, por serem impostos por mecanismos normativos ou regimentais. Em outros exemplos, também já enfrentados pelo CENIN, algumas das decisões da Mesa Diretora da Casa, que modificam os processos administrativos já automatizados em soluções de *software*, usualmente impõem modificação dos sistemas de informação que os suportam. Em alguns casos, as decisões também estabelecem os prazos para sua implementação que, no entanto, não se mostram adequados e proporcionais à complexidade dos trabalhos que ensejam. Importa ressaltar que o atendimento destas demandas, sazonais ou extemporâneas, usualmente obriga o Centro a modificar seu planejamento tático e a redefinir as tarefas de seus colaboradores, de forma a obter com que alguns projetos previamente estabelecidos tenham seus recursos desviados, postergando e alongando, conseqüentemente, seus prazos de execução. Esta é uma característica do ambiente no qual a Câmara dos Deputados se insere. A solução para os problemas daí advindos não passa meramente pelo aumento do quadro de pessoal do CENIN, pois não há um padrão para o dimensionamento de uma equipe ideal. Não é tampouco razoável que se dimensione o tamanho do corpo técnico a partir dos picos de serviços ou das demandas sazonais e/ou extemporâneas. A solução encontrada para dotar o Centro de Informática dos recursos extraordinários que lhe permitam prover oportunamente os serviços de Tecnologia da Informação foi contratar, mediante licitação pública, empresa especializada na prestação dos serviços de desenvolvimento e manutenção de sistemas. A futura contratada, então, proverá a capacidade adicional necessária ao atendimento das demandas que superam a atual capacidade de produção do CENIN. A contratação de tais serviços impõe, entretanto, que o órgão da Administração defina critérios objetivos para a avaliação da qualidade dos serviços prestados [2]. Assim, este relato apresenta o trabalho desenvolvido no Centro de Informática para cumprir tal requisito, *i.e.*, o procedimento, elaborado no ciclo 2008, para objetivamente avaliar a qualidade dos serviços de desenvolvimento e de manutenção de sistemas prestados por terceiros.

2 Objetivos e Justificativa

A avaliação da qualidade dos serviços técnicos, além de expressa obrigação para órgão da Administração que contrata bens e serviços [4], configura-se como um aspecto importante no controle da execução do contrato [2], pois pode subsidiar de maneira bastante prática, objetiva e direta a aplicação de sanções à contratada, em razão de falhas, defeitos e imperfeições eventualmente emergentes. Um aspecto importante a ser considerado é o fato de que o fornecimento e a aceitação de serviços insatisfatórios pode resultar em prejuízos à Administração [2]. Assim, ao se adotar um procedimento formal para avaliação da qualidade dos serviços prestados por terceiros, mitigam-se os riscos para a Casa quanto a essa prestação. Também é abrandado o risco para a prestadora de serviços, pois é sobremaneira diminuída a avaliação subjetiva da qualidade dos serviços que eventualmente prestará. Desse modo, a prestadora poderá avaliar, dimensionar e orçar de maneira mais precisa os recursos que serão necessários para a prestação dos serviços. Por fim, a falta de um procedimento explícito para a avaliação da qualidade de serviços executados, especialmente aqueles que decorrentes do desenvolvimento e da manutenção de sistemas de informação, resulta em um processo subjetivo, sujeito a questionamentos da contratada, não se podendo descartar a possibilidade de restarem inviabilizados o adequado controle do contrato, a aplicação de sanções e, pior, a própria realização dos serviços. Assim, o objetivo do projeto submetido pela Câmara dos Deputados ao ciclo 2008 do Programa Brasileiro da Qualidade e Produtividade em *Software* do Ministério da Ciência e Tecnologia foi conceber um procedimento claro e objetivo para a avaliação da qualidade dos serviços de desenvolvimento e manutenção de sistemas de informação prestados por terceiros ao seu Centro de Informática.

3 Metodologia de Execução

O projeto foi desenvolvido mediante abordagem de revisão bibliográfica do tema. Inicialmente, foi efetuada ampla pesquisa e revisão teórica do assunto, com especial atenção para as normas e os modelos de referência. Foram estudadas as normas NBR ISO/IEC 12207 [6], ISO/IEC 9126 [5,7] e ISO/IEC 14598 [5,8], além da literatura tradicional de Engenharia de *Software* e de títulos específicos das temáticas de qualidade de *software* [6]. Também foram utilizadas como insumo para a elaboração do procedimento de avaliação da qualidade dos serviços as orientações do Tribunal de Contas da União [5], com especial atenção para o Acórdão 786/2006–Plenário [3], que apresentou ampla justificativa e forneceu os alicerces para o desenvolvimento do projeto. A revisão teórica, então, forneceu os insumos suficientes para a proposição de um procedimento de avaliação da qualidade dos serviços de desenvolvimento e manutenção de sistemas de informação, doravante simplesmente referenciados como “serviços”.

4 Resultados Obtidos

O desenvolvimento do projeto em tela resultou em um procedimento prático para avaliação da qualidade dos serviços. O Anexo A deste documento apresenta o procedimento parametrizado pelo Centro de Informática para a avaliação da qualidade daqueles serviços. Uma característica importante que se buscou adicionar ao procedimento elaborado foi o de extrapolar o conceito usual de qualidade. Tradicionalmente, associa-se qualidade ao atendimento dos requisitos previamente estabelecidos. Assim, diz-se que determinado produto ou serviço possui “qualidade” quando os seus requisitos são plenamente atendidos. No contexto do *software*, outro aspecto de qualidade também é importante: a opinião do destinatário final dos serviços, ou seja, do usuário dos aplicativos. Assim, o procedimento de avaliação de qualidade elaborado considera a expectativa quanto aos efeitos de uso do produto desenvolvido ou mantido por terceiros, pois seu usuário muitas vezes não deseja o produto em si, mas os efeitos resultantes do uso do produto, que caracterizam, desta forma, suas necessidades e, conseqüentemente, a qualidade desejada. O procedimento de avaliação da qualidade descrito nesse anexo está intimamente associado às próprias características do ambiente do Centro de Informática. Assim, traz referências explícitas a algumas características lá encontradas como, *e.g.*, os artefatos preconizados pelo processo de desenvolvimento de sistemas, a plataforma de desenvolvimento e operação de aplicativos e as ferramentas de *software* utilizadas no desenvolvimento e na manutenção de sistemas de informação.

5 Aplicabilidade dos Resultados

O desenvolvimento do projeto resultou em um procedimento para avaliação da qualidade dos serviços, que viabilizará, no âmbito da Câmara dos Deputados, a contratação, mediante licitação pública, de iniciativa privada devidamente capacitada para a prestação de tais serviços. Adicionalmente, o desenvolvimento do projeto também capacitou a equipe nele envolvida e forjou, no corpo técnico do Centro de Informática da Câmara dos Deputados, os conhecimentos necessários para se atuar em dois campos distintos e complexos: o da Tecnologia da Informação, especificamente no segmento de Engenharia de *Software*; e o de licitações e contratos públicos.

6 Características Inovadoras

Um resultado relevante e inovador, também obtido com o desenvolvimento do projeto, foi a comprovação da viabilidade de se avaliar objetivamente a qualidade de serviços prestados por terceiros, especialmente quando da contratação do desenvolvimento e da manutenção de sistemas de informação. Assim, o procedimento de

avaliação da qualidade de serviços resultante do projeto auxilia no rompimento do paradigma até então estabelecido na Administração: contratação de mão-de-obra frente *versus* contratação de serviços.

A contratação de mão-de-obra difere substancialmente da de serviços. Não há como se sobrepor ou comparar atividades absolutamente distintas. Contratação de serviços, mesmo sendo esses desempenhados por seres humanos, não se confunde com a de mão-de-obra. O primeiro deve ser pontual e definido, de onde se espera um resultado específico. O outro é a mera agregação de ajuda claramente individualizada, seja ela especializada ou não. A contratação de serviços obriga a contratada ao resultado esperado e àquela caberá providenciar a estrutura necessária para atingi-lo, seja com pessoal ou com material. A contratação de mão-de-obra vincula o resultado à proficiência do profissional escolhido para seu desempenho. Assim, a qualidade dos serviços é diretamente dependente da capacidade e competência do profissional. Uma das corriqueiras argumentações favoráveis à preferência pela contratação de mão-de-obra residia na inviabilidade da avaliação da qualidade dos serviços. Assim, contratava-se o profissional de informática para se desenvolver ou manter sistemas de informação ao invés de se contratar a iniciativa privada para se prover tais serviços. Então, uma vez disponível um procedimento confiável para avaliação da qualidade de serviços, derruba-se mais um argumento contrário à contratação de tais serviços. Consequentemente, aprimora-se a Administração e a indústria de Tecnologia da Informação, pois ambas podem utilizá-lo como preceito, com os devidos ajustes, para a celebração de contratos para a execução dos serviços.

7 Conclusão e Perspectivas Futuras

O presente projeto, cujo objetivo foi o de elaborar um procedimento para avaliação da qualidade de serviços de desenvolvimento e de manutenção de sistemas de informação, foi alcançado. O procedimento resultante, conforme exposto no Anexo A, é aplicável ao Centro de Informática da Câmara dos Deputados. Certamente ainda não atingiu o estado da arte. Entretanto, será utilizado em futuras licitações daquele órgão com vistas na contratação de empresa privada para a prestação de serviços. Assim, sua aplicação prática certamente ensejará melhorias e aperfeiçoamentos. O desenvolvimento do projeto trouxe à sua equipe diversos desafios, especialmente por tratar de temas usualmente disjuntos: contratações públicas e Engenharia de *Software*. Assim, a equipe teve que consolidar conhecimentos e construir correlações em segmentos tão distintos que às vezes parecia impossível identificar afinidades. O desafio, entretanto, resultou em um processo que pode ser diretamente utilizado em certames públicos para a contratação de serviços de desenvolvimento e manutenção de sistemas. Pode, também, servir de referência pela iniciativa privada para o estabelecimento de acordos e contratos para a prestação de tais serviços.

O procedimento também poderá ser adaptado para utilização por outros órgãos da Administração. Para tanto, bastará contemplar as características de seu próprio ambiente operacional, tais como os processos que executa, os artefatos e a plataforma tecnológica que utiliza.

Por fim, a definição de um modelo de avaliação objetiva da qualidade pode servir de base para novas iniciativas, que visam a avaliar a qualidade de serviços diversos, tais como os de limpeza, de transporte ou mesmo de segurança. Ao avaliar objetivamente a qualidade dos serviços, a Administração atenua os riscos envolvidos na prestação dos serviços e, consequentemente, aplica melhor os recursos públicos.

8 Referências Bibliográficas

[1] BRASIL. Câmara dos Deputados. **Resolução da Câmara dos Deputados Nº 16, de 22 de janeiro de 1997**. Cria o Centro de Informática da Câmara dos Deputados e dá outras providências. Brasília: Diário da Câmara dos Deputados, 2007.

[2] _____.Tribunal de Contas da União. Monitoramento da Concorrência 06/2005, promovida pelo Ministério do Desenvolvimento, Indústria e Comércio Exterior – MDIC, com o objetivo de contratar serviços de informática nas áreas de desenvolvimento de sistemas e acompanhamento de projetos. **Acórdão Nº 786/2006 – Plenário**. Relator: Augusto Sherman Cavalcanti.

[3] _____.Tribunal de Contas da União. **Licitações e contratos: orientações básicas**. 3. ed, rev. atual. e ampl. Brasília: TCU, Secretaria de Controle Interno, 2006. 409 p.

[4] _____.**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.

[5] VILLAS BOAS, André Luiz de Castro. **Qualidade e avaliação de produto de software**. Lavras: Universidade Federal de Lavras – UFLA/FAEPE, 2005.

[6] ABNT. NBR ISO/IEC 12207 – Sistemas e engenharia de *software* - Processos de ciclo de vida de *software*.

[7] _____. NBR ISO/IEC 9126 – Engenharia de *Software* – Qualidade de produto.

[8] _____. NBR ISO/IEC 14598 – Tecnologia de Informação – Avaliação de produto de *software*.

ANEXO A:

Procedimento de avaliação de qualidade

Objetivo da Avaliação

O procedimento de avaliação de qualidade descrito neste Anexo visa aferir objetivamente a qualidade dos serviços prestados.

Visão Geral

A avaliação de qualidade define o modelo de qualidade que será adotado para a execução do contrato. O modelo de qualidade é formado por um conjunto de atributos de qualidade, os quais definem os indicadores, os critérios e os procedimentos objetivos para a avaliação da qualidade dos produtos de *software* elaborados e entregues pela contratada à Administração. Também são definidos os responsáveis pela avaliação e apontadas as escalas de valores e referências mínimas de qualidade de forma que seja possível determinar, explicitamente, se um produto qualquer é ou não aceitável pela Administração. Acessoriamente, o modelo de qualidade deve também ser utilizado para determinar a sanção administrativa aplicável à contratada por não alcançar os padrões aceitáveis de qualidade definido em um atributo qualquer.

O modelo de avaliação de qualidade apresenta duas modalidades distintas de atributos de qualidade: a primeira modalidade reúne os atributos de qualidade que não possuem itens de conformidade1 e, a segunda, congrega somente os atributos de qualidade que possuem itens de conformidade. O modelo de qualidade define os seguintes atributos de qualidade para a primeira modalidade: “Cobertura dos Testes”, “Completude da Implementação Funcional”, “Efetividade da Documentação de Usuário ou do Sistema de Ajuda” e “Entendimento Funcional”, todos detalhados na Tabela 1. A avaliação de qualidade dos atributos desta modalidade se dará a partir da execução dos procedimentos previstos para cada um deles sobre os artefatos entregues pela contratada após a execução dos serviços. Como exemplo, o atributo de qualidade “Cobertura dos Testes” exigirá a contagem de duas variáveis, as variáveis “A” e “B”. O valor da primeira delas, a variável “A”, será obtido através da contagem dos casos de teste que foram efetivamente executados pela contratada, cujas evidências foram entregues juntamente com o produto final. A segunda variável, a “B”, será valorada a partir da própria especificação entregue pela contratante à contratada para se construir o produto solicitado.[2]

Como cada caso de uso consiste em um conjunto de funcionalidades inter-relacionadas de um sistema, é possível extrair um conjunto de casos de teste através da análise da especificação textual deste artefato, tomando-se como referência os fluxos básico, alternativo e de exceção, conforme o procedimento:

1. Através da análise do fluxo básico pode-se obter um caso de teste que contemple a execução completa[2] desse caso de uso. Neste caso, conta-se um caso de teste para o fluxo básico.

2. Através da análise do fluxo alternativo, consegue-se gerar, para cada item deste fluxo, um caso de teste específico para verificar as variações de comportamento. O número de casos de teste possíveis, portanto, dependerá do número de opções do fluxo alternativo. Aplica-se o mesmo procedimento aos fluxos de exceção.

3. Após contadas todas as possibilidades, ou seja, após valorada a variável “B”, aplica-se o cálculo[3] e obtém-se o resultado da aplicação deste atributo de qualidade ao produto entregue pela contrata que, então, deverá ser comparado com o seu valor aceitável.

A segunda modalidade, que abarca os atributos de qualidade “Completude da Documentação Técnica de Absorção de Sistema Legado” e “Manutenibilidade”, possui diversos itens de conformidade, listados na “Tabela 2”. A avaliação de qualidade dos produtos elaborados pela contratada com o uso dos atributos desta modalidade se dará através da avaliação e verificação da conformidade de cada um dos itens de conformidade definidos para o atributo qualidade.

Ao se demandar um serviço qualquer à contratada, o demandante deverá indicar quais atributos de qualidade serão aplicados, conforme o tipo do serviço a ser executado e o definido na “Tabela 3”.

[2] 1 Item de conformidade: determinada condição que necessita ser atendida.
2 “Caminho feliz” em um fluxo básico de um caso de uso.
3 Coluna “Cálculo” da “Tabela 1”.

Tabela 1: Procedimentos para aferição da qualidade dos produtos entregues.

Atributo de Qualidade	Aspecto de Qualidade Avaliado	Método de Aplicação	Cálculo	Interpretação do resultado	Tipo de Medida	Responsável pela avaliação	Valores aceitáveis
COBERTURA DOS TESTES	Quão adequadamente testado foi o produto.	Contagem do número de casos de teste submetidos ao produto durante o seu teste comparado com o número de casos de teste suficientes para cobrir todos os requisitos da ordem de serviço.	X = A/B, onde: "A" é o número dos casos de teste executados no produto e "B" é o número de casos de teste suficientes para cobrir todos os requisitos da ordem de serviço.	0 ≤ X ≤ 1 Quanto mais próximo de 1, melhor o resultado obtido.	Contagem absoluta das variáveis "A" (através das evidências de teste fornecidas) e "B".	Gestor da Ordem de Serviço.	X > 0,80
COMPLETUDE DA DOCUMENTAÇÃO TÉCNICA DE ABSORÇÃO DE SISTEMA LEGADO	Quão completa é a documentação técnica produzida para um determinado sistema legado.	Contagem do número de itens de conformidade **que não foram alcançados** comparado com o número total de itens de conformidade definidos para o atributo de qualidade.	X = 1 – (A/B), onde: "A" é o número de itens de conformidade especificados e que não foram implementados e "B" é o número total de itens de conformidade definidos para este atributo de qualidade na "Tabela 2".	0 ≤ X ≤ 1 Quanto mais próximo de 1, melhor o resultado obtido.	Contagem absoluta das variáveis "A" e "B".	Gestor da Ordem de Serviço.	X = 1
Atributo de Qualidade	**Aspecto de Qualidade Avaliado**	**Método de Aplicação**	**Cálculo**	**Interpretação do resultado**	**Tipo de Medida**	**Responsável pela avaliação**	**Valores aceitáveis**

COMPLETUDE DA IMPLEMENTAÇÃO FUNCIONAL	Quão completa é a implementação conforme a especificação dos requisitos da ordem de serviço.	Contagem do número de funções que não foram implementadas comparada com o número total de funções especificadas nos requisitos da ordem de serviço.	X = 1 – (A/B), onde: “A” é o número de funções não implementadas detectadas na avaliação e “B” é o número total de funções especificadas na ordem de serviço.	0 ≤ X ≤ 1 Quanto mais próximo de 1, melhor o resultado obtido.	Contagem absoluta das variáveis “A” e “B”.	Gestor da Ordem de Serviço.	X = 1
EFETIVIDADE DA DOCUMENTAÇÃO DE USUÁRIO OU DO SISTEMA DE AJUDA	Qual proporção das tarefas pode ser corretamente desenvolvidas após a leitura da documentação de usuário ou do próprio sistema de ajuda do produto.	Observação da utilização do produto pelo usuário final e contagem do número de tarefas que foram corretamente desempenhadas apenas com auxílio da documentação de usuário ou do próprio sistema de ajuda do produto.	X = A/B, onde: “A” é o número de tarefas corretamente completadas após consulta da documentação de usuário ou ao próprio sistema de ajuda do produto e “B” é o número total de tarefas submetidas à avaliação.	0 ≤ X ≤ 1 Quanto mais próximo de 1, melhor o resultado obtido.	Contagem absoluta das variáveis “A” e “B”.	Gestor da Ordem de Serviço. Grupo de usuários finais da aplicação.	X > 0,95

Atributo de Qualidade	Aspecto de Qualidade Avaliado	Método de Aplicação	Cálculo	Interpretação do resultado	Tipo de Medida	Responsável pela avaliação	Valores aceitáveis
ENTENDIMENTO FUNCIONAL	Qual porção das funcionalidades do produto criado será corretamente entendida pelos	Entrevista com os usuários do produto. Contagem do número de funções disponíveis ao	X = A/B, onde: “A” é o número de funções corretamente descritas pelo usuário durante a entrevista e	0 ≤ X ≤ 1 Quanto mais próximo de 1, melhor o resultado obtido.	Contagem absoluta das variáveis “A” e “B”.	Gestor da Ordem de Serviço. Grupo de usuários finais da aplicação.	X > 0,90

	seus usuários finais.	usuário cujos propósitos são facilmente entendidos comparado com o número total de funções disponíveis para uso.	“B” é o número total de funções disponíveis para uso no produto.				

Atributo de Qualidade	Aspecto de Qualidade Avaliado	Método de Aplicação	Cálculo	Interpretação do resultado	Tipo de Medida	Responsável pela avaliação	Valores aceitáveis
MANUTENIBILIDADE	Quão compatível é a manutenibilidade do produto entregue em relação aos procedimentos padronizados e à capacidade de avaliação do impacto das eventuais manutenções.	Contagem do número de itens de conformidade **que não foram alcançados** comparado com o número total de itens de conformidade definidos para o atributo de qualidade.	X = 1 – (A/B), onde: “A” é o número de itens de conformidade especificados e que não foram implementados e “B” é o número total de itens de conformidade definidos para este atributo de qualidade na “Tabela 2”	0 ≤ X ≤ 1 Quanto mais próximo de 1, melhor o resultado obtido.	Contagem absoluta das variáveis “A” e “B”.	Gestor da Ordem de Serviço.	X > 0,95

Tabela 2: Itens de conformidade especificados por atributo de qualidade

Atributo de Qualidade	Itens de Conformidade
COMPLETUDE DA DOCUMENTAÇÃO TÉCNICA DE ABSORÇÃO DE SISTEMA LEGADO	1. Sumário Executivo completamente preenchido com as informações pertinentes no repositório de dados do sistema legado absorvido da ferramenta Borland CaliberRM; 2. Requisitos de Escopo, Funcionais e Não Funcionais completamente identificados e descritos no repositório de dados do sistema absorvido da ferramenta Borland CaliberRM; 3. Diagramas de Casos de Uso criados para o sistema legado na ferramenta Borland Together Eclipse Edition e inseridos no repositório de dados do sistema legado absorvido da ferramenta Borland StarTeam; 4. Criadas todas as especificações para todos os Casos de Uso identificados na absorção do sistema legado e inseridas no repositório de dados do sistema legado absorvido da ferramenta Borland StarTeam; 5. Diagramas de Navegação criados para o sistema legado na ferramenta Borland Together Eclipse Edition e inseridos no repositório de dados do sistema legado absorvido da ferramenta Borland StarTeam.
MANUTENIBILIDADE	1. Todos os Casos de Uso da ordem de serviço são rastreáveis, *i.e.*, são associados, via ligação, aos requisitos funcionais que implementa; 2. Todos os Casos de Uso da ordem de serviço estão associados, via ligação, às respectivas especificações; 3. Todos os diagramas de casos de uso da ordem de serviço estão associados aos respectivos diagramas de navegação; 4. Todas as classes de interface dos diagramas de navegação do protótipo de interface da ordem de serviço estão associadas, via *hyperlink*, às páginas e/ou formulários criados; 5. Todos os atributos das classes de interface (os campos das interfaces dos protótipos) são rastreáveis, *i.e.*, estão associados aos elementos de dados do dicionário corporativo de dados; 6. Todas as funcionalidades das classes de interface dos diagramas de navegação são rastreáveis, *i.e*, estão associadas aos diagramas de seqüencia, ou de classes, que materializam a implementação da aplicação; 7. É possível manter o controle de versão de todos os artefatos que compõem o produto através do uso das ferramentas adotadas pelo Centro de Informática; 8. É possível manter o controle de configuração (*baseline*) de todas as versões dos artefatos que compõem o produto através do uso das ferramentas adotadas pelo Centro de Informática; 9. As classes foram organizadas (empacotadas) em conformidade com o empacotamento padrão definido pelo processo de desenvolvimento adotado pelo Centro de Informática; 10. Os diagramas criados para a modelagem do sistema seguiram o empacotamento padrão definido pelo processo de desenvolvimento adotado pelo Centro de Informática.

Tabela 3: Atributos de qualidade aplicáveis aos possíveis tipos de ordem de serviço.

Tipos de Serviço	Atributos de qualidade aplicáveis
DESENVOLVIMENTO E MANUTENÇÃO MIGRATÓRIA	1. Cobertura dos Testes; 2. Completude da Implementação Funcional; 3. Efetividade da Documentação de Usuário ou do Sistema de Ajuda; 4. Entendimento Funcional; 5. Manutenibilidade.
MANUTENÇÃO CORRETIVA	1. Entendimento Funcional; 2. Efetividade Da Documentação de Usuário ou do Sistema de Ajuda; 3. Manutenibilidade.
MANUTENÇÃO EVOLUTIVA	1. Cobertura dos Testes; 2. Completude da Implementação Funcional; 3. Manutenibilidade; 4. Efetividade Da Documentação de Usuário ou do Sistema de Ajuda; 5. Entendimento Funcional.

Um *Framework* para Avaliação da Implantação do MR MPS em Grupos Cooperados de Empresas

Rafael Prikladnicki

Faculdade de Informática – PUCRS – POA – RS – Brasil

rafaelp@pucrs.br

Resumo. *Neste artigo apresenta-se o relatório final do projeto 2.10, cujo objetivo foi desenvolver um framework para avaliação da implantação do MR MPS em grupos cooperados de empresas. Acredita-se que o framework proposto neste projeto, que inclui um modelo e uma ferramenta de apoio, pode contribuir para melhorar a execução de projetos cooperados de implantação do MR MPS no país.*

1. Introdução

Nos últimos anos, o Brasil tem vivenciado um crescimento bastante significativo em relação ao investimento na melhoria e qualidade dos produtos de software. Isto tem sido possível, entre outros fatores, pelo crescimento dos programas de melhoria de processo de desenvolvimento de software nas empresas. Mas a implementação de um programa de melhoria não é uma tarefa simples para muitas empresas brasileiras, principalmente micro, pequenas e médias empresas. E nesta realidade, as diferenças de serviços prestados por essas empresas tornaram-se cada vez mais distantes de grandes empresas fornecedoras de software, pois estas últimas geralmente têm maior facilidade para investir no aperfeiçoamento dos seus processos.

Dentro deste contexto, em dezembro de 2003, foi criado o programa MPS.BR, uma iniciativa liderada pela SOFTEX (Associação para Promoção da Excelência do Software Brasileiro), cujo objetivo era atender as necessidade de padrões de qualidade na indústria de software no Brasil, e que pudesse ser aplicado em empresas com menor capacidade de investimento (Weber et al, 2005). O MPS.BR, segundo Softex (2008) é um programa para melhoria de processo do software brasileiro, e é coordenado pela SOFTEX, contando com apoio do MCT, da FINEP, do SEBRAE e do BID. Ele baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e melhoria da qualidade e produtividade de empresas de software e serviços correlatos. Ele é composto por três componentes: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS), sendo este último o tema do projeto aqui relatado.

O modelo de referência MR MPS foi desenvolvido de forma adequada para a realidade das empresas brasileiras, e foi baseado em modelos e normas de qualidade tais como a ISO 12207, a ISO 15504 e o CMMI (Softex, 2008). O surgimento do MPS.BR e seu modelo de referência é um dos motivos para o aumento do número de empresas brasileiras que estão implantando melhorias nos seus processos (Moreira et. al., 2005; Nunes et. al., 2006). Uma facilidade criada pelo programa MPS.BR foi a possibilidade de desenvolver projetos de melhoria em cooperativas de empresas, podendo assim haver uma ajuda mútua entre as empresas, tanto financeira (divisão de custos) como também na busca de problemas e soluções comuns. É neste contexto que este projeto se insere.

2. Objetivos e Justificativa

Projetos em grupos cooperados têm sido bastante utilizados nos últimos anos no Brasil (Wangenheim et al, 2005). Vários grupos cooperados foram organizados tendo por base o antigo SW-CMM, o CMMI, e agora o MR MPS (Prikladnicki et al, 2008; Prikladnicki et al, 2007; Palestino & Mendonça, 2008). No MPS.BR, o MNC (Modelo de Negócio Cooperado) prevê a organização de grupos de empresas para a implementação do MR MPS (Palestino & Mendonça, 2008; MN-MPS.BR, 2007).

A partir da experiência desenvolvida com grupos cooperados no Rio Grande do Sul (IOGE Softsul) e da perspectiva de continuidade das cooperativas de empresas, entendeu-se que havia a necessidade de melhorar constantemente os projetos em grupos de empresas. Percebeu-se com isso a necessidade de criar mecanismos formais de avaliação e melhoria destes projetos não apenas em nível local, mas também de forma mais abrangente, em nível nacional. Isto permitiria uma maior troca de experiência e nivelamento entre os diversos grupos cooperados em planejamento e em andamento.

Desta forma, o objetivo deste projeto foi o desenvolvimento de um *framework* para avaliação da implantação do MR MPS em grupos cooperados de empresas. Sendo assim, este projeto envolveu a criação de um modelo de avaliação específico, denominado de MA-MNC, e uma ferramenta de apoio, denominada de *mps.tool*.

3. Metodologia de Execução

Este projeto teve início no ano de 2006. O desenvolvimento do modelo de avaliação MA-MNC e da ferramenta *mps.tool* envolveu diversos colaboradores durante este período, quais sejam:

- um aluno de pós-graduação da PUCRS entre 2006 e 2007;

- um aluno de graduação da PUCRS em 2008;

- consultores de implementação vinculados a Instituição Implementadora responsável pelos grupos cooperados no RS (II Softsul), entre 2006 e 2008;

- empresas participantes dos dois primeiros grupos cooperados de implantação do MR MPS no RS (um finalizado em 2007 e outro finalizado em 2008).

Para garantir a riqueza das informações e manter a visão geral de todo o projeto, apresenta-se neste relatório todas as atividades desenvolvidas, destacando quais foram especificamente planejadas e executadas no ciclo de 2008. O projeto foi dividido em cinco fases, quais sejam:

Fase 1 – Revisão bibliográfica: inicialmente, foi desenvolvida uma revisão bibliográfica destinada a ampliar a visão sobre o assunto, permitindo a formação de um conhecimento coerente e embasado. Como fontes bibliográficas de sustentação do estudo foram utilizados artigos publicados em congressos e periódicos nacionais e internacionais, com foco especial no Simpósio Brasileiro de Qualidade de Software (SBQS) e os Workshops MPS.BR, livros e principalmente o site da Sociedade SOFTEX (http://www.softex.br). Este estudo foi realizado inicialmente pelo aluno de pós-graduação (Furlanetto, 2007), e posteriormente revisado pelo aluno de graduação (Heineck, 2008).

Fase 2 – Proposta de modelo preliminar: a partir da revisão de literatura e do plano de implementação de cooperativas de empresas desenvolvido na Softsul, um modelo preliminar de avaliação de grupos cooperados foi proposto em meados de 2006. Este modelo envolveu um instrumento de pesquisa que abordava os aspectos de implementação do MR-MPS em grupos cooperados a partir de duas dimensões: a importância de cada aspecto para o sucesso do projeto e a satisfação com cada aspecto após a conclusão do mesmo. Este instrumento de avaliação está documentado em Furlanetto (2007).

Fase 3 – Avaliação do modelo preliminar: o modelo preliminar foi posto em prática, sendo avaliado com dados reais dos dois primeiros grupos de empresas organizados pela Softsul. Os dados destes grupos foram coletados em 2007 e 2008 respectivamente, sendo cinco empresas em cada grupo. Os dados da avaliação com o primeiro grupo de empresas está documentado em Prikladnicki et al (2007), enquanto que uma comparação entre as duas avaliações pode ser encontrada em Prikladnicki et al (2008). A avaliação do modelo preliminar com o segundo grupo de empresas fez parte do escopo do ciclo 2008 deste projeto.

Fase 4 – Modelo consolidado: a partir do retorno recebido, melhorias foram incorporadas ao instrumento e ao modelo como um todo. Estas melhorias diziam respeito principalmente à análise dos dados coletados e ao ajuste dos aspectos de importância e satisfação que seriam avaliados. Com isto, propôs-se, em 2008, o MA-MNC, um modelo consolidado de avaliação de grupos cooperados de empresas que implementam o MR-MPS. Este modelo será aplicado no final do terceiro grupo de empresas organizado pela IOGE Softsul, previsto para terminar em meados de 2009.

Fase 5 – Ferramenta de apoio: para facilitar a replicação das avaliações em diferentes grupos de empresas organizados por diferentes IOGEs, bem como a comparação dos resultados e o uso do modelo em nível nacional, uma ferramenta de apoio foi desenvolvida. A versão inicial desta ferramenta foi denominada *mps.tool* e está documentada em Heineck (2008). Pretende-se utilizá-la na avaliação do terceiro grupo de empresas no RS, além de disponibilizá-la para uso externo. É de interesse da equipe do projeto disponibilizar tanto o modelo como a ferramenta para uso de outros grupos cooperados em execução no Brasil, fomentando a melhoria contínua do programa MPS.BR como um todo.

4. Resultados Obtidos

O projeto produziu os seguintes resultados:

- **produtos de software gerados (módulos ou programas de computador resultantes, disponibilizados para o mercado):** desenvolveu-se a ferramenta *mps.tool*, para dar suporte ao modelo de avaliação proposto. Sua documentação técnica pode ser obtida no endereço http://www.inf.pucrs.br/~rafael/mpstool.

- **outros produtos gerados (que foram disponibilizados para o mercado):** desenvolve-se o *website* do projeto, que inclui principalmente informações sobre a ferramenta *mps.tool*. Informações sobre o MA-MNC também serão disponibilizadas neste local.

- **processos que contribuem para a melhoria da qualidade de software, descritos e institucionalizados:** o desenvolvimento da ferramenta só foi possível a partir do desenvolvimento do MA-MNC (que incluiu o processo de avaliação e o guia de análise), o modelo de avaliação para grupos cooperados de empresas que implementam o MR MPS. O modelo prevê um processo a ser seguido durante a avaliação.

- **métodos ou algoritmos desenvolvidos:** durante o desenvolvimento da ferramenta *mps.tool*, foram desenvolvidos algoritmos para implementar a sistemática de avaliação de forma automatizada. Alem disso, foram desenvolvidos algoritmos para implementar a comparação de avaliações em uma IOGE e entre IOGEs.

Estas comparações podem ser gerais ou específicas, incluindo a diferenciação das empresas em um mesmo grupo cooperado (comparações entre empresas de nível G e de nível F, por exemplo).

- **artigos publicados:** este projeto gerou diversos artigos. O primeiro deles foi publicado em 2007 no SIMPROS (Simpósio Internacional de Melhoria de Processo de Software), documentando a avaliação preliminar do modelo de avaliação proposto (Prikladnicki et al, 2007). Especificamente no ciclo 2008, um relato de experiência foi publicado no Simpósio Brasileiro de Qualidade de Software (Prikladnicki et al, 2008). Além disso, um artigo técnico foi submetido para o SBQS 2009 (Prikladnicki & Galarraga, 2009).

- **recursos humanos capacitados:** no escopo deste projeto, um trabalho de especialização foi finalizado no ano de 2007 (Furnaletto, 2007). Especificamente neste ciclo de 2008, a ferramenta *mps.tool* desenvolvida fez parte do trabalho de conclusão de curso de um aluno do Bacharelado em Sistemas de Informação da PUCRS (Heineck, 2008). Ambos os alunos faziam parte do quadro de colaboradores de empresas que haviam participado dos dois primeiros grupos cooperados de empresas coordenados pela Softsul. Além disso, as avaliações executadas a partir do modelo proposto contribuíram para capacitar os recursos humanos da IOGE Softsul no planejamento e gestão de grupos cooperados de empresas. Isto se refletiu no reconhecimento do projeto cooperativa MPS.BR Softsul como um dos destaques do ciclo 2007 do PBQP Software (Becker et al, 2008).

- **dissertações e/ou teses geradas:** em 2007 foi gerado um volume de monografia de conclusão de curso de especialização da PUCRS (Furlanetto, 2007). No ciclo de 2008 foi gerado um volume de trabalho de conclusão de curso de graduação (Heineck, 2008).

- **eventuais parcerias ou programas de transferência de tecnologia efetuados:** tanto o modelo como a ferramenta serão utilizados na avaliação do terceiro grupo de empresas no RS, prevista para ocorrer no primeiro semestre de 2009. Além disso, existe a possibilidade de usar o MA-MNC e a *mps.tool* de forma periódica em todos os grupos cooperados no RS, a partir de conversas já iniciadas com a empresa *Software Process* (www.swprocess.com.br), parceira da Softsul, e que será a II responsável pela implementação do MR MPS nas próximas cooperativas de empresas no RS coordenadas pela IOGE Softsul.

5. Aplicabilidade dos resultados

O modelo MA-MNC foi avaliado em duas oportunidades na sua versão preliminar (nos dois primeiros grupos de empresas da cooperativa MPS.BR – Softsul). Estas avaliações concentraram-se principalmente nos componentes “Instrumento de Avaliação” e “Guia de Análise”. No primeiro grupo foram avaliadas duas empresas nível G e três empresas nível F. No segundo grupo foram avaliadas duas empresas nível F e três empresas nível G. Nas duas ocasiões melhorias foram identificadas e incorporadas nas edições subseqüentes do projeto cooperativa MPS.BR – Softsul (Prikladnicki et al, 2007; Prikladnicki et al, 2008).

Além disso, esta avaliação também resultou em melhorias no MA-MNC. Entre as principais melhorias, citamos a revisão do instrumento de avaliação e a correção de perguntas. A análise dos dados também foi revisada, alterando a estratégia inicial de uso de estatística descritiva para uma estratégia de ponderação das respostas a partir dos pesos dos participantes e a distribuição dos resultados em *quartis*. Existe ainda uma melhoria prevista para a ferramenta desenvolvida, que envolve a inclusão e acompanhamento de planos de ação a partir dos resultados das avaliações realizadas. Este acompanhamento pode ser restrito para a melhoria da operação de uma IOGE ou de um grupo de IOGEs e será detalhado oportunamente.

Em relação à aplicabilidade do modelo, a proposta foi avaliada em dois dos diversos grupos cooperados atualmente em execução no país e os resultados ainda não podem ser generalizados para outros projetos cooperados. Apesar de trazer aspectos relevantes que podem ser aproveitados por outras iniciativas semelhantes, os resultados refletem a percepção e ação sob a ótica das empresas que fazem parte de apenas uma IOGE. Desta forma, uma conseqüência imediata deste estudo foi o planejamento de ações para serem colocadas em prática no terceiro grupo cooperado de empresas no RS, iniciado em 2008. Além disso, o MA-MNC será utilizado para avaliar este terceiro grupo, bem como será disponibilizado para uso de outras cooperativas de empresa que implementam o MR MPS no Brasil .

6. Características inovadoras

Este projeto inova ao apresentar um modelo de avaliação para grupos cooperados de empresas que implementam o MR MPS. O modelo em si (MA-MNC) inova ao apresentar não apenas um processo de avaliação, mas também um guia de análise a partir de uma abordagem quantitativa e uma ferramenta de apoio (Figura 1). Também é importante destacar que este modelo de avaliação pode ser utilizado não apenas em grupos cooperados baseados no MR-MPS, mas também em grupos que se baseiam em outros modelos de qualidade. Para isso, acredita-se que se faça necessário uma revisão no instrumento de avaliação e nos itens que serão avaliados.

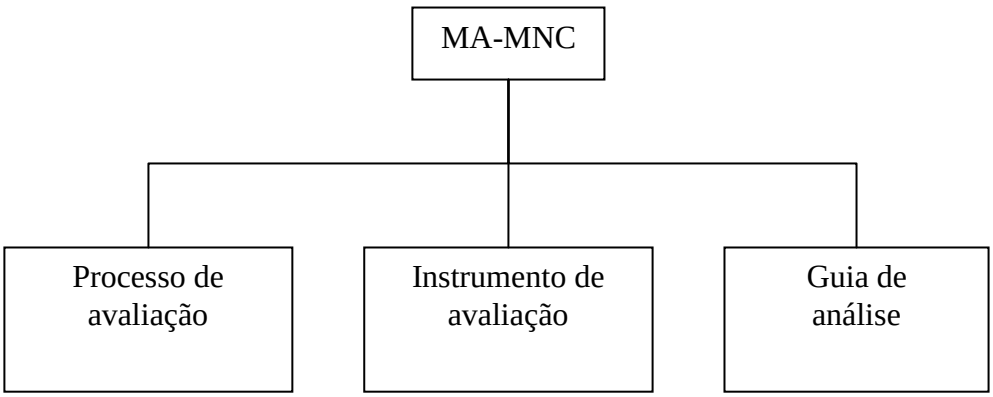


Figura 1. MA-MNC e seus componentes

Além disso, a partir da participação do autor em fóruns e eventos específicos sobre o programa MPS.BR, percebeu-se a necessidade de uma maior integração em âmbito nacional sobre as diferentes experiências com o MNC. Por este motivo, o MA-MNC é uma iniciativa pioneira nesta direção.

7. Conclusões e Perspectivas Futuras

Neste artigo apresentou-se o relatório final do projeto submetido ao ciclo 2008 do PBQP Software, cujo objetivo foi desenvolver um *framework* para avaliação da implementação do MR MPS em grupos cooperados de empresas. Um modelo (MA-MNC) e uma ferramenta (*mps.tool*) foram desenvolvidos, sendo esta parte integrante do modelo. O MA-MNC foi desenvolvido a partir da experiência vivenciada nos dois primeiros grupos cooperados de empresas, organizados pela IOGE Softsul no RS, onde se identificou a necessidade de estabelecer um processo sistemático de avaliação dos grupos cooperados.

Em relação a perspectivas futuras, entende-se que este modelo de avaliação apresenta-se como um instrumento importante para que os agentes responsáveis por programas cooperados possam melhorar a condução de outros projetos desta natureza. Neste sentido, acredita-se ser interessante a disponibilização do modelo e o uso em outros projetos cooperados no país, através de uma iniciativa que poderia ser coordenada pela SOFTEX, visando não apenas comparar os grupos cooperados de uma forma mais abrangente, mas também buscar melhorias para o MPS.BR como um todo.

8. Referências Bibliográficas

Becker, C. A., Prikladnicki, R. Galarraga, O. (2008). “Cooperativa MPS.BR - Primeiro Grupo de Empresas do RS”. ProQuality (UFLA), v. 4, p. 17-22.

Furlanetto, T. M. (2007). “Fatores Críticos para Implementação Cooperada do MR-MPS: Um Estudo da Cooperativa MPS.BR - Softsul”, Monografia de Conclusão, Especialização em Governança e Estratégia de TI, Faculdade de Administração, Contabilidade e Economia, PUCRS.

Heineck, A. (2008). “Uma Ferramenta para Apoio na Avaliação de Grupos Cooperados de Empresas que Implementam o Modelo MR MPS”, Trabalho de Conclusão, Bacharelado em Sistemas de Informação, Faculdade de Informática, PUCRS.

MN-MPS.BR (2007). “Modelo de Negócio para Melhoria de Processo de Software”, Disponível em www.softex.br.

Moreira, L. S. R., Samrsla, V. C, Mora, M. C., Prikladnicki, R., Bossle, R., Back, R. (2005). “Uma Avaliação das Equipes de Projeto em um Momento Pré-Avaliação Oficial de um Programa de Melhoria de Processo de Software visando o CMMI nível 2”, In: IV Simpósio Brasileiro de Qualidade de Software, Porto Alegre.

Nunes, E. D., Pinto, R., Rocha, A. R., Santos, G. (2006). “MPS.BR Nível E – Uma Avaliação em Verde e Amarelo”, In: V Simpósio Brasileiro de Qualidade de Software, Vila Velha.

Palestino, C. V. B., Mendonça, R, M. L. (2008). “Lições Aprendidas na Organização de Grupos de Empresas no Programa MPS.BR”, In: da Rocha, A. R. C, Weber, K. C, MPS.BR Lições Aprendidas, Softex, pp. 19-30.

Prikladnicki, R., Galarraga, O. (2009). “MA-MNC: Modelo de Avaliação para Grupos Cooperados de Empresas que Implementam o Modelo MR-MPS”, Submetido ao VIII Simpósio Brasileiro de Qualidade de Software.

Prikladnicki, R., Galarraga, O., Becker, C. A. (2008). “Comparando a Implantação de Projetos Cooperados baseados no MR-MPS através da Replicação de um Instrumento de Avaliação: Análise Quantitativa sob a Ótica das Empresas”, In: VII Simpósio Brasileiro de Qualidade de Software, Florianópolis.

Prikladnicki, R., Furlanetto, T. M., Becker, C. A. (2007). “Uma Análise Quantitativa da Implantação do MR MPS em Projetos Cooperados: Avaliação sob a ótica das Empresas”, In: SIMPROS, São Paulo, Brasil.

Softex. (2008). “MPS.BR – Melhoria de Processo do Software Brasileiro – Guia Geral – versão 1.2”, Sociedade SOFTEX, Dezembro de 2008. Disponível em www.softex.br.

Wangenheim, G. V. C., Pickler, K., Thiry, M., Zoucas, A. (2006). “Uma Abordagem para a Modelagem Colaborativa de Processos de Software em Micro e Pequenas Empresas” In: V Simpósio Brasileiro de Qualidade de Software, Vila Velha, Brasil.

Weber, K. C., Araújo, E., Machado, C. A. F., Scalet, D., Salviano, C. F., da Rocha, A. R. C. (2005). “Modelo de Referência e Método de Avaliação para Melhoria de Processo de Software – versão 1.0 (MR-MPS e MA-MPS)”, In: IV Simpósio Brasileiro de Qualidade de Software, Porto Alegre.

Estratégia para a Implementação do Modelo CMMI ou MPS-BR na DATAPREV

Rosana Fernandes Osório1

1Empresa de Tecnologia e Informações da Previdência Social - Dataprev

Rio de Janeiro – RJ – Brasil

Rosana.osorio@previdencia.gov.br

Abstract. *This paper describes the strategy adopted for the implementation of the CMMI model (Capability Maturity Model Integration) or MPS.BR (Modelo de Qualidade de Software Brasileiro) in the Units of Software Development of DATAPREV - software factories located in the states of Ceará, Paraíba, Santa Catarina and Rio de Janeiro.*

Resumo. *Este artigo descreve a estratégia adotada para a implementação do modelo CMMI (Capability Maturity Model Integration) ou MPS.BR (Modelo de Qualidade de Software Brasileiro) nas Unidades de Desenvolvimento de Software da DATAPREV - fábricas de software situadas nos estados do Ceará, Paraíba, Santa Catarina e Rio de Janeiro.*

1. Introdução

A Empresa de Tecnologia e Informações da Previdência Social - DATAPREV é uma empresa pública, instituída pela Lei n° 6.125, de quatro de novembro de 1974, vinculada ao Ministério da Previdência e Assistência Social, cuja missão é "*Prover soluções de tecnologia da informação e da comunicação para o êxito das ações de governo, de forma a preservar o interesse público*".

A DATAPREV tem sede e foro em Brasília, Distrito Federal, e filial regional na cidade do Rio de Janeiro, estado do Rio de Janeiro. Organizada em quatro diretorias (Financeira e Comercial; Relacionamento, Desenvolvimento e Informações; Operações e Telecomunicações e Pessoas), possui quatro Unidades de Desenvolvimento (fábricas de software) e vinte e sete Unidades Regionais. Dos três mil cento e sessenta e sete integrantes de seu quadro de pessoal, em torno de seiscentos e vinte e oito estão envolvidos com desenvolvimento de software.

Após duas décadas de promessas não totalmente cumpridas sobre ganhos de produtividade e qualidade a partir da aplicação de novas metodologias de software e novas tecnologias, a indústria e as organizações governamentais estão compreendendo que seu problema fundamental é a inabilidade para gerenciar o processo de software.

Diante deste contexto, a DATAPREV está buscando a adoção do modelo CMMI, assim como outros padrões, tais como MPS.BR, Gestão de Projeto (PMBOK), Pontos de Função (IFPUG), Gerência de TI (ITIL) etc. Não é mais uma questão de importância, prioridade empresarial ou diferencial competitivo, mas sim um pré-requisito básico para as empresas de desenvolvimento de software que pretendem continuar atuando no mercado corporativo, assim como nas áreas de governo.

A implementação do modelo CMMI é vital para que a DATAPREV aprimore o atendimento aos seus clientes, tais como: INSS (Instituto Nacional do Seguro Social), Ministério da Previdência Social (e suas secretarias: Secretaria da Receita Previdenciária, Secretaria da Previdência Complementar e Secretaria de Previdência Social), Ministério do Trabalho e Emprego e Ministério do Desenvolvimento Social e Combate à Fome.

É cediço que a utilização do modelo CMMI já é realizada por alguns dos entes públicos, tais como o Tribunal Superior Eleitoral – TSE (avaliado nível 2 no CMMI).

A proposta deste projeto é que a DATAPREV implante gradativamente as práticas relativas às PAs (*Process Areas*) do CMMI-DEV 1.2 (*Capability Maturity Model Integration For Development, version 1.2*), institucionalizando processos dos projetos de software e possibilitando à organização, desta forma, repetir práticas bem sucedidas em sucessivos projetos.

2. Objetivos e Justificativas

A DATAPREV, como provedora de soluções cujo núcleo frequentemente é um produto de software, necessita de um processo de desenvolvimento organizado. Hoje, os produtos de software desenvolvidos pela organização são operados basicamente no ambiente on-line (web), sendo assim, a qualidade percebida pelos usuários desses produtos é imediatamente afetada quando da ocorrência de uma falha. Surge a necessidade de incorporar características de qualidade durante todo o processo de desenvolvimento, através das boas práticas da engenharia de software descritas no modelo CMMI.

São características importantes a serem consideradas no atual ambiente de desenvolvimento e manutenção de software da DATAPREV:

- A multiplicidade de tecnologias e ferramentas dificulta os esforços para padronização e gestão do ambiente de desenvolvimento.
- A falta de treinamento para os gestores de projeto.
- A decisão da tecnologia a ser usada no processo de desenvolvimento dos projetos de software é baseada muito mais no conhecimento da equipe do projeto do que nas necessidades identificadas no projeto.
- O conhecimento gerado por uma equipe de projeto, durante o processo de desenvolvimento de software, nem sempre é aproveitado por outros projetos similares.
- Frequentemente, a equipe de projeto não é treinada nas ferramentas necessárias ao desenvolvimento do projeto de software.
- Nem sempre utilizamos padrões para avaliação dos serviços executados por terceiros, garantindo que o que foi contratado foi atendido pelo serviço entregue.
- Falta prática na execução de medições e métricas que permitam avaliar sistematicamente o processo de desenvolvimento de software e os produtos de software gerados.

O objetivo deste trabalho é apresentar a estratégia proposta e que está sendo colocada

em prática, para a implementação do Modelo CMMI ou MPS.BR na DATAPREV.

3. Metodologia de Execução

3.1. Fase 1 - Iniciação

A iniciação desse trabalho ocorreu no ano de 2005, quando a DATAPREV criou o Comitê Gestor de Modernização Tecnológica - CGMT. O objetivo desse grupo era ser o agente de mudança, na busca de um novo paradigma de desenvolvimento de software para a DATAPREV. É importante lembrar que contamos com a BRQ – Informática Ltda., apesar de num curto período de tempo, como uma colaboradora nesse processo de mudança.

As seguintes premissas, tratadas como projetos na organização, eram a base deste grupo: nova infra-estrutura de desenvolvimento de software: plataforma J2EE; framework Dataprev e padronização da camada de apresentação; melhoria do processo de desenvolvimento de software: adequação ao RUP – *Rational Unified Process* da atual metodologia de desenvolvimento de sistemas da organização e avaliação da organização no Nível 2 do Modelo CMMI; capacitação das pessoas:criação de um plano de capacitação e projetos piloto: adoção dessas novas tecnologias eram testadas em alguns projetos na organização.

Em atendimento ao processo de mudança da organização, no final do ano de 2005, os diversos grupos de trabalho, sob a gestão desse comitê, tiveram suas atribuições refletidas numa nova estrutura organizacional da DATAPREV. A estrutura antiga (Figura 1) apresentava vários problemas, tais como: esforço duplicado de recursos humanos e materiais, pouco controle quanto ao processo de produção de software, diferentes processos de desenvolvimento (falta de padronização), custo elevado na produção de software, ausência da garantia e controle de qualidade, vários canais de entrada de demandas e o não cumprimento de prazos.

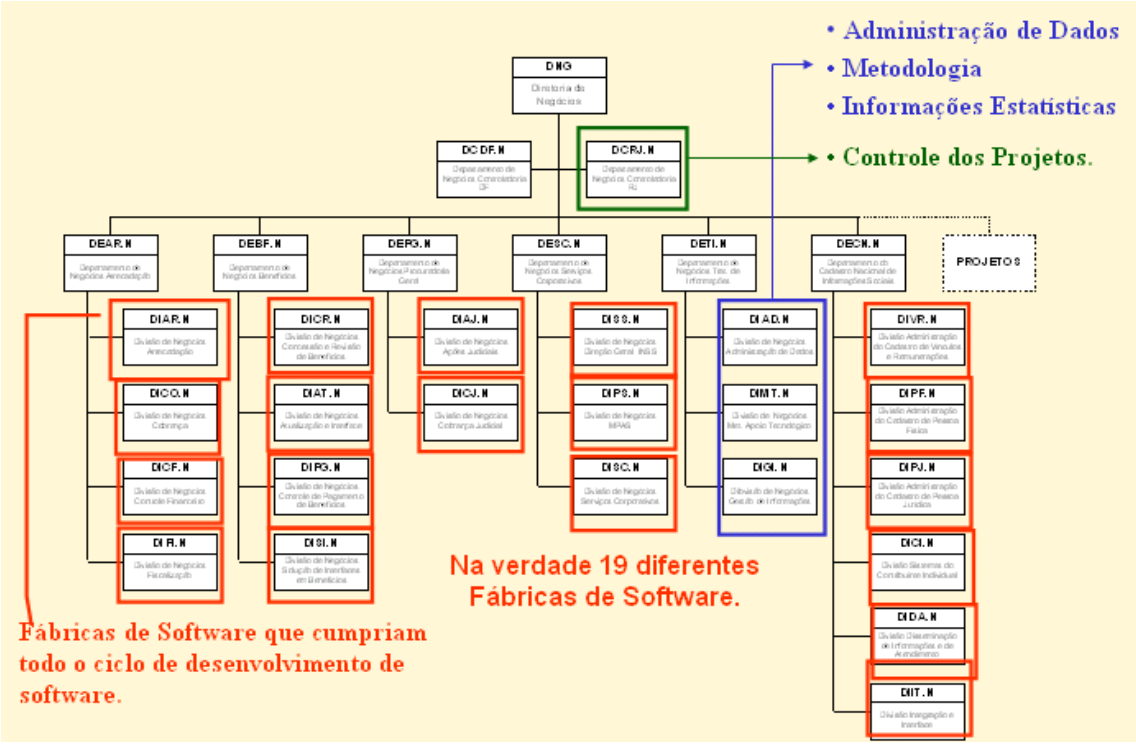


Figura 1. Estrutura Organizacional antes da Mudança

É importante relatar que se buscou resolver alguns desses problemas na nova estrutura organizacional (Figura 2). Nesse momento foi criado o Departamento de Qualidade de Software, com a atribuição de realizar a gestão do processo de software da empresa, garantindo a melhoria contínua da qualidade do processo e dos produtos gerados. Para o modelo (CMMI) processo de software: é um conjunto de atividades, métodos, práticas e transformações utilizadas para desenvolver e manter softwares e seus produtos relacionados.

O Departamento contava com duas áreas:

- **Área de Processo, com a atribuição de** definir e divulgar os processos de software a serem utilizados pelas áreas de desenvolvimento, prestando suporte e avaliando a implementação desses processos, garantindo a aderência às metodologias de desenvolvimento e ao modelo de melhoria de software adotado.
- **Área de Testes, com a atribuição de** planejar, projetar, suportar, executar e gerenciar os testes dos projetos desenvolvidos, realizando, com isso, a verificação e validação dos processos e produtos de software, garantindo a melhoria de sua qualidade.

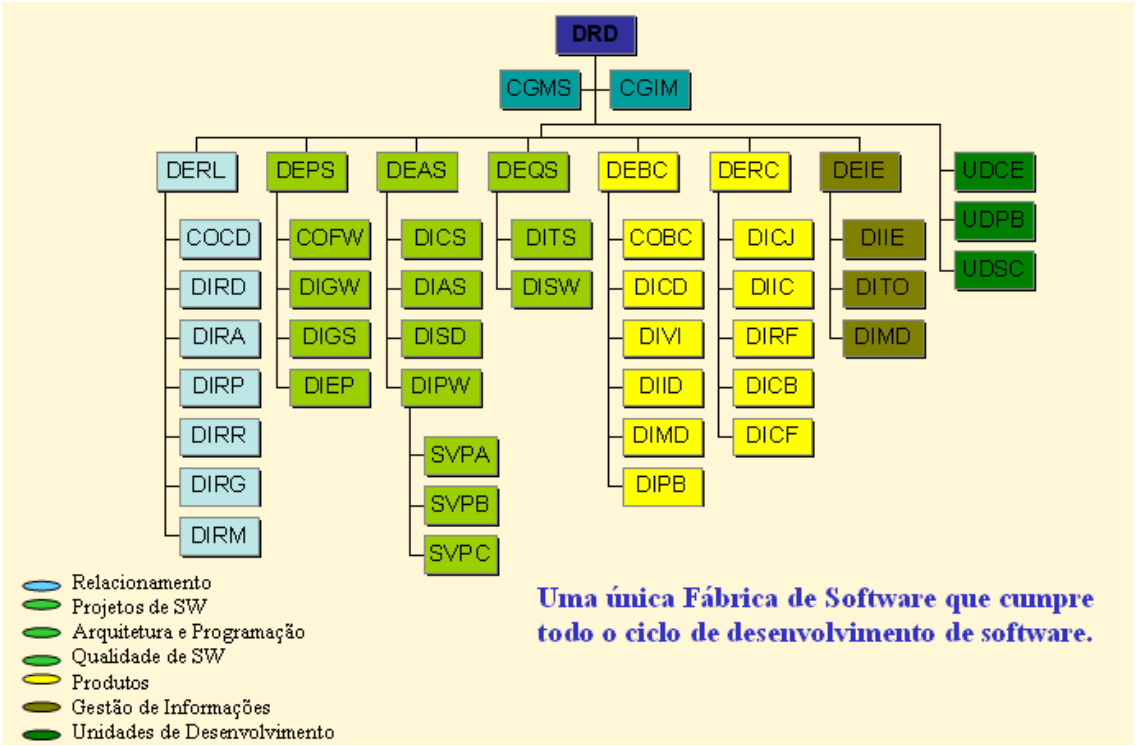


Figura 2. Estrutura organizacional após a primeira re-estruturação da mudança

3.2 Fase 2 - Elaboração

Cabe ressaltar que o ano de 2006 foi um ano marcado pelo planejamento/definição/ aquisição de toda uma infra-estrutura: hardware, software, mobiliário, etc., necessária para operacionalizar a nova forma de a DATAPREV construir seus produtos de software, atendendo a estrutura organizacional (Figura2). As primeiras atividades na busca da definição de um processo de desenvolvimento organizado foram as seguintes:

- Gap Analisys do Processo de Desenvolvimento Atual.
- Criação do primeiro site com os ativos do que viria a ser o novo processo de desenvolvimento.
- Publicação do primeiro processo de desenvolvimento de software da DATAPREV, apresentado em macro-processos: “Desenvolvimento de Sistemas”, “Geração de Informações” e seus processos pertinentes.
- Publicação da atual metodologia de desenvolvimento de sistemas.
- Elaboração de um projeto básico para a contratação de consultoria técnica especializada – implementação do CMMI níveis 2 e 3.
- Elaboração de um projeto básico para a aquisição de suíte integrada para apoio no processo de desenvolvimento de software.
- Definição do pacote de ferramentas pontuais para apoio no processo de desenvolvimento de software (Quadro 1).
- Contratação de 240 desenvolvedores, concurso do ano de 2006.
- Inauguração das três Unidades de Desenvolvimento de Software (Fábricas de Software).

Categorias	Ferramentas
Modelagem de Processos de Negócio	Star UML
Especificação de Requisitos	BROffice
Prototipação	HTML, BROffice
Especificação e Modelagem UML	StarUML 5.0
Modelagem de Dados	Designer 2000
Codificação e Teste Unitário	Eclipse 3.X >
Teste Funcional	BROffice
Deployment e Automação	Maven, Ant
Geracao de Relatório web/J2EE	iReport- Jasper e JFreeChart
Rede, Organograma	Dia

Categorias	Ferramentas
Teste - Automação Funcional	Selenium (automação)
Teste de Desempenho	Jmeter
Integração Contínua	CruiseControl
Cobertura de Código	Cobertura
Controle de Versão	CVS - Tortoise
Gestão de Mudança	Mantis
Gestão de Bugs	Mantis
Tratamento da Conformidade	Mantis
Estimativa e Contagem de PF	BrOffice

Quadro 1. Ferramentas pontuais para apoio ao Processo de Desenvolvimento de Software da DATAPREV.

Vale alertar que algumas das ferramentas pontuais mencionadas seriam utilizadas enquanto a DATAPREV ainda não tivesse adquirido as novas ferramentas para apoiar o processo de desenvolvimento de software. O (Quadro 1) demonstra a versão em uso atualmente, pois tivemos três evoluções referentes ao pacote de ferramentas.

A criação das Unidades de Desenvolvimento de Software da DATAPREV (Figura 2) foi um grande marco desse processo de mudança. Nesse momento, foram criados grupos de qualidade alocados nas Unidades de Desenvolvimento, mas subordinados tecnicamente ao Departamento de Qualidade de Software, contando aproximadamente com três técnicos em cada Unidade.

Como ainda não tínhamos a conclusão do nosso processo de aquisição para a contratação de consultoria técnica especializada – implementação do CMMI níveis 2 e 3 – e com as Unidades de Desenvolvimento prestes a iniciar seu funcionamento, decidiu-se por implementar as práticas do modelo CMMI com o conhecimento atual das pessoas do Departamento de Qualidade de Software e das pessoas que compõem os grupos de qualidade das Unidades de desenvolvimento. Muitas dessas pessoas trouxeram o conhecimento de como implementar o modelo de outras organizações, mesmo assim, alguns treinamentos emergenciais foram necessários e contamos com empresas colaboradoras, tais como: ISD Brasil, SOFTEX, DISMORE ASSOCIATES, ASR – Consultoria e Assessoria em Qualidade Ltda.

Cada grupo ficou responsável pela definição das seguintes áreas de processos: Gerência de Requisitos (REQM); Gerência de Configuração e Mudança (CM); Garantia da Qualidade do Processo e do Produto (PPQA); Medição e Análise (MA); Verificação (VER); Validação (VAL) e pela Adequação ao RUP (*Rational Unified Process*) da atual metodologia de desenvolvimento de sistemas da organização. As áreas de processo de Planejamento de Projeto (PP) e Acompanhamento e Controle de Projeto (PMC) são mantidas por uma área responsável por manter o portfólio dos projetos desenvolvidos na Diretoria de Relacionamento, Desenvolvimento e Informações.

A ordem de implementação das áreas de processo deverá seguir o modelo MPS.BR, como já mencionado antes, pela similaridade entre o CMMI e o MPS.BR. Acredita-se ser essa uma das vantagens, pois o grupo de qualidade já poderá apresentar resultados da implementação do modelo para a organização e para os patrocinadores. O Quadro 2 apresenta a sequência dessas áreas de Processo.

3.3 Fase 3 - Construção

O planejamento e acompanhamento do projeto foram realizados pelo Departamento de Qualidade de Software. Devido a vários estudos realizados, a diretoria patrocinadora optou por realizar uma nova alteração na sua estrutura organizacional, mudança esta que representou um grande impacto na implementação do modelo CMMI na organização. A nova estrutura organizacional extinguiu o Departamento de Qualidade de Software e suas áreas, criando, em seu lugar, a Coordenação Geral de Qualidade de Software, entendendo que a qualidade deveria atuar como uma área de normatização, auditoria e não como executora de uma das etapas do ciclo de vida de desenvolvimento de software, como, por exemplo, a etapa de testes.

Para isso necessitaria de um pequeno grupo de técnicos especializados.

O impacto da mudança no ano de 2007 – diante de um cenário que colocava em risco o trabalho realizado pela área de testes na organização, decidiu-se unir ao grupo de qualidade os responsáveis por manter e disseminar o processo de testes, já que a área de qualidade se firmava como responsável por todo o processo de desenvolvimento de software dos projetos desenvolvidos na tecnologia orientada a objetos. A (Figura 3) apresenta a estrutura organizacional após a segunda re-estruturação da mudança.

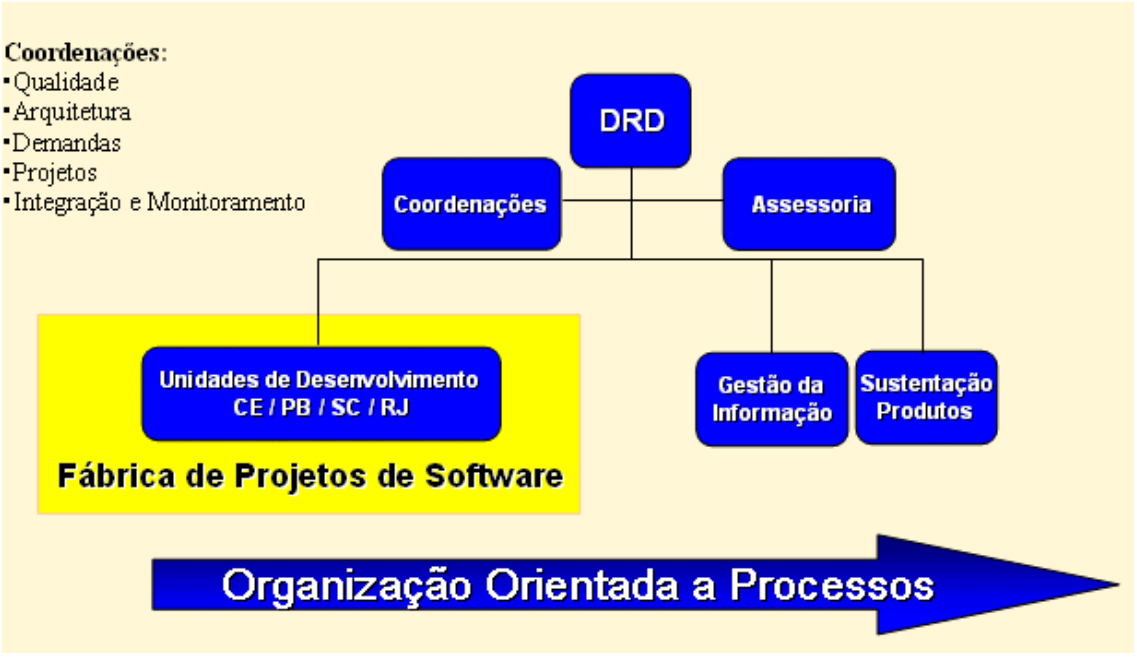


Figura 3. Apresenta a estrutura organizacional após a segunda re-estruturação da mudança

É fato que as mudanças apresentadas na nova estrutura organizacional buscavam melhorar a governança dos projetos de desenvolvimento de software; fortalecer as áreas de Qualidade e Arquitetura, de forma a melhorar o controle do processo de desenvolvimento de software da organização; balancear as áreas de sustentação de produtos na especialização do negócio e em novos clientes e adequar a estrutura de desenvolvimento da nova Unidade de Desenvolvimento criada no Rio de Janeiro às demais Unidades de Desenvolvimento.

Reconhecimento da organização em 2008 :

> Os grupos de qualidade nas Unidades de Desenvolvimento passam a ser reconhecidos como Gerências de Serviço de Qualidade de Software.
>
> Alocação funcional dos serviços de qualidade de software nos estados da Paraíba, Ceará, Santa Catarina e Rio de Janeiro e respectivas equipes de qualidade à Coordenação Geral de Qualidade de Software,

permitindo autonomia funcional das equipes da qualidade para realização de avaliações de conformidade dos projetos em relação aos gestores funcionais dos projetos.

Aprovação do Projeto de Institucionalização do Modelo CMMI na DATAPREV.

3.3 Fase 4 - Transição

É relevante destacar que as Unidades de Desenvolvimento de Software da DATAPREV, já há dois anos em funcionamento, apresentam uma quantidade considerável de novos projetos de software. As Gerências de Serviço de Qualidade, além de executarem suas atividades, continuam dividindo parte de seu tempo na definição e melhoria do processo de desenvolvimento de software da DATAPREV, até que a consultoria técnica especializada – Implementação CMMI níveis 2 e 3 – possa aplicar a sua estratégia, como colaboradora nesse processo de mudança, com o objetivo de implementar os modelos nesta organização.

4. Resultados Obtidos

4.1 Processos que contribuem para melhoria da qualidade de software descritos e institucionalizados.

O Processo de Desenvolvimento de Software da DATAPREV (PD-DATAPREV) é um conjunto de processos necessários para a execução de projetos e serviços de software de diferentes naturezas (desenvolvimento, manutenção e etc.). O PD-DATAPREV está descrito em (http://www-pddataprev) (Figura 4).



Figura 4. Processo de Desenvolvimento de Software da DATAPREV

4.2 Recursos humanos capacitados (especialistas, mestres, doutores, etc.)

Quantidade: um recurso (especialista) - trabalho de conclusão apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-Graduação Lato

Sensu em Melhoria de Processo, para a obtenção do título de especialização.Título do Trabalho: DIAGNÓSTICO DE PROCESSOS À LUZ DO NÍVEL 2 DO

CMMI EM UMA EMPRESA PÚBLICA DE DESENVOLVIMENTO DE SOFTWARE.

5. Aplicabilidade dos Resultados

Relevância **-** anualmente o Ministério de Ciência e Tecnologia divulga pesquisa sobre a Qualidade e Produtividade no Setor de Software Brasileiro, tratando dessa questão sob duas dimensões: produtos e processos. Esse trabalho demonstra claramente a relevância que o governo dá a essas questões. A DATAPREV, como empresa pública, não esta fora desse movimento e trata a qualidade de seus produtos e processos com importância estratégica, de quem precisa oferecer informações confiáveis com a certificação de qualidade.

Abrangência **-** comprometido com a gestão da qualidade, combinando organização de processos, tecnologia e pessoas. Este projeto de abrangência nacional busca a total aceitação por parte de seu corpo técnico, introduzindo a padronização e melhorias do processo de software padrão da organização. Este projeto se mostrou ainda mais necessário com o advento das Unidades de Desenvolvimento: Ceará, Paraíba e Santa Catarina que atuam como Fábricas de Software.

Impacto **-** o processo de desenvolvimento de software da DATAPREV até então se caracterizava pela dificuldade em estabelecer um processo de garantia e controle e qualidade, trazendo como consequência o baixo nível de aderência às práticas de mercado. O impacto observado no desenvolvimento deste projeto, está na forte articulação que os responsáveis por este projeto buscam com as demais áreas, além de um trabalho de disseminação para obter maior mobilização da empresa como um todo.

6. Características Inovadoras

Considerando o universo da DATAPREV, podemos dizer que se trata de um projeto inovador.

Entretanto, mais importante do que ser inovador, ou não, é o fato de ser um caminho fundamental para as pretensões futuras da empresa. Desta forma, estamos conduzindo os trabalhos tendo como foco a criatividade e as disciplinas da engenharia de software, para que a aceitação de todos os envolvidos seja agradável e gradativa, porém efetiva.

7. Conclusão e Perspectivas Futuras

Com relação ao projeto apresentado, entende-se que:

Embora seja difícil de implementar modelos de qualidade de software inicialmente sem apoio de uma consultoria técnica especializada, a DATAPREV conseguiu implementar parcialmente as seguintes práticas das áreas de Processo do Modelo CMMI, nível 2: Processo de Planejamento de Projeto (PP) e Acompanhamento e Controle de Projeto (PMC); Gerência de Requisitos (REQM); Gerência de Configuração e Mudança (CM); Garantia da Qualidade do Processo e do Produto (PPQA); Medição e Análise (MA).

É importante destacar que um ponto forte da DATAPREV no processo de avaliação nível 2 do modelo CMMI seriam os testes, pois alguns tipos de testes já são executados na organização.

Apesar dos testes estarem representados no modelo CMMI nas áreas de processo do nível 3 Verificação (VER); Validação (VAL), podem ser considerados pontos fortes num processo de avaliação nível 2 do modelo CMMI.

Uma das grandes vantagens na implementação do modelo CMMI na DATAPREV foi termos iniciado na definição da área de processo Definição do Processo Organizacional (OPD). Apesar das várias alterações que vêm ocorrendo no processo, hoje os desenvolvedores já sabem onde obter informações, tais como procedimentos, documentos de orientação, templates, checklists, etc.

As novas perspectivas estão incorporadas no projeto de Institucionalização do Modelo CMMI na DATAPREV, que inclui avaliações intermediárias no modelo MPS.BR e a contratação da consultoria técnica especializada.

8. Referências Bibliográficas

SEI/CMU (2006) “CMMI for Development”, Version 1.2, CMU/SEI-2006-TR-008, Pittsburgh - PA, 2006.

MPS.BR (2007) “Guia Geral”, Versão 1.2, SOFTEX.

FAPS - Ferramenta de Suporte à Avaliação de Processos de Software Alinhada com os Modelos CMMI e MPS.BR

Marcello Thiry[1 2], C. G. Gresse von Wangenheim[1 2], Alessandra Zoucas[1 2], Leonardo Reis Tristão[1], José Luiz da Silveira[1]

[1] Universidade do Vale do Itajaí (UNIVALI) – Campus São José
Mestrado em Computação Aplicada
São José, SC - Brasil

[2] Incremental Tecnologia em Informática
Florianópolis, SC - Brasil

{marcello.thiry, gresse, alessandra.zoucas, leutristao, joseluizsilveira2}@gmail.com

Resumo. *Este trabalho apresenta uma ferramenta para apoiar uma avaliação de processos de software que reúne os requisitos de modelos de referência reconhecidos nacional e internacionalmente como os modelos MPS.BR e CMMI-DEV. A ferramenta FAPS foi inicialmente desenvolvida em compatibilidade com os métodos de avaliação MA-MPS e SCAMPI. Uma nova versão da ferramenta foi desenvolvida para apoiar também avaliações integradas (FAPS-INT), ou seja, apoiar a análise da situação de uma organização em relação a um modelo a partir dos resultados de uma avaliação usando outro modelo. Esta avaliação integrada foi baseada no método MARES-INT. Para uma verificação inicial da ferramenta, foi realizada uma avaliação integrada com o apoio da ferramenta considerando os resultados de uma avaliação MPS.BR nível G em paralelo com as práticas mapeadas do CMMI-DEV.*

1. Introdução

Muitas vezes, no início, durante e no final de um programa de melhoria são executadas seções de avaliação de conformidade dos processos de software da unidade organizacional foco da melhoria, também chamadas de *GAP analysis*. Estas seções de avaliação buscam identificar e registrar a capacidade atual dos processos em relação a um modelo de referência. Desta forma, é possível verificar não apenas a aderência da organização em relação a um modelo ou norma, mas a evolução de sua melhoria, salientando pontos fortes e fracos para orientar a continuação do trabalho de implementação.

Tipicamente, estas avaliações analisam a implementação dos processos em conformidade a apenas uma norma ou modelo de referência. Visando executar um processo de avaliação que atenda modelos reconhecidos nacionalmente e internacionalmente em paralelo, foi desenvolvido o MARES-INT (WANGENHEIM et al, 2005), o qual é um modelo de avaliação, contendo diretrizes para a aplicação e adaptação na prática, que integra os modelos de referência do CMMI, ISO/IEC 15504-5 e o MR-MPS. Além disso, o MARES-INT também estabelece um método de avaliação com guias de adaptação que pode ser utilizado para a melhoria de processo no contexto de diferentes tipos de avaliação (avaliação de contextualização e/ou avaliação detalhada), em empresas de diferentes portes e em conformidade com os requisitos de uma avaliação alinhada e satisfazendo os requisitos de métodos de avaliação como o CMMI (ARC) V1.2 (SEI, 2006b), com a ISO/IEC 15504-2 e com o MA-MPS (SOFTEX, 2007). O uso adequado do MARES-INT possibilita realizar avaliação de processos de forma integrada podendo reduzir seu esforço e duração e aumentar valor dos resultados.

2. Objetivos e Justificativa

Geralmente, durante a execução de um processo de avaliação de conformidade, há um esforço relevante com a gerência de documentos, tanto na fase preparatória, de coleta de dados, como na documentação dos resultados. Ocorre que muitos dados são utilizados em mais de um documento, contendo informações interrelacionadas, que tipicamente são redigitadas. Em uma avaliação integrada o volume destas informações aumenta consideravelmente.

Portanto, é relevante o desenvolvimento de trabalhos que venham apoiar a gerência dos documentos relacionados consumidos ou produzidos durante uma avaliação e que semi-automatize procedimentos quando possível. Atualmente, não existe uma ferramenta que suporte uma avaliação alinhada ao MPS.BR e a outro modelo de referência simultaneamente. Logo, uma ferramenta que tenha um mecanismo de avaliação integrada e que suporte o trabalho das equipes de avaliação pode contribuir para a agilidade e gerenciamento e processo de tomada de decisão.

Uma ferramenta alinhada com mais de um modelo de referência permite a realização de avaliações integradas, onde os artefatos coletados na fase preparatória de uma avaliação poderão ser reutilizados para evidenciar a execução de práticas de um segundo e terceiro modelo de referência, sem a necessidade de serem coletados novamente. Isto oferecerá maior flexibilidade e abrangência para a equipe de avaliação.

Esta ferramenta também poderá apoiar o trabalho de uma equipe de avaliação na condução de avaliações recorrentes em uma mesma unidade organizacional, onde os artefatos coletados para a primeira avaliação de

um determinado projeto ficarão armazenados no banco de dados da ferramenta e não precisarão ser coletados novamente para a realização de avaliações futuras. Além de apoiar a equipe avaliadora, a ferramenta permitirá também que a própria organização possa realizar avaliações periodicamente e orientar seu processo de melhoria.

Também é esperado que com a ferramenta adequada, uma equipe de avaliação possa ser composta por avaliadores com menos experiência e ainda assim, obter resultados relevantes. Além disso, o suporte computacional oferecerá um controle histórico mais detalhado da seqüência de avaliações, permitindo um acompanhamento efetivo no processo de melhoria de uma determinada empresa.

3. Metodologia de Execução

De acordo com o método científico, a natureza deste trabalho foi uma pesquisa aplicada, objetivando gerar conhecimentos para aplicação prática dirigida à solução de problemas específicos. Para alcançar os objetivos propostos, este trabalho utilizou a seguinte estratégia de desenvolvimento:

Estudo da literatura: os modelos CMMI e MPS.BR foram estudados assim como seus respectivos métodos de avaliação, SCAMPI e MA-MPS, incluindo os documentos utilizados por eles. Desta forma, houve uma base referencial para a modelagem da ferramenta.

Análise de experiências: foram analisadas experiências práticas e melhores práticas na condução de avaliações de processos de software. Para o contexto deste trabalho, foram consideradas as experiências adquiridas em avaliações realizadas pelo LQPS da UNIVALI Campus São José, além de experiências relatadas na literatura. O objetivo desse estudo de experiências foi criar uma base sistemática para o desenvolvimento da ferramenta, orientada pelas reais necessidades das equipes de avaliação na prática.

Estudo do método MARES-INT: o Método Integrado de Avaliação de Processos de Software MARES-INT tem por objetivo auxiliar na avaliação de processos em empresas de software brasileiras. O estudo deste método foi necessário para estabelecer um mapeamento entre os modelos CMMI-DEV e MPS.BR, e preparar a ferramenta com um mecanismo de comparação dos resultados da avaliação.

Desenvolvimento da ferramenta de avaliação: a ferramenta de avaliação foi desenvolvida para abranger todas as etapas do processo de avaliação, considerando também aspectos de usabilidade e restrições em relação aos recursos necessários para garantir sua utilização durante a avaliação. A metodologia de desenvolvimento de software foi baseada no modelo iterativo simplificado, incluindo as atividades de Análise de Requisitos, Projeto (Design), Implementação e Testes. Ela foi desenvolvida em Java 1.6, utilizando o ambiente Netbeans 6.1 e a arquitetura adotada foi cliente/servidor.

Avaliação por meio de um estudo de caso: a ferramenta FAPS foi avaliada inicialmente por meio de um estudo de caso em uma MPE de software, durante uma avaliação com dados reais. O estudo de caso foi planejado explicitamente, definindo a forma do estudo e as medidas relevantes a serem observadas durante a mesma. Durante a avaliação foram coletados os dados relevantes com base no plano de medição definido.

4. Resultados Obtidos

Inicialmente, como resultado de uma parceria entre o LQPS (Laboratório de Qualidade de Software) da UNIVALI (Universidade do Vale do Itajaí) e a Instituição Implementadora do MPS.BR Incremental Tecnologia, foi desenvolvida uma ferramenta para apoiar avaliações de processos de software denominada FAPS. O registro de Patente para a ferramenta desenvolvida ainda está em estudo e deverá ocorrer durante o ano de 2009.

A FAPS foi desenvolvida considerando o método de avaliação MA-MPS, os requisitos estabelecidos no CMMI-ARC. Embora, na sua versão inicial, a ferramenta ainda não permitisse avaliações integradas, o atendimento destes requisitos ofereceu o suporte necessário para a implementação de avaliações integradas baseadas nos modelos MPS.BR e CMMI. A partir da sua segunda versão (FAPS-INT), é possível obter, de modo semi-automático, a conformidade de uma avaliação realizada com base na norma ou modelo escolhido em relação a outro modelo. Também, é possível ter apoio no gerenciamento dos documentos utilizados durante uma avaliação e auxílio para avaliações recorrentes em uma mesma unidade organizacional. A ferramenta FAPS-INT permite a reutilização de evidências coletadas e de resultados da avaliação de um dos modelos e norma suportados pela ferramenta.

Após uma avaliação, a FAPS-INT permite que um controle histórico da seqüência de avaliações de uma mesma unidade organizacional seja armazenado. Isto auxilia a manter *baselines* do processo, que posteriormente podem vir a ser comparadas estabelecendo um parâmetro de quanto de melhoria os processos de software adquiriram com o trabalho de implementação. A estrutura da ferramenta não apresenta nada específico para uma avaliação integrada, uma vez que os avaliadores não precisam se preocupar com esta possibilidade. Na visão do avaliador, ele irá conduzir uma avaliação usual selecionando apenas um dos modelos disponíveis. Ao final da avaliação, após a caracterização dos resultados é que o avaliador poderá optar por verificar a correspondência com outro modelo a ser selecionado. Desta forma, o avaliador não precisa nem estar familiarizado com o segundo modelo/norma.

Após cadastrar informações básicas sobre a avaliação, a unidade organizacional pode iniciar o cadastramento das evidências. Na figura 1, pode-se visualizar 4 evidências cadastradas para o resultado GPR7 do processo Gerência de Projetos do MPS.BR. Neste exemplo, está sendo considerado o nível G de maturidade. Os processos com seus respectivos resultados são apresentados na parte esquerda da tela, assim como os atributos de processo também com seus resultados. O usuário pode selecionar qualquer resultado para visualizar as evidências a ele relacionadas. Na parte superior da tela, é apresentado o resultado selecionado com seu detalhamento. A estrutura central é similar àquela encontrada nas planilhas de avaliação do MA-MPS e do SCAMPI. Além da coluna para descrever a evidência, existem colunas para indicar o tipo de evidência (direta, indireta ou afirmação) e a fonte da evidência. Depois, pode-se verificar a existência da coluna organizacional e uma coluna para cada projeto cadastrado (o número de projetos pode ser variável). As evidências cadastradas são também armazenadas em um repositório (parte inferior direita da tela), permitindo sua reutilização em outros resultados.

Durante a avaliação apoiada pela FAPS-INT, é possível alterar a cor das colunas organizacional e projetos. A cor branca significa que a evidência ainda não foi avaliada, a cor verde indica que a evidência foi julgada adequada, a cor amarela indica alguma fraqueza na evidência e a cor vermelha é usada para indicar que a evidência não está adequada. A figura 1 mostra como as cores podem ser alteradas em cada célula da planilha.

No rodapé da tela, é possível visualizar três botões. Eles permitem armazenar os achados (pontos fortes, fracos e oportunidades de melhoria). Além dos achados do resultado visualizado, a ferramenta permite também que sejam cadastrados os achados do modelo e os achados gerais. Estes achados podem ser visualizados nas abas correspondentes a qualquer momento durante a avaliação. Vale comentar que se um resultado específico não estiver selecionado, o primeiro botão não será habilitado. Dando seqüência ao processo de avaliação, a equipe de avaliadores pode registrar também na FAPS a caracterização de cada resultado e processo sendo avaliado.

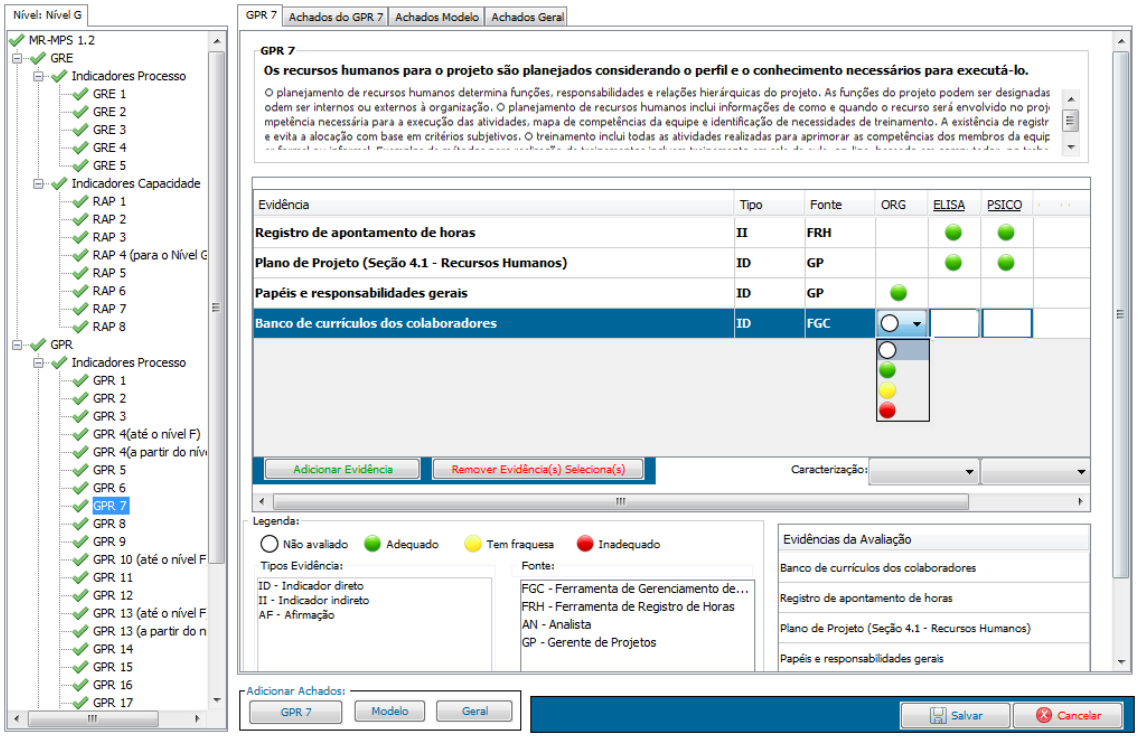


Figura 1: Tela de evidências e avaliação (exemplo com o resultado esperado GPR7 do MPS.BR)

Apesar do exemplo aqui apresentado ser referente ao modelo MR-MPS, é possível também criar uma avaliação CMMI ou ISO/IEC 15504. Uma vez que a FAPS-INT permite editar os perfis de capacidade de processo, é possível ainda utilizar uma avaliação com representação contínua. Ou seja, os avaliadores podem selecionar os processos que desejam avaliar, bem como o grau de capacidade esperado em cada processo.

A primeira publicação de resultados da ferramenta foi durante o Workshop de Avaliadores do MPS.BR, realizado em novembro de 2008, na cidade de Campinas/SP. Um artigo para o SBQS 2009 também foi submetido e está aguardando o resultado da avaliação.

5. Aplicabilidade dos Resultados

Como base para uma avaliação da ferramenta na prática, foi utilizada a planilha de avaliação de uma empresa que participou de um programa cooperado MPS.BR, coordenado pela ACATE (Associação Catarinense de

Empresas de Tecnologia) e pela Instituição Implementadora Incremental Tecnologia. Esta empresa (Poligraph) foi avaliada com sucesso no nível F do MPS.BR.

Para esta primeira aplicação da ferramenta, foram considerados apenas os processos Gerência de Projetos (GPR) e Gerência de Requisitos (GRE), simulando uma avaliação nível G. Entretanto, apesar da redução do escopo, todas as evidências com seus respectivos links foram cadastrados. Seguiu-se o método MA-MPS, onde inicialmente, as evidências foram cadastradas sem a respectiva caracterização. Uma vez que as evidências utilizadas já foram aquelas utilizadas na avaliação final da empresa, não houve a etapa de ajuste da planilha. Entretanto, foram simulados inclusões e exclusões. Após o cadastro de todas as evidências, a ferramenta FAPS-INT foi utilizada para a avaliação das evidências. Sem considerar ainda a caracterização dos resultados, cada evidência foi julgada utilizando o mesmo padrão de cores da planilha oficial.

Os achados para resultado de processo foram cadastrados também neste momento. Vale lembrar que eles já haviam sido estabelecidos pela avaliação final oficial. Somente após o julgamento da adequação das evidências, o que permitiu simular a avaliação inicial, iniciou-se o processo de caracterização dos resultados. Inicialmente, para cada projeto e depois para cada resultado de processo, resultado de atributo de processo e para cada atributo de processo. Os achados do modelo e gerais foram cadastrados neste momento, antes da caracterização final do grau de implementação dos processos na unidade organizacional. Ao final, os processos foram caracterizados e o nível MPS.BR foi atribuído. Durante todo o processo, foram identificados alguns ajustes que resultaram em mudanças, principalmente relacionadas com a usabilidade. Como a avaliação da ferramenta foi realizada por um avaliador credenciado, foi possível identificar algumas facilidades que permitiram maior agilidade no manuseio dos dados da planilha.

Com a planilha totalmente preenchida na ferramenta, foi possível avaliar o mapeamento entre os indicadores. Para esta primeira aplicação, foi utilizado apenas o mapeamento com o CMMI, considerando as áreas de processo equivalentes ao nível G do MPS.BR. Portanto, foram considerados as áreas de processo PP (Planejamento de Projeto), PMC (Monitoração e Controle de Projeto) e REQM (Gerência de Requisitos).

A versão atual da ferramenta não possui ainda uma interface para visualizar o mapeamento. Deste modo, foi utilizada uma consulta direta no banco de dados da ferramenta para verificar se a ferramenta conseguiria mostrar o relacionamento entre os indicadores. A partir da execução da consulta, verificou-se que a ferramenta FAPS-INT conseguiu mostrar a relação entre os indicadores do MPS.BR e do CMMI a partir do mapeamento, indicando o grau de implementação.

Nesta aplicação da avaliação integrada, foram considerados apenas os indicadores de processo (resultados esperados de processo do MPS.BR com as práticas específicas do CMMI-DEV). Outra limitação foi que, na versão atual, um indicador de um modelo é associado com apenas um indicador do outro modelo. Isso pode apresentar algumas distorções e uma solução baseada em percentuais de similaridade está sendo atualmente estudada. Entretanto, percebeu-se que é possível realizar avaliações integradas a partir de uma única avaliação em um modelo específico. A questão é melhorar a qualidade e flexibilidade do mapeamento.

6. Características Inovadoras

Atualmente, existe uma carência em ferramentas que apóiem avaliações no modelo MPS.BR. Neste sentido, a ferramenta FAPS (em sua versão atual FAPS-INT) oferece uma inovação ao auxiliar organizações durante o avaliações, permitindo o gerenciamento de documentos, armazenamento e recuperação de achados. Sejam estas organizações instituições implementadoras ou as aquelas que pretendem ser avaliadas, ou ainda aquelas que buscam somente a melhoria contínua de seus processos.

Além disso, o fato da ferramenta ter sido desenvolvida em ambiente cliente-servidor com uma instalação simplificada (cópia direta de um diretório) reduz necessidades específicas como uma infra-estrutura de rede ou acesso à Internet.

A avaliação pode ser conduzida por equipe composta por um número variável de avaliadores, assim como determinam os métodos de avaliação. O recurso de importação/exportação facilita o gerenciamento das informações coletadas por diferentes mini-equipes e reduz a possibilidade de erros com cópias de planilha.

Ainda, a ferramenta oferece um repositório de resultados de avaliações, permitindo que uma organização possa manter um histórico do seu programa de melhoria, identificando e comparando resultados em diferentes momentos.

Outro aspecto inovador está na própria avaliação integrada. Em uma mesma avaliação, é possível determinar de modo semi-automatizado a situação dos processos de uma organização nos modelos MPS.BR e CMMI-DEV.

7. Conclusão e Perspectivas Futuras

A ferramenta desenvolvida é significativa no contexto de MPEs brasileiras, oferecendo suporte para a realização de avaliação de processos de software e avaliações integradas. Ela permite que as avaliações sejam executadas de forma mais adequada, flexível e abrangente. A ferramenta desenvolvida tem como foco

apoiar a organização das atividades de avaliação de processos de software, gerenciar os documentos gerados no processo de avaliação de processos de software, facilitando o trabalho dos avaliadores.

O objetivo do desenvolvimento da FAPS não foi apenas apoiar equipes de avaliação, mas também equipes de implementadores, uma vez que, durante a implementação de um programa de melhoria é fundamental o acompanhamento da evolução da organização em relação à aderência dos processos que estão sendo implementados. Desta forma, os implementadores poderão identificar novas ações de melhoria a partir de informações quantitativas.

Um piloto de avaliação com o apoio da ferramenta foi conduzido pela equipe da Incremental nas dependências do LQPS e os resultados iniciais foram satisfatórios. Embora a validação da ferramenta esteja no início, já é possível identificar uma tendência de adequação para apoiar a condução de avaliações de processos de software. Vale ressaltar que a FAPS também foi desenvolvida para ser usada pela unidade organizacional em avaliação, uma vez ela permite o cadastramento prévio de todas as evidências e links. A metáfora para este cadastramento foram as planilhas dos métodos SCAMPI (SEI, 2006c) e MA-MPS (SOFTEX, 2007).

Com a divulgação e disponibilização desta ferramenta é esperado que as atividades do processo de avaliação demandem menos esforço da equipe de avaliação. Também se espera que a própria organização consiga realizar avaliações periódicas na unidade organizacional foco da melhoria de processos de software, para nortear seu processo de melhoria. Embora a infra-estrutura para a avaliação integrada já faça parte da FAPS-INT, falta ainda desenvolver o módulo que apresente os resultados nos diferentes modelos e norma. Atualmente, isso só é possível por acesso direto aos dados da ferramenta.

Outra funcionalidade prevista é a preparação da apresentação dos resultados da avaliação, incluindo a caracterização dos processos, pontos fortes, pontos fracos e oportunidades de melhoria diretamente pela ferramenta ou via relatório. Desta forma, o avaliador líder não precisará repassar as informações para uma apresentação específica, utilizando diretamente a FAPS-INT.

A validação das funcionalidades da ferramenta FAPS está sendo iniciada com outras aplicações práticas de avaliação executadas pelas equipes do LQPS e da Incremental Tecnologia para apoiar avaliações de conformidade com os modelos MPS.BR e CMMI, com a geração de resultados de avaliações integradas. Inicialmente, a ferramenta será disponibilizada sem a parte da avaliação integrada que ainda precisa de mais testes. Porém, pretende-se disponibilizar uma versão funcional para avaliações específicas nos modelos MPS.BR e CMMI já no primeiro semestre de 2009.

A ferramenta FAPS foi construída com uma concepção *stand-alone,* não havendo a necessidade de estar conectada à rede. Embora este requisito seja relevante para permitir que os avaliadores possam realizar avaliações em qualquer lugar, sem a necessidade de acessar a rede da organização avaliada, os resultados das avaliações acabam ficando espalhados, dificultando a análise de resultados históricos. O registro histórico permite o acompanhamento efetivo sobre a evolução dos processos, oferecendo indicadores de melhoria da organização. Neste sentido, estamos elaborando um novo projeto busca construir uma infra-estrutura computacional para reunir todos os resultados das avaliações realizadas e disponibilizar as informações via um portal Web da organização. O objetivo não é somente oferecer a centralização das informações, mas criar um ambiente com indicadores de melhoria que possam ser utilizados para apoiar a medição efetiva da melhoria.

Agradecimentos

Os autores gostariam de agradecer à empresa Poligraph (MPS.BR nível F) de Florianópolis que permitiu que os dados de sua planilha de avaliação fossem utilizados para este trabalho. Este trabalho foi apoiado pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), pela UNIVALI (Universidade do Vale do Itajaí) e pela Incremental Tecnologia.

8. Referências Bibliográficas

SOFTWARE ENGENEERING INSTITUTE (SEI). ***Appraisal Requirements for CMMI, Version 1.2*** (ARC, V1.2). Software Engeneering Institute, Carnegie Mellon University, Pittisburg, Pennsylvania, 2006b.

SOFTWARE ENGENEERING INSTITUTE (SEI). ***Standard CMMI Appraisal Methos for Process Improvement*** (SCAMPI[SM]) A, Version 1.2: Method Definition Document. Software Engeneering Institute, Carnegie Mellon University, Pittisburg, Pennsylvania, 2006c.

SOFTEX (2007). ***MPS.BR - Guia de Avaliação, versão 1.1***, Associação para Promoção da Excelência do Software Brasileiro (SOFTEX). Junho, 2007. Disponível em: www.softex.br.

WANGENHEIM, C.; THIRY, M.; ZOUCAS, A.; SALVIANO, C. (2005). ***Modelo de Avaliação Integrado CMMI-SE/SW, ISO/IEC 15504, e MPS.BR***. Working Paper LQPS, UNIVALI, São José, 2005.

Ambiente da Qualidade de Produto de Software

Regina Thienne, Ana Guerra, Marbilia Sérgio, Márcia Pimenta

Centro de Tecnologia da Informação Renato Archer-CTI
Rodovia Dom Pedro I km 143,6 Campinas SP – Brasil
Divisão de Qualificação em Software

{regina.thienne, ana.guerra, marbilia.sergio, marcia.pimenta}@CTI.gov.br}

Resumo. *Este artigo descreve a importância da avaliação de produtos de software. Apresenta a metodologia de Avaliação MEDEPROS® desenvolvida pela Divisão de Qualificação em Software – DQS do Centro de Tecnologia da Informação Renato Archer - CTI, sua estrutura teórica básica de Avaliação, a experiência prática no mercado brasileiro. Apresenta como pode ser tratada a avaliação da conformidade dos diferentes tipos de software e conclui a respeito do custo beneficio de avaliações genericamente. Para o ciclo de projetos do PBQP o foco trabalhado no ano de 2008 foi o projeto de Qualidade de Produto de Software e seus resultados.*

1. Informações Gerais

Na sociedade da informação, importância de sistemas de informações baseados em software no mundo moderno não pode ser subestimada. Software está presente em produtos e serviços de todos os tipos, nos negócios, na indústria, no governo e no dia-a-dia dos indivíduos de nossa sociedade. Setores econômicos como manufatura, finanças, comunicações, saúde, energia, transportes, educação e gestão pública dependem do uso de software para a condução de suas operações diárias, utilizando-se desde aplicações em computadores pessoais até grandes sistemas em redes de complexidade gigantesca.

Não é exagero dizer que o software tornou-se um recurso vital para o desenvolvimento social, para o bem estar e para a competitividade de qualquer nação.

A sociedade cada vez mais depende de software, porém os históricos problemas associados ao seu desenvolvimento, ainda não foram adequadamente solucionados, seja em tempo de projeto ou quando em uso no mercado.

O setor de software é uma das prioridades do governo brasileiro na área de Tecnologia da Informação. As áreas de pesquisa vêem atuando no sentido de melhorar os produtos de software para o mercado interno e também para exportação, e para isso é necessário padrões internacionais. Algumas das iniciativas em andamento no CTI com o objetivo de apoiar as empresas exportadoras de software a aumentarem sua competitividade no mercado nacional e internacional; o Subcomitê de Software da Associação Brasileira de Normas Técnicas, com o objetivo de gerar normas brasileiras relacionadas com software; e o Subcomitê de Software do Programa Brasileiro de Qualidade e Produtividade do Ministério da Ciência e Tecnologia - MCT, com o objetivo de estimular, articular, orientar e apoiar os esforços da sociedade na busca da competitividade internacional no setor de software. A realidade reflete uma necessidade crescente de conhecimento especializado capaz de se materializar em práticas efetivas de melhoria da qualidade de produtos de software nas organizações. O ponto de partida comum a estas iniciativas é verificar o quanto os produtos de software estão em conformidade com as normas existentes no mercado. Buscar novas formas de qualificar os produtos de software de acordo com as padronizações vigentes motivam para o desenvolvimento e uso de novos métodos de avaliação.

No contexto internacional, a situação é a mesma do Brasil. Existe a necessidade de avaliar a conformidade dos produtos de software, sendo que esta área tem as primeiras iniciativas e ações.

2. Breve Histórico

Desde 1993 o CTI atua na área de Qualidade de Software, tem aplicado tecnologias para auxiliar interessados na questão da melhoria da qualidade do produto final e do desenvolvimento de ambientes e sistemas. Sendo que a tendência é qualidade nas aplicações de Tecnologia da Informação (TI) para a sociedade.

Como resultado de um esforço internacional de padronização, em 1991 foi publicada a primeira versão da Norma ISO/IEC 9126[1], que define as características de qualidade de software, colocando o software na perspectiva universal de uso de métodos, técnicas e ferramentas de avaliação de qualidade. Dois projetos se destacaram nesse contexto:

¨ Projeto SCOPE (Comunidade Européia) que desenvolveu métodos e ferramentas para avaliação e certificação de produtos, que resultou num conjunto de Normas na área de Qualidade de Produto de Software [3] e,

¨ Projeto SPICE (Estados Unidos, Europa, Japão, entre outros) cujo objetivo foi a geração de normas internacionais para avaliação do processo de desenvolvimento de software, que se consolidou com a publicação da norma internacional ISO/IEC 15504[8].

O CTI interessado em desenvolver conhecimentos nesta área, identificou inicialmente a oportunidade de trabalhar com as normas de Avaliação de Produto, resultado do projeto SCOPE1, através da sua participação na comissão julgadora do Prêmio ASSESPRO em 1994 [10]. Naquela ocasião, a ASSESPRO apresentou a necessidade de estruturar um modelo de avaliação dos produtos concorrentes, e em conjunto com o CTI buscaram recursos e desenvolveram um método, que originou mais tarde no método MEDE-PROS®[2].

Este projeto possibilitou a criação e a capacitação de uma equipe no CTI para elaborar o método de avaliação da qualidade de produto de software, estabelecendo um processo contínuo para divulgar tal tecnologia para a comunidade nacional e internacional de produtores, compradores, usuários de software [6], entre outros.

Desde então, o CTI atua no desenvolvimento de tecnologias na área, o que incluiu como resultados; o desenvolvimento de um ambiente de apoio à avaliação que é composto de uma Base de Dados – SISAVAL (Sistema de Administração de Avaliações MEDE-PROS – versão 3.0), uma Lista de Verificação automatizada, a BMA – Biblioteca de Módulos de Avaliação que é uma base contendo métodos para a avaliação de produtos e o módulo de análise estatística dos resultados quando da aplicação de um dos seus métodos. Além de contribuir, divulgando esta tecnologia através de várias dissertações, publicações, palestras, cursos dentre outros, estimulando assim uma cultura de qualidade nos produtos de software [4].

Para garantir a contínua evolução do método de qualidade de produtos de software, o CTI assinou em 1997, com duração de 3 anos, um termo de cooperação com empresas e universidades da Comunidade Européia (Itália, Dinamarca, Inglaterra) no âmbito do Projeto SQUAD - *Software Quality Accross Diferent Regions* – para troca de experiências na área de Avaliação de Qualidade de Software. De 1997 a 2000 junto do Projeto SQUAD, participaram mais dois parceiros: Venezuela, Chile, além de dois grupos brasileiros, sendo um deles o CTI, através dos membros da DQS.

De 2000 até o momento, o CTI tem feito transferência de tecnologia para laboratórios interessados em avaliar conformidade de produtos de software, os quais são chamados de laboratórios credenciados, além disso, existe um trabalho de pesquisa que atua na melhoria contínua do método, sendo sua última versão atualizada em 2006. Estes trabalhos norteiam a atuação das pesquisas em tecnologia de software orientada para a sociedade, e têm como meta, auxiliar interessados na obtenção de produtos com maior qualidade, usabilidade e visibilidade de mercado.

3

[3] 1 SCOPE - *Software CertificatiOn Programme in Europe*

3. Método de Avaliação da Qualidade de Produto de Software, MEDEPROS

O MEDE-PROS® - Método de Avaliação de Qualidade de Produto de Software foi desenvolvido para avaliar a Qualidade de Produto de Software sob o ponto de vista de um usuário final, tendo como referência as Normas Internacionais: ISO/IEC 9126 [11] (NBR 13596 – Engenharia de Software - Avaliação de produto de software - Características de qualidade e diretrizes para o seu uso - Abr. 1996) e NBR ISO/IEC 121192 - Engenharia de Software - Pacotes de Software - Testes e Requisitos de Qualidade [12], que definem as 6 (seis) características de qualidade de software que devem estar presentes em todos os produtos: Funcionalidade, Confiabilidade, Portabilidade, Usabilidade, Eficiência e Manutenibilidade, e requisitos de qualidade de um pacote de software.

Esse método encontra-se registrado na Fundação Biblioteca Nacional, sob o número 135.620, livro 216, folha 84 e com o registro de marca no INPI sob o número 820166243.

4. Tipos de avaliação

Produtos de software são classificados de acordo com o grau a que o adquirente pode especificar as características do produto. São eles assim classificados na norma IEEE- 1062[7]: *Commercial Off The Shelf* (COTS), comprados com requisitos previamente estabelecidos, *Modified Off The Shelf* (MOTS), quando o cliente pode customizar o [4]produto e *Fully developed* (FD) produto inteiramente desenvolvido de acordo com os requisitos do cliente. A tabela 1 mostra mais características dessa classificação.

Características	COTS	MOTS	FD
Escopo	Fixo	Parcial/e customizado	Total/e customizado
Adequação ao uso	Demonstrado	Demonstrado em aplicações similares	Sem precedentes
Manutenção	Sem controle	Controle parcial	Controle total
Prazo de entrega	Imediato	Pequeno-Grande	Grande
Custo	Baixo-Médio	Médio-Alto	Alto
Qualidade	Não controlada	Parcial/e controlada	Controlada em suamaior parte

Tabela 1- Caracterização do IEEE 1062-*Recommended Practice for Software Acquisition*

O MEDE-PROS® fornece um modelo de qualidade, um método de avaliação baseado nas normas NBR ISO/IEC 9126-1, 12119 e 14598-5. Este método compõe também um conjunto consistente de orientações para realização de uma avaliação. O modelo de qualidade deve definir quais as características de qualidade que um produto de software deve conter como mostra a figura 4.1. Esta estrutura de Modelo de Qualidade pode ser vista como um modelo genérico para avaliar qualquer pacote de software COTS [13], independente da sua área de aplicação podendo ser adaptado aos objetivos de uma avaliação explicitados pelo Requisitante [5].

[4] A NBR ISO/IEC 12119 foi substituída pela NBR ISO/IEC 25051 [12], em 2008.

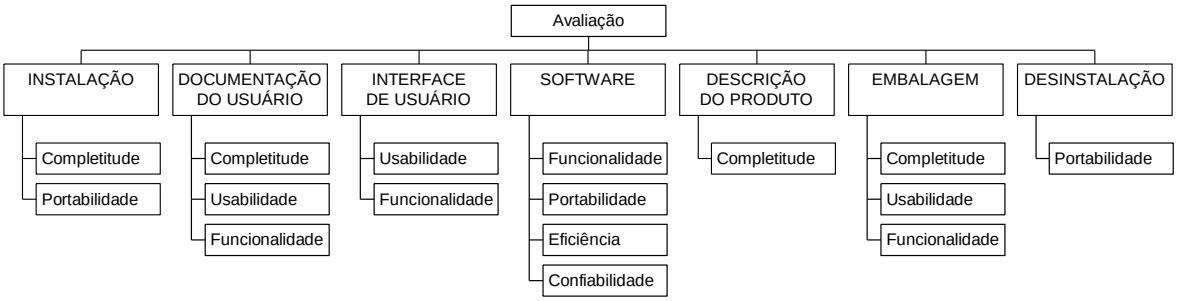


Figura 4.1-Estrutura do modelo de qualidade para Avaliação de Pacotes de Software

Da mesma forma no que se refere ao domínio de aplicação, pode-se utilizar o método para os requisitos de qualidade de produto, sendo que ao tratar das funcionalidades especificas utilizar os requisitos específicos do domínio para avaliação.

Como mostra figura 4.2, onde as características de Funcionalidade são destacadas [14].

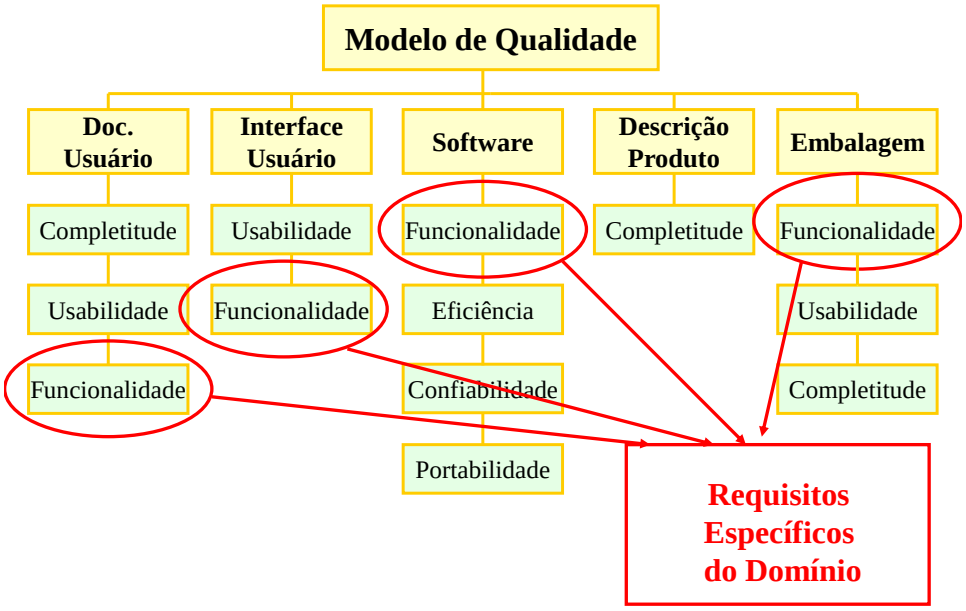


Figura 4.2 – Modelo de Qualidade com requisitos de domínio

Dessa forma, são avaliados produtos de software classificados na literatura como MOTS e FD, possibilitando assim o uso do modelo para avaliação de todos os tipos de software.

5. Benefícios e resultados de uma Avaliação

São vários os benefícios alcançados tanto para o produtor de software como para o usuário final, decorrente de uma avaliação:

- Produtor poderá assegurar a qualidade do produto final;
- Diminuir os custos com a manutenção do software;
- Aumentar a satisfação do usuário, pois estará adquirindo um produto de qualidade;
- Vendedor poderá usar como argumento de venda a qualidade assegurada do produto que está vendendo.

Como resultado da utilização do MEDE-PROS na avaliação dos produtos de software, as empresas receberão um “Relatório de Avaliação” contendo, para cada uma das características de qualidade avaliadas, os aspectos de destaque positivo e aspectos a serem revistos. Com isso, terão condições de implementar melhorias nos seus produtos, gerar software com melhor qualidade e, consequentemente, aumentar a competitividade dos mesmos frente aos concorrentes estrangeiros, de forma a deixá-los compatíveis com as necessidades de respostas aos desafios de um mundo globalizado e cada vez mais competitivo.

O CTI conta atualmente com Laboratórios Credenciados. Estes laboratórios foram capacitados para realizar avaliações de produtos de software com esse método.

Entre as instituições credenciadas estão: INSOFT- Instituto do Software do Ceará; ITS – Instituto de Tecnologia de Software - SP, Unisinos – Universidade do Vale do Rio dos Sinos - RS, CESAR – Centro de Estudos e Sistemas Avançados de Recife, Instituto Gene Blumenau (FURB/GENE), a FACTI – Campinas-SP e o INPI - *Instituto Nacional de la Propiedad Industrial* - Argentina. A FURB/GENE, a FACTI, o ITS e o INPI estão devidamente equipados com plataformas computacionais adequadas aos requisitos dos produtos de software e recursos humanos treinados pelo CTI, aptos a fazer as avaliações dos produtos de software.

6. Experiências de Avaliações

Os objetivos das 480 avaliações de produtos realizadas segundo o Modelo de Qualidade podem ser descritas como: Na experiência das avaliações do Prêmio Assespro foi o de divulgar aos produtores os critérios e ações para evolução da qualidade de seu produto de software, assim como destacar, dentre os produtos inscritos no prêmio, aquele com maior nível de qualidade.

A experiência das avaliações para a Chamada Nacional Softex foi o de selecionar projetos de empresas brasileiras que tinham como meta a concretização de negócios de comercialização no mercado externo de produtos de software, e serviços desenvolvidos no Brasil, e que queiram fazer uso de linhas de créditos especiais oferecidas por organismos de fomento industrial e tecnológico do Governo Brasileiro, tais como CNPq, FINEP, BNDES e outros.

No projeto PNAFM - Programa Nacional de Apoio à Gestão Administrativa e Fiscal dos Municípios Brasileiros, ainda com edital de convocação em aberto, foi o de pré-qualificar empresas e produtos de software para gestão de municípios brasileiros com menos de 50.000 habitantes. Com esse projeto o Ministério da Fazenda, com a ajuda do CTI, busca estabilidade macroeconômica por meio de um equilíbrio fiscal auto-sustentável, fundado em uma política pública transparente e eficiente na gestão da receita e do gasto público municipal. http://www.ucp.fazenda.gov.br/PNAFM.

7. Resultados de 2008 no Ciclo de Projetos do PBQP

- Credenciamento de laboratório MEDE-PROS v 2006 e sua evolução. A Fundação de Apoio à Capacitação em TI – FACTI - Campinas-SP; O INPI - *Instituto Nacional de la Propiedad Industrial*- Argentina;
- Planejamento para credenciar o Laboratório de Engenharia de Software (LABES) da Universidade Federal do Pará;
- Desenvolvimento de Modelo de Qualidade de Produto de Software para domínios específicos;
- Estudo e evolução do método para a nova série de norma NBR ISO 25000 [12];
- Evolução da pesquisa em Qualidade de Software;
- Submissão do livro “Qualidade de Produto de Software”, com foco em requisitos e avaliação, na nova linha de ação do PBQP Software, que compreende a realização de um concurso anual para promover a publicação de um livro na área de Engenharia de Software, com foco na Qualidade e Produtividade.
- Estudo do programa para Certificação de Software.
- A credenciada para avaliar software com o MEDE-PROS - FURB publicou artigo sobre o credenciamento no livro do Platic [9].

Conclusão

Os resultados aqui relatados, do processo de avaliação da conformidade de produtos de software, dão uma clara indicação de sua importância para o produtor, para o comprador, para o desenvolvedor e para o usuário desses produtos. Para o produtor, o uso das normas estabelece uma diferenciação do seu produto em relação a produtos similares; o próprio produtor pode aplicar os conceitos da norma em seu produto, realimentando agilmente o seu processo de desenvolvimento a partir da informação levantada. Para o comprador, a norma estabelece orientações seguras para a aquisição de software, alertando-o sobre o quê deve ser observado no exame de um produto sob consideração. O usuário receberá produtos mais confiáveis e com padronização de nível internacional.

Quanto à elaboração de um Método de Avaliação Especialista de Domínio, ele se caracterizou por ser uma atividade complexa que requer tempo e necessita o envolvimento de profissionais especialistas no domínio da aplicação. Sendo que na tomada de decisão de aplicá-lo ou não, deve ser feita a análise da relação custo benefício.

Quanto à execução de um processo de avaliação da conformidade, observa-se a forte demanda, tempo para preparar os recursos humanos, materiais e artefatos a serem entregues. Além disso, existe a necessidade de recursos financeiros, lembrando que nem sempre é viável executá-lo na sua plenitude. Muitas vezes aplicar parte do processo de avaliação é suficiente para garantir a qualidade esperada a custo reduzido.

Finalizando, pode-se dizer que esse método pode apoiar a certificação de software no Brasil, cuja discussão tem sido pauta nos encontros entre o governo, empresas e academia como o que ocorreu em 31 de maio 2007 em painel Setorial no INMETRO. http://www.inmetro.gov.br/noticias/conteudo/programaSoftware.htm

Perspectivas Futuras

Na reestruturação do CTI, esta será a última apresentação do grupo que deverá participar de outros projetos no contexto de uma nova estrutura hierárquica. O projeto de Avaliação de Produto de Software deverá ter continuidade se a FACTI - Fundação de Apoio à Capacitação em TI abarcar esse desafio.

Referências

[1] ISO/IEC 9126, 1991. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION ISO/IEC 9126 Information Technology – Software product evaluation - Quality characteristics and guidelines for their use;13 p. Geneve ISO Dez / 1991

[2] MARTINEZ, M. R. M., et al **The Software Product Evaluation Data Base – Supporting MEDE-PROS.** In: ISESS - International Software Engineering Standards Symposium – Best Software Practices for the Internet age, 4, 1999, Curitiba. Anais da 4 ISESS. Curitiba:Gráfica, maio 1999.

[3] SCOPE – **Software CertificatiOn Programme in Europe** Final Report – Verilog – March 1993.

[4] TSUKUMO, A . N., et al ISO/IEC 9126 : **An Experiment of Application on Brazilian Software Products**. Proceedings Second IEEE International Software Engineering Standards Symposium, Montréal, Quebec, Canada, 1995; pg. 184-190.

[5] COLOMBO, R. M. T.; Guerra, A . C. **The Evaluation Method for Software Products.** In: ICSSEA 2002 15 th Internacional Conference Software & Systems Engineering and their Applications – Paris, França – Dezembro de 2002.

[6] GUERRA, A. C.; Aguayo, M. T. V.; Colombo, R. T. - **Processo de Avaliação de Produtos de Software**. In: Conferência Ibero-Americana WWW/Internet 2005, International Association for Development of the Information Society. Lisboa, 2005. p. 437-442;

[7] IEEE COMPUTER SOCIETY. IEEE - **Software Engineering Standards Colletion**. IEEE STD 1062 - IEEE Recommended Practice for Software Acquisition. New York, NY. 1998a. 43p.

[8] ISO/IEC - International Organization for Standardization., “ISO/IEC 15504: Information Tecnology Process Assessment, Part 1 to Part 5”, ISO/IEC International Standard, 2005.

[9] CORAL,E.;PEREIRA V.A.;BIZZOTTO C.E.N. **Arranjo produtivo catarinense – Tecnologia da informação e comunicação Série Platic** –, Vol I, Ed. Instituto Euvaldo Lodi - IELsc 2007.319p.

[10] ROCHA, A. R. C.; MALDONADO, J.C.; WEBER, K.C. **Qualidade de Software – Teoria e Prática**. 1. ed. São Paulo: Prentice Hall, 2001. 303 p

[11] ISO/IEC 9126-1. INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. Information technology – Software product quality – Part 1: Quality model. Genebra, 2001. 25 p.

[12] NBR ISO/IEC 25000 ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS Engenharia de software - Requisitos e avaliação da qualidade de produtos de software (SQuaRE) - Guia do SQuaRE Rio de Janeiro ABNT, 2008

[13] COLOMBO, R. M. T. Processo de Avaliação da Qualidade de Pacotes de Software. Campinas, SP, 2004. 169pp. Trabalho Final de Mestrado Profissional. Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

[14] MAITINGUER, S. T. Um método de avaliação especialista para produtos de software, desenvolvido a partir dos requisitos de um edital. Campinas, SP, 2004. 145pp. Trabalho Final de Mestrado Profissional, Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica. 2004.

Aplicação da Teoria das Restrições na Gestão de Desenvolvimento de Software

Lauro César Araujo[1], Américo Ferreira da Silva Neto[2]

Caixa Econômica Federal
SBS Quadra 4 Bloco A Lote 3/4 – Brasília, DF – Brasil

[1] lauro.c.araujo@caixa.gov.br ;[2] americo.f.neto@caixa.gov.br

Abstract. This paper presents the evaluation of one phase of the 2007/2008 macro-project "*Guia de Orientação para Uso da Teoria das Restrições a Projetos de Software*" sent to "*Programa Brasileiro da Qualidade e Produtividade em Software - PBQP Software*". This phase applied the Theory of Constraints on a critical software development project at *Caixa Econômica Federal*.

Resumo. Este artigo expõe a avaliação da etapa realizada em 2007/2008 do macro-projeto "Guia de Orientação para Uso da Teoria das Restrições a Projetos de Software" postado ao Programa Brasileiro da Qualidade e Produtividade em Software - PBQP Software. Esta etapa consistiu no uso da Teoria das Restrições no gerenciamento de um projeto de desenvolvimento de software de missão crítica para a Caixa.

1. Introdução

Um grande desafio no desenvolvimento de software é orquestrar processos com produtividade e retorno financeiro adequado. Esses processos envolvem a complexidade de várias áreas de conhecimento, informações e padrões. A complexidade inerente a atividade de desenvolvimento de software torna necessária a existência de uma gerência baseada em pradronizações e gestão aderentes às boas práticas administrativas, a exemplo da norma NBR ISO/IEC 12207 – Processos do Ciclo de Vida do Software e o PmBok - *Guide to the Project Managemente Body of Knowledge*. Com intuito de orientar as decisões tomadas durante o gerenciamento de projetos de desenvolvimento/manutenção de software, sugerimos acrescentar a Teoria das Restrições (TOC) como uma ferramenta para o gerenciamento estratégico.

A Teoria das Restrições foi desenvolvida no fim da década de 70 pelo físico israelense Eliyahu M. Goldratt. Produto do aprimoramento de uma fórmula matemática de produção otimizada, a TOC – Theory of Constraints – foi publicada num romance chamado A Meta (*The Goal*), por Goldratt e Jeff Cox, em 1984. O alicerce da Teoria das Restrições está no foco da maximização do resultado da organização. A TOC pressupõe que todo sistema tem pelo menos um gargalo/restrição (ponto frágil), caso contrário, se não houvesse algum tipo de limitador, a produção seria infinita. A filosofia é agir sobre o ponto mais frágil do sistema. Conforme Goldratt (2002), "a produtividade é o ato de fazer uma empresa ficar mais próxima de sua meta. Todas as ações que fazem com que a empresa fique mais próxima de sua meta são produtivas."

2. Objetivos e Justificativa

O departamento de desenvolvimento de software da Caixa, em 2007, registrou cerca de 1.700 solicitações mensais de clientes internos. Estas solicitações eram referentes à melhoria, correção de falhas e erros dos softwares existentes, construção de novos projetos de solução tecnológica, serviços de informação, entre outras. Devido à grande demanda por tecnologia inerente às atividades de uma instituição financeira com a diversidade de atividades da Caixa, é necessário inovar para encontrar formas de gerenciamento adequadas às expectativas de clientes internos e externos à instituição.

Este artigo expõe a avaliação da etapa realizada em 2007/2008 do macro-projeto "Guia de Orientação para Uso da Teoria das Restrições a Projetos de Software". O projeto-macro tem como objetivo gerenciar projetos de software utilizando os princípios da Teoria das Restrições para, a partir das experiências obtidas, construir um Guia de Uso da TOC para a Caixa Econômica Federal e, possivelmente, um guia genérico a ser disponibilizado a toda comunidade de software brasileira. Os resultados descritos neste artigo são referentes à aplicação da Teoria das Restrições na gestão de desenvolvimento de um módulo de software de missão crítica específico na instituição. O desenvolvimento do projeto durou cerca de 8 meses. O módulo em questão é o responsável pela automação do produto de loterias denominado TIMEMANIA, considerado um software de missão crítica na empresa.

3. Metodologia de Execução

Primeiramente, selecionou-se um projeto crítico para um dos departamentos de desenvolvimento de software da instituição. O projeto envolvia prazo fixo determinado em lei, alta complexidade de requisitos funcionais e não funcionais. A data máxima em que a aplicação deveria estar disponível era factível, mas não confortável. A equipe continha especialistas em testes e testadores de software, arquitetos, analistas e programadores Java e COBOL, analistas de requisitos, líderes de projeto e gerente de configuração. Ao todo, a equipe totalizava 104 integrantes. Externo à equipe, mas com fronteira direta com ela, haviam: a equipe de produção, responsável por manter o software em ambiente produtivo; a equipe de analistas do negócio, responsável pelo detalhamento das funcionalidades necessárias; e equipe de negócios e desenvolvimento de outros sistemas, que deveriam fazer interface com o módulo a ser construído.

Neste cenário, o coordenador do projeto embasou suas decisões às premissas da TOC.

3.1 Premissas da TOC

Definir a meta

Goldratt (2002) afirma que a meta de qualquer empresa com fins lucrativos deve ser ganhar dinheiro hoje e sempre. A lucratividade não conflita com nenhuma missão, uma vez que é ela que garante a sustentabilidade da organização. Os produtos e serviços prestados pela empresa devem estar alinhados com a meta, pois são eles que possibilitam e sustentam o alcance dela.

Definir medidas

Além da definição da meta, Goldratt (2002) estabelece medidas operacionais da TOC, das quais inferimos o conteúdo, como segue:

Ganho ou *Throughput* - "índice pelo qual o sistema gera dinheiro através das vendas". Segundo a TOC, produção não é ganho enquanto o produto não é vendido.

Inventário - "todo o dinheiro que o sistema investiu na compra de coisas que ele pretende vender", como o estoque de matérias-primas, produtos em processo, produtos acabados, ativos como máquinas e construções, registrados pelo valor de custo.

Despesas Operacionais - "todo o dinheiro que o sistema gasta a fim de transformar o inventário em ganho". São despesas como energia elétrica ou mão-de-obra e valores de bens que faziam parte do inventário e foram utilizados ou depreciados.

3.2 Premissas da TOC para o projeto selecionado

Baseado nas premissas da TOC, o primeiro passou foi definir a meta e as unidades de medida para o projeto selecionado.

Especialmente no que diz respeito à meta a ser atingida, o escopo dela foi reduzido à fronteira do projeto específico que trata este artigo. O projeto, por sua vez, está alinhado à estratégia da instituição. A CAIXA, como empresa pública de direito privado, possui metas próprias, que não se reduzem apenas ao ganho financeiro.

Definições da TOC para o projeto selecionado:

a) Definir a meta a ser atingida:

Meta do projeto: Cumprir a lei - entregar o produto

b) Definir unidades de medida:

Medida do ganho: testes de software satisfatórios

Medida do inventário: revisões de software, solicitadas por testes insatisfatórios

Medida das despesas operacionais: custo diário dos recursos do projeto

3.3 Passos adotados pelo coordenador do projeto

O coordenador do projeto utilizou os “5 passos da TOC” (Goldratt, 2002) para identificar restrições/gargalos no processo e tomar ações, conforme segue:

Identificação da Restrição 1 (R1)

R1 - Passo 1 - Identificar as restrições do sistema:

Problemas com a identificação do escopo, que estava consumindo tempo muito superior à prevista no projeto. O prazo final não poderia ser alterado, por limitação legal.

R1 - Passo 2 - Decidir como explorar as restrições:

Identificou-se que existiam problemas no escopo do projeto (e na elucidação de requisitos) referentes à integração com outros sistemas, especificamente quanto ao sistema de contabilidade.

R1 - Passo 3 - Subordinar qualquer coisa à decisão

Nenhuma outra ação foi tomada antes da solução desta restrição.

R1 - Passo 4 - Elevar a restrição do sistema

A ação tomada foi obter comprometimento formal de gerentes e técnicos de todas as áreas/departamentos envolvidos e que não sabiam que estavam no caminho crítico do projeto. Para resolver a restrição do sistema, os gerentes e técnicos das áreas envolvidas deveriam agir prioritariamente na solução das pendências identificadas.

R1 - Passo 5 - Sem inércia

O último passo da TOC diz que não é permitido parar. Por isso, a avaliação recomeçou e identificou nova restrição.

Identificação da Restrição 2 (R2)

R2 - Passo 1 - Identificar as restrições do sistema:

Depois que o escopo foi definido, que os requisitos críticos foram especificados e a codificação foi iniciada, as primeiras compilações do software foram testadas. O ciclo completo de testes da aplicação – aproximadamente 24.000 mil itens de verificação distribuídos em 103 roteiros de teste referentes a 79 casos de uso – demorava cerca de 8 dias. Conforme os trabalhos de implementação eram concluídos, parte da equipe de implementação ficava ociosa aguardando testes serem realizados pela equipe de testes. A equipe de testes registrava revisões de software (registro de falhas a serem corrigidas) apenas ao final dos testes. Como no exemplo da cadeia produtiva de Goldratt (2002), parte da equipe tinha muito trabalho e a outra parte ficava parada até chegar trabalho.

Decidir como explorar as restrições

Agir no tamanho dos ciclos de testes e no momento de abertura das revisões.

Subordinar qualquer coisa à decisão

Embora fosse possível identificar outras restrições no processo, esta era a mais importante e de maior custo ao projeto. Nenhuma energia deveria ser gasta em outro ponto não necessário.

Elevar as restrições do sistema

Como solução, aumentou-se a freqüência de testes, diminuindo o escopo de cada ciclo às alterações realizadas. Isso foi possível com o auxílio da gerência de configuração e dos arquitetos de software, que informavam à equipe de Teste os possíveis impactos arquiteturais de cada alteração. Com isso, os testes tinham duração menor e revisões eram identificadas mais cedo. Também foram criados testes unitários na aplicação Java. Eles eram executados automática e diariamente para auxiliar na identificação de falhas. Foram utilizados testes completos somente após resolvidos os problemas pontuais.

Sem inércia

Com essas ações, esta restrição foi resolvida e o produto liberado no prazo esperado.

4. Resultados Obtidos

Para o projeto selecionado, além do produto liberado com os requisitos e prazo determinado legalmente, identificaram-se ações futuras a realizar em novos projetos:

a) alinhar todos, sem exceção, a um mesmo objetivo;

b) necessidade de identificar componentes Java para melhorar a definição de estratégia e escopo de testes;

c) antecipar identificação de revisões de software;

d) criar mais testes unitários;

e) descentralizar conhecimento de softwares COBOL;

f) aumentar a freqüência de testes;

Um artigo acadêmico com o título “Teoria das Restrições na gestão de desenvolvimento de software - Um estudo de caso” também foi produzido, mas ainda não publicado.

5. Aplicabilidade dos Resultados

Os resultados obtidos até agora podem ser estudados e replicados em projetos semelhantes na Caixa e outras instituições.

A Caixa possui um programa de identificação de Melhores Práticas em Tecnologia da Informação. Este programa seleciona, bienalmente, projetos locais de todo o país para aplicação corporativa. A experiência obtida com a aplicação da TOC descrita neste artigo e experiências anteriores levou a inscrição do projeto neste fórum para disseminação e aplicação em larga escala na instituição.

6. Características Inovadoras

O uso da TOC na gestão de atividades é algo novo para a CAIXA e para as empresas em geral no Brasil. Utilizar a TOC é gerir processos sempre com a visão do todo, do objetivo final, evitando desperdícios e ações que tomariam tempo e dinheiro sem ajudar no cumprimento das metas.

7. Conclusão e Perspectivas Futuras

A Teoria das Restrições pode ser incorporada ao conjunto de conhecimentos necessários para a administração de empresas. Identificar o principal gargalo evita ações que não buscam atingir a meta da empresa, abre caminhos para ações que minimizam os inventários e maximiza o retorno financeiro para a instituição, além de ajudar na condução de inovações e criação de novos produtos.

O passo 5 da TOC (“Se nos passos anteriores uma restrição for quebrada, volte ao passo 1, mas não deixe que a inércia se torne uma restrição do sistema”) indica que o processo de avaliação e melhoria é contínuo e não uma ação isolada. Sendo assim, é importante que a visão da identificação dos gargalos seja constante. Além do mais, gargalos mudam a todo instante e são diferentes dentro de cada projeto.

Uma boa reflexão em torno da aplicação de TOC à engenharia de software pode ser expressa nas palavras de Milea e Lyttle (1986): "Uma das melhores lições que uma criança aprende ao jogar videogames é que ficar parado é a melhor maneira de acabar eliminado".

Referências Bibliográficas

GOLDRATT, Eliyahu M.; Cox, Jeff. **A Meta**. 2º edição. Tradução: Thomas Corbett Neto. São Paulo: Nobel, 2002

MILEA, Jinx; LYTTLE, Pauline. ***Why Jenny Can't Lead***: *Understanding The Male Dominant System*. Saguache CO; 1986.

Modelo de Processo de Aquisição de Software e Serviços Correlatos da FUNCEF

Nirian Martins S. Santos1, Luciana Roldan de Almeida1, Lania V. A. Lacorte1

1Fundação dos Economiários Federais(FUNCEF) – Diretoria de Administração e Tecnologia da Informação – Brasília – DF – Brasil

{nirian,lucianaroldan,lania}@funcef.com.br

Abstract. *Actions to improve the processes of software in organizations commonly have focused on aspects of the software development process. However, with the emergence of the need to purchase products and services of Information Technology (IT), is relevant to establish a procedure Acquisition of Software and Related Services (S&SC), which is the reference of the best practices needed to guide and implement the purchase products or software services, help in mitigating risks to the business. Thus, the process involves several activities and tasks, which through various artifacts support the procurement process. The acquisition of S&SC, in general, begins with the identification of need and ends with the acceptance by the customer. The purpose of this paper is to present the context of the Model Procedure for Acquisition of S&SC FUNCEF, and their results.*

Resumo. *As ações de melhoria de processos de software nas organizações comumente têm focado nos aspectos do processo de desenvolvimento de software. Entretanto, com o surgimento da necessidade de aquisição de produtos e serviços de Tecnologia da Informação (TI), é pertinente estabelecer um Processo de Aquisição de Software e Serviços Correlatos (S&SC), que é a referência das melhores práticas indispensáveis para orientar e executar a aquisição de produtos ou serviços de software, auxiliando na mitigação de riscos ao negócio. Desta forma, o processo envolve várias atividades e tarefas, que por meio de diversos artefatos apóiam o processo de aquisição. O processo de aquisição de S&SC, de maneira geral, inicia com a identificação da necessidade e encerra com a aceitação pelo cliente. O objetivo deste artigo é apresentar a contextualização do Modelo de Processo de Aquisição de S&SC da FUNCEF, e seus resultados obtidos.*

1. Introdução

A partir da demanda do mercado para que as organizações avaliem questões culturais, processos de negócio, aumento da produtividade com padrão de qualidade competitivo, a FUNCEF, por meio da Gerência de Tecnologia da Informação viu-se diante da necessidade de estabelecer um Modelo de Processo de Aquisição de S&SC, para apoiar o planejamento da aquisição, a classificação técnica de fornecedores, a gestão de contratos e a aceitação dos produtos e serviços de TI.

Diante deste contexto, tornou-se necessário desenvolvimento da pesquisa na literatura, focado na aquisição de produtos e/ou serviços, tendo como referência os modelos MPS.BR e CMMI, os guias PMBOK e COBIT, a norma ISO/IEC 12207 e a legislação brasileira. A partir desses referenciais, foi elaborada a proposta do processo customizado de aquisição de produtos e serviços de TI, realizada no âmbito acadêmico por meio de trabalho de conclusão de curso, utilizando a organização em questão como referência para o Estudo de Caso. A partir deste ponto, os próximos passos remetiam a execução de trabalhos em projeto piloto com a finalidade de avaliar e auxiliar os procedimentos necessários à formalização e institucionalização do modelo de processo de aquisição de S&SC da FUNCEF.

2. Objetivos e Justificativa

O objetivo macro do projeto remete a mitigação riscos ao negócio a partir de modelo próprio de aquisição de S&SC, fundamentado às melhores práticas e referências de mercado. Assim, entre os objetivos específicos deste projeto estão:

- Elaboração de artefatos suficientes para entendimento do processo de aquisição de S&SC;
- Customização do Processo de Aquisição de S&SC;
- Formalização e institucionalização do Processo de Aquisição de S&SC.

O projeto justifica-se pela necessidade de formalização, qualidade, transparência, dinamismo do negócio e peculiaridades da legislação e gestão das Entidades Fechadas de Previdência Complementar (EFPC).

3. Metodologia de Execução

Para customizar o processo de aquisição S&SC, a fim de obter o Modelo de Processo de Aquisição de S&SC da FUNCEF, foram executadas 4 etapas conforme prevê a Figura 1.

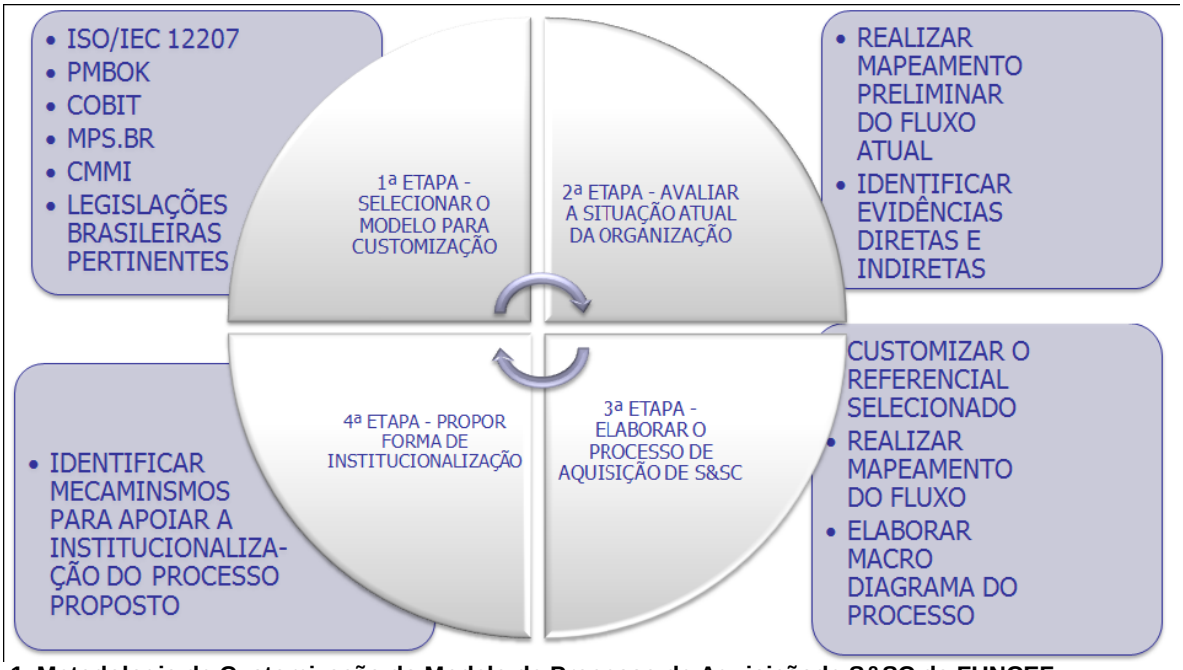


Figura 1. Metodologia de Customização do Modelo de Processo de Aquisiçãode S&SC da FUNCEF

As etapas da Figura 1 foram realizadas no âmbito acadêmico, com vistas à proposição do Modelo de Processo de Aquisição de S&SC da FUNCEF.

3.1 Seleção do Modelo para Customização - 1ª ETAPA

A seleção deu-se através da comparação entre as atividades e tarefas propostas pelos referenciais dos modelos MPS.BR e CMMI, os guias PMBOK e COBIT, a norma ISO/IEC 12207 e a legislação brasileira. Também foram avaliados os normativos internos da organização. Outros fatores tidos como pontos fortes, como por exemplo, a proposta de *templates* e a possibilidade de implementação e avaliação futura da organização, motivaram a seleção do modelo MPS.BR para customização, especialmente por que as atividades e tarefas propostas pelo mesmo estão mais próximas das atividades executadas pela organização. Esse modelo ainda apresenta de modo claro e objetivo como a adquirente deve proceder para atingir os resultados esperados no processo de aquisição, inclusive apresentam nos anexos do guia alguns modelos que poderão servir à elaboração de *checklists* e *templates*.

3.2 Avaliação da Situação Atual da Organização - 2ª ETAPA

Para realizar a análise da situação atual adotou-se parcialmente o Modelo de Avaliação do MPS.BR (MA-MPS.BR) com o objetivo de identificar as evidências diretas e indiretas e as oportunidades de melhoria com relação aos resultados esperados do processo de aquisição de *Software* do MPS.BR.

Portanto, a avaliação deu-se por meio da análise do ambiente organizacional, incluindo Circulares Normativas e procedimentos observados na prática da execução do processo vigente, o que tornou possível a identificação das evidências diretas e indiretas, em associação com cada Resultado Esperado do Processo de Aquisição do MPS.BR.

O Guia de Implementação - Parte 2 do MPS.BR apresenta em detalhes os objetivos de cada resultado esperado do processo de aquisição. Estas definições em conjunto com o Guia de Aquisição do MPS.BR auxiliaram na identificação dos produtos requeridos de acordo com cada tarefa prevista no modelo selecionado.

3.3 Elaboração do Modelo de Processo de Aquisição de S&SC - 3ª ETAPA

A partir da análise realizada na 2ª Etapa, observou-se que como a organização não possuía formalizado um modelo de processo específico para aquisição de produtos e serviços de TI, propôs-se melhorias ao processo, a partir do modelo selecionado e do modelo praticado pela organização, apontadas com base em cada um dos 9 (nove) resultados esperados do processo de aquisição do MPS.BR. Assim, elaborou-se a customização do processo de aquisição com foco no MPS.BR.

A Figura 2 apresenta a arquitetura final do processo proposto para a organização. Neste processo são descritas atividades e tarefas de origem do modelo MPS.BR, são incluídas tarefas próprias da organização, incrementados com os pontos fortes dos referenciais estudados e foram observadas as circulares normativas internas da organização, com a finalidade de evitar discrepâncias e sobreposições do processo proposto com os atos normativos existentes.

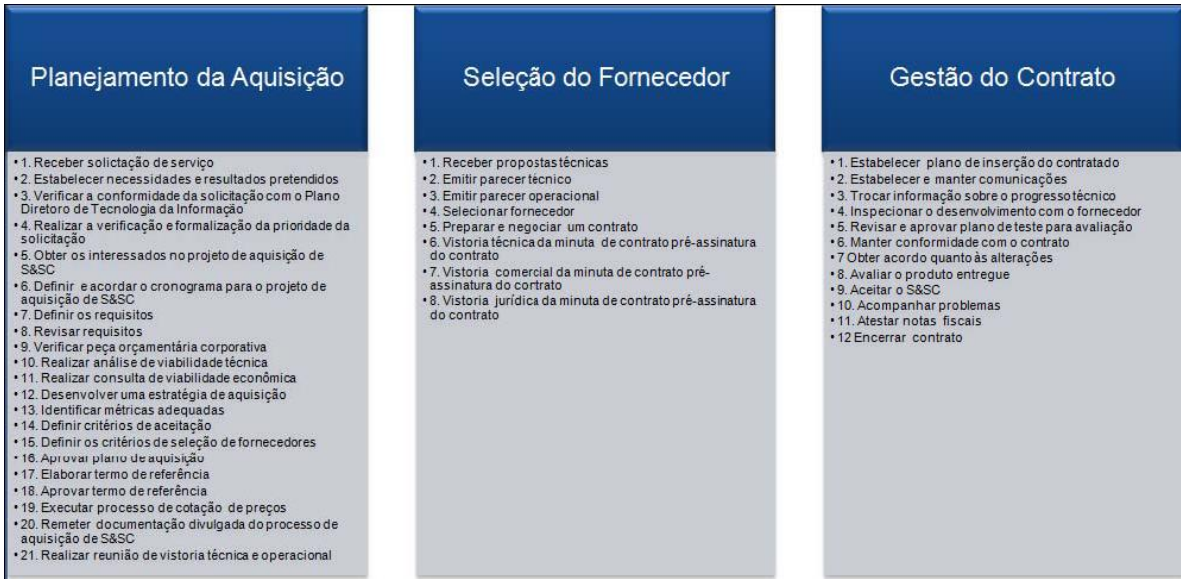


Figura 2. Proposta de Modelo de Processo de Aquisição de S&SC da FUNCEF

3.4 Proposição de Forma de Institucionalização – 4ª ETAPA

A partir do processo proposto na 3ª Etapa, as seguintes ações para institucionalização do Modelo seriam necessárias:

- Divulgação no ambiente corporativo;
- Treinamento dos envolvidos no processo de aquisição de S&SC;
- Elaboração de *Templates* e *Checklists* para apoiar a execução do processo de aquisição de S&SC;
- Aplicação do Modelo de processo proposto por meio de projeto piloto;
- Avaliação e melhorias necessárias para Normatização;
- Normatização do Modelo de processo de aquisição de S&SC.

4. Resultados Obtidos

A partir da execução do modelo de processo de aquisição de S&SC proposto, em modalidade de projeto piloto em contratações de produtos e serviços de TI, por meio da aplicação de *templates* e *checklists* dos artefatos elaborados na fase de planejamento, seleção e gestão dos contratos, foram identificadas melhorias para o modelo de processo proposto. Essas melhorias foram demandadas especialmente em função do dinamismo necessário, da tempestividade requerida para o processo e também em razão de novos normativos internos publicados ao longo da realização dos projetos piloto, o que também favoreceu a revalidação do modelo proposto. Assim, entre os resultados obtidos destacam-se:

- Proposição de modelo de processo de aquisição de S&SC customizado;
- Conclusão de elaboração de artefato suficiente para entendimento do processo de aquisição de S&SC, quais sejam:
    - Plano de Aquisição – Artefato direcionador do processo de aquisição;
    - Modelo de Termo de Referência – Artefato remetido para cotação de preços.
- Elaboração de templates e checklists para apoiar a execução do processo de aquisição de S&SC, quais sejam:
    - Termo de Referência; Modelo de avaliação de propostas técnicas; Minuta Contratual; Monitoramento contratual.
- Aplicação do modelo de processo proposto por meio de projeto piloto;
- Avaliação e melhorias necessárias no modelo de processo proposto;
- Publicação de artigo no V Simpósio de Excelência em Gestão e Tecnologia, ministrado pela Associação Educacional Dom Bosco, Resende - RJ;
- Recursos humanos capacitados a partir dos seguintes treinamentos:
    - Curso de Melhoria de Processo de Aquisição de Software do MPS.BR;
    - Certificação MPS.BR no Processo de Aquisição;
    - Participação no I Congresso Nacional de Contratação de TI.

5. Aplicabilidade dos Resultados

No âmbito do ciclo de 2008, a relevância do projeto remete a customização de Modelo de Processo de Aquisição de S&SC, aderente às melhores práticas de mercado, e estruturado para as necessidades da organização.

O projeto destina-se a área de Tecnologia da Informação, abrangendo também as áreas gestoras da FUNCEF que requerem produtos e serviços de TI. Entretanto, sabe-se que novas melhorias do processo serão demandas objetivando o alinhamento entre novos normativos internos da organização e as tarefas e responsabilidades descritas no modelo de processo de aquisição proposto.

O principal impacto deste projeto refere-se à definição de atividade interna na Gerência de Tecnologia da Informação da organização, com vistas a apoiar a definição, execução e institucionalização do processo de contratação de produtos e serviços de TI, assim como o monitoramento dos contratos firmados a partir da apuração de indicadores.

6. Características Inovadoras

O projeto inova com modelo de processo customizado denominado “Modelo de Processo de Aquisição de S&SC”, incluindo *templates* e *checklists* para apoiar a execução do processo de aquisição de produtos e serviços de TI, e fazendo uso das melhores práticas de mercado.

7. Conclusão e Perspectivas Futuras

Ao propor um processo de contratação de produtos e serviços de TI, realizou-se uma avaliação da situação atual da organização, de modo a identificar as atividades e evidências que possibilitassem a constatação deste processo dentro da organização.

Nesta análise identificou-se que a organização executa tarefas correlatas ao tema, que não possuem procedimentos formalizados em sua totalidade.

Observou-se ainda que nos normativos internos da organização, algumas tarefas são mencionadas como necessárias para a promoção de um ambiente de governança de TI exeqüível, mas não estão direcionadas especificamente ao processo de contratação de produtos e serviços de TI.

No desenvolvimento do processo proposto optou-se por manter tarefas executadas pela organização e acrescentar tarefas que são pertinentes ao processo de aquisição de S&SC, customizando o modelo MPS.BR, tornando-o mais completo e aderente à organização. As atividades do processo definidas foram: Planejamento da Aquisição, Seleção do Fornecedor e a Gestão do Contrato.

A atividade de “Aceitação pelo cliente” descrita no MPS.BR não foi adotada conforme proposta no Guia de aquisição do MPS.BR, porque na prática, a tarefa de “Definição de critérios de aceitação” é realizada durante o planejamento da aquisição e as tarefas “Avaliar o produto entregue, Manter conformidade com o contrato e Aceitar o S&SC” são realizadas durante a atividade de gestão do contrato.

Por ter o foco na área de Tecnologia da Informação da organização, o processo de aquisição foi elaborado com vistas ao atendimento dessa área de modo geral, ou seja, incluindo tarefas que observassem os serviços de desenvolvimento de sistemas, suporte a sistemas e infra-estrutura. Mas, de uma maneira geral o processo pode ser utilizado para subsidiar a aquisição de qualquer produto e/ou serviço de Tecnologia da Informação.

Portanto, pode-se concluir que a maioria dos objetivos específicos deste projeto foi atingida, a exceção da institucionalização do modelo, por meio das atividades “Divulgação no ambiente corporativo” e “Normatização do Modelo de processo de aquisição de S&SC”, em razão da necessidade de revisão do modelo, ora em andamento, a partir da execução do projeto piloto.

Além da institucionalização do modelo, como proposta para trabalhos futuros, tem-se que realizar a melhoria contínua do processo, a capacitação da equipe envolvida, o desenvolvimento de ferramenta que apóie a implantação e uso do processo, e por fim estabelecer indicadores para mensuração do processo.

8. Referências Bibliográficas

BEAL,Adriana. O que é COBIT?. Disponível em: <http://2beal.org/ti/artigos/index.php?entry=entry080121-015652>. Acesso em: 02 Abr. 2008.

COBIT. IT Governance Institute, COBIT 4.1- Framework, Control Objectives, Management Guidelines, Maturity Models. Disponível em: <www.itgi.org>. Acesso em: 04 Abr. 2008. p. 26.

CRUZ, Cláudio Silva da. Projeto Contratar TI (Parâmetros para Contratação de Serviços de TI). Disponível em: <http://www.ti.controle.gov.br/sitesinteresse/index.htm> Acesso em: 21 Fev. 2008.

DOU. Diário Oficial da União. INSTRUÇÃO NORMATIVA Nº 4, DE 19 DE MAIO DE 2008. Disponível em: <http://diario.in.gov.br/imprensa/pesquisa/pesquisaresultado.jsp?texto_sem=&submit=P esquisar&d-7825134-p=5&dataPublicacaoInicial=20%2F05&texto_expressao=&texto_todas=&dataPublicac

aoFinal=20%2F05&ano=2008&texto_qualquer=Instru%E7%E3o+Normativa+n%BA+ 4>. Acesso em: 26 Mai. 2008.

ISO/IEC 12207. Software & Systems Engineering Standards Committee of the IEEE Computer Society. Systems and software engineering - Software life cycle processes, 2008. p. 18.

MPS.BR. SOFTEX - MPS.BR - Melhoria de Processo de Software Brasileiro, Guia de Aquisição, versão 1.2. Disponível em: <http://www.softex.br/portal/mpsbr/_guias/ MPS.BR_Guia_Geral_V1.1.pdf>. Acesso em: Dez. 2007.

PAULK M, et al. Software Quality and the Capability Maturity Model. Communications of the ACM, Vol. 40, n. 6, p. 30/45, Jun. 1997.

PMI - Project Management Institute. Um Guia do Conjunto de Conhecimentos do Gerenciamento de Projetos - PMBOK Guide – 3 ed. Four Campus Boulevard, Newton Square, 2004.

SEI. SOFTWARE ENGINEERING INSTITUTE. CMMI® for Acquisition. Improving processes for acquiring better products and services, Version 1.2, Nov. 2007. Technical Report CMU/SEI-2007-TR-017 e ESC-TR-2007-017.

FROISPI - Framework Return on Investment of Software Process Improvement

Paulino W. P. Viana1,2, Alexandre M. L. de Vasconcelos1

1Centro de Informática – Universidade Federal de Pernambuco (UFPE)
CEP: 50.732-970 – Recife – PE – Brazil

2Fundação Centro de Análise, Pesquisa e Inovação Tecnológica
Instituto de Ensino Superior FUCAPI – CESF
CEP: 69.075-351 – Manaus – AM - Brazil

{pwpv, amlv}@cin.ufpe.br, wagner.palheta@fucapi.br

Abstract. *The Brazilian companies of Software search to conquer each more time the national and international market, that are more competitive. The viable strategy is to invest in the increase of the quality and productivity that are the weapons of this battle. The focus of this work is to investigate significant factors to measure the Return on Investment (ROI) of Software Process Improvement (SPI). With the objective to consider a Framework consisting of phases based in the concepts ROI Methodology, and using indicating used by David Rico in ROI of SPI and an election of metrics used for MPS.*

Resumo. *As empresas de Software Brasileiras buscam conquistar cada vez mais o mercado nacional e internacional, os quais estão cada vez mais competitivos. A estratégia viável é investir no aumento da qualidade e produtividade. O foco desse trabalho é investigar fatores significantes para mensurar o Return on Investment (ROI) em Melhoria de Processo de Software (MPS). Com o objetivo de propor um Framework constituído por fases baseado nos conceitos da ROI Methodology, utilizando indicadores utilizados por David Rico em ROI of SPI e uma seleção de medições utilizadas para MPS.*

1. Introdução

A Melhoria de Processo do Software (MPS) tem recebido muita atenção recentemente. Diversos esforços podem ser encontrados na academia e na indústria. Os modelos como *CMMI* [CMU/SEI 2007] e o *SPICE* da ISO [ISO 1998] são iniciativas conhecidas de MPS. Geralmente, implementar normas ou modelos de qualidade é considerado uma atividade bastante demorada e requer um investimento muito alto. Conseqüentemente, as pequenas e médias empresas (*PMEs*) enfrentam dificuldades para controlar e melhorar seus processos de software devido a seus recursos limitados. Em resposta às necessidades e às limitações específicas das PMEs para implementar programas de MPS, foi criado em 2003 o MPS.BR – Melhoria de Processo de Software Brasileiro [Softex 2007] sob a coordenação da SOFTEX – Associação para Promoção da Excelência do Software Brasileiro. Mas como controlar os custos de uma implementação de MPS, quais são os aspectos relevantes que possam demonstrar um Retorno de Investimento (*Return on Investment-ROI*) para tranqüilizar os empresários.

Este artigo apresenta alguns resultados da pesquisa experimental realizada em três empresas sobre os aspectos relevantes de retorno de investimento em Melhoria de Processo de Software.

2. Objetivos e Justificativa

As empresas brasileiras do software querem conquistar ainda mais o mercado nacional e internacional, mercado que está cada vez mais competitivo com a ascensão das empresas da Índia e da China. A estratégia viável é investir no aumento da qualidade e da produtividade. A eficiência de processos do software é uma exigência deste mercado competitivo e para não contradizer a máxima que um bom processo garante um bom produto, uma boa prática é monitorar todo o ciclo da vida do processo [Fuggetta 2000].

Para uma boa monitoração e controle, é necessário ter objetivos simples e alcançáveis, com marcos de projetos bem definidos e uma seleção de indicadores de qualidade, obtidos através de medições. O grande desafio para empresas de software brasileiras é quais medições devem ser utilizadas, caso contrário, poderão demonstrar resultados indesejáveis, dessa forma é importante não apenas saber claramente o que se medir e qual a periodicidade desta medição e a forma de coleta dos dados, e o principal, o que fazer com os resultados dessas medições. Realizar muitas medições pode elevar os custos de um processo e ainda assim não atingir seus objetivos, isso acontece por falta de objetividade nas medições. Sabe-se que, para fazer melhorias em um Processo de Software, devemos conhecer como esse Processo se comporta. Uma vez identificado o comportamento de tal Processo de Software, e o mesmo sendo insatisfatório, pode-se propor uma Melhoria de Processo de Software (MPS)

Sabe-se que um gerenciamento adequado de projetos de Melhoria de Processo de Software (MPS) é vital para a concretização das estratégias organizacionais. Entretanto como saber se um projeto de MPS foi bem sucedido e de que forma impactou ou impacta no negócio da organização? Por outro lado, as organizações sempre vão esperar bons resultados de seus investimentos, e como garantir esse Retorno sobre o

Investimento (*Return on Investment – ROI*)? O sucesso de uma MPS não pode ser medido apenas por estar dentro do orçamento, dentro do prazo e pelos produtos entregues de acordo com as especificações do modelo de qualidade utilizado. Deve-se considerar o impacto do sucesso no negócio da organização.

As organizações precisam de um processo estruturado e sistematizado que permita e viabilize a avaliação do processo de gerenciamento de projetos de MPS e seus respectivos resultados que possam impactar no negócio, funcionando não apenas como uma ferramenta de medição, mas principalmente como ferramenta de melhoria de processo contínua.

Os métodos tradicionais de avaliação econômica de retorno de investimento não são tão eficientes, visto que não contemplam uma das principais características da área de Tecnologia de Informação (TI) que são os benefícios intangíveis. Tais benefícios intangíveis e não-monetários têm extremo valor para as organizações.

3. Metodologia de Execução

O *FROISPI* foi idealizado seguindo o conceito de modelos de melhoria de processo e foram definidas cinco fases que se inter-relacionam, como visto na Figura 1 [Viana, 2008]. As fases são: Identificação do problema; Diagnóstico detalhado; Estimativa de ROI; Implementação do Projeto piloto e Encerramento. Para cada fase, baseados no paradigma *GQM – Goal-Question-Metric* [Solingen e Berghout, 1999] foram definidos fatores críticos de sucesso através de indicadores de medição para monitorar o andamento do framework. Também foram definidos os resultados esperados para cada fase. Os fatores críticos de sucesso foram definidos exclusivamente para o acompanhamento do framework, não houve a intenção de definir indicadores para o acompanhamento do gerenciamento de projeto.

As quatro primeiras fases seguem o conceito clássico do PDCA, que para cada solução sugerida de melhoria, analisa seus resultados e se os mesmos forem considerados plenamente satisfatórios, seguirá para a fase de Encerramento, caso contrário o processo cíclico continua até a necessidade de melhoria ser satisfeita. Na fase de Encerramento serão apresentados à alta direção os resultados alcançados com o *FROISPI*.

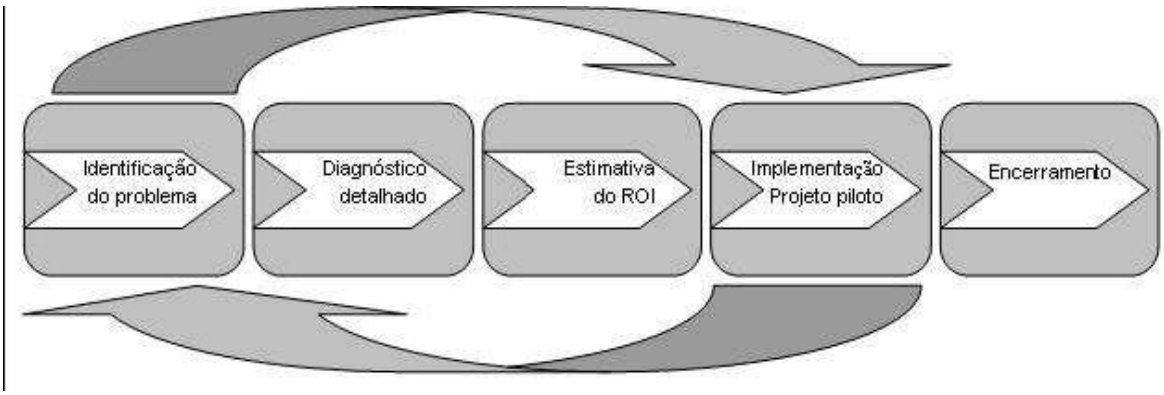


Figura 1. Framework Return on Investment of Software Process Improvement.

3.1 Identificando o problema

A primeira fase tem como objetivos: identificar os Objetivos Estratégicos da organização se possui algum processo de software, quais são os principais problemas existentes na organização, quais são as expectativas em relação à Melhoria de Processo e se possui recurso financeiros para o investimento. Esses dados são registrados no questionário da primeira fase.

3.2 Diagnóstico detalhado

Na segunda fase é realizado um diagnóstico mais detalhado. Tendo os objetivos estratégicos como direcionamento, é realizado um detalhamento do processo identificado observando seus pontos fortes e pontos fracos, verifica-se os resultados alcançados, número de projetos que obtiveram sucesso ou não. Verifica-se a existência de qualificação dos colaboradores da organização e se existe Gerencia de Projetos. Esses dados são registrados no questionário da segunda fase.

Nessa mesma fase será apresentada uma solução em Melhoria de Processo baseado no diagnóstico que pode ser: implantação de um processo; capacitação de recursos humanos caso tenha sido identificado falta de habilidade em determinada tecnologia do processo; aquisição de ferramentas para automatizar partes do processo; e os benefícios que podem trazer para a organização.

3.3 Estimativa do ROI

A terceira fase é muito importante, pois verifica a maturidade da organização no aspecto do processo de software. Identifica se a organização tem alguma avaliação ou certificação de processo ou produto. Qual é o nível de experiência do *SQA-Software Quality Assurance* e do *SEPG-Software Engineering Process Group*. Qual é o nível de complexidade dos processos envolvidos para o MPS. Verifica a existência de dados históricos da Qualidade. Define metas como, por exemplo, aumento da produtividade; redução de retrabalho; otimização do Processo e ganho em qualidade do produto. Os resultados dessa fase servirão para o planejamento do projeto piloto.

3.4 Implementando um Projeto Piloto

A melhoria de processo de software precisa ser gerenciada, portanto é tratada como um projeto onde tem inicio e fim, e com um objetivo. Elabora-se o planejamento do projeto completo analisando escopo e seus limites, os papeis responsáveis, as atividades, o cronograma, os riscos identificados com suas mitigações e com tudo isso conseguimos ter o custo do projeto de melhoria.

Executa o projeto piloto com um caso real, onde pode ser um projeto de software ou iterações de projeto muito grande. O importante é executar todo o projeto para registrar o comportando da melhoria e servir para analise. Com isso é gerado os relatórios de analise e caso seja identificado necessidades de algum ajuste, inicia-se o ciclo novamente.

3.5 Fechamento

Finalmente no encerramento apresentam-se os relatórios das evidências do ROI para a organização, os objetivos alcançados e dos objetivos não alcançados com suas respectivas justificativas. As lições aprendidas são registradas para serem utilizadas no futuro e a sugestão de melhoria continua através de medições.

4. Resultados Obtidos

Os resultados obtidos foram os seguintes:

- Métodos e/ou algoritmos desenvolvidos: A proposta FROISPI foi experimentada em três empresas como um Framework (método);
- Artigos publicados: [Viana, 2008] e [Viana, 2009];
- Dissertações e/ou teses geradas: Dissertação do Mestrado Profissional do Centro de Informática da Universidade Federal de Pernambuco (CIN/UFPE).

5. Aplicabilidade dos Resultados

A pesquisa foi realizada para identificar os benefícios em MPS em três empresas de software. O objetivo do estudo é investigar como FROISPI pode colaborar para demonstrar os aspectos relevantes que justifiquem o ROI em MPS. As duas fases iniciais utilizam um questionário para caracterizar o contexto da empresa, onde são identificadas as características.

O primeiro contexto demonstrado na Tabela 1 é identificar qual o tipo da organização, e quantos empregados possui, se existe alguma avaliação da garantia de qualidade e se a empresa sabe seus problemas e aonde quer chegar.

Tabela 1. As Empresas

Contexto das Empresas
Empresa A. A empresa com sede em João Pessoa/PB, desenvolve Soluções Integradas para gestão públicas. Atuando há mais de 28 anos âmbito nacional. Está implementando o nível G do modelo de melhoria de processo de software brasileiro MPS.BR.
Empresa B. Instituição privada com sede em Manaus/AM, sem fins lucrativos, está voltada para o desenvolvimento de pesquisa e serviços tecnológicos. Possui uma unidade de ensino médio, técnico, superior e pós-graduação e tem colaborado com o desenvolvimento sustentável no Amazonas e na região amazônica. Está implementando o CMMI nível 2.
Empresa C. A empresa com sede em Recife/PE, é um instituto privado de inovação que cria produtos, processos, serviços usando Tecnologia da Informação e Comunicação (TIC). Atuando há mais de 10 anos em âmbito nacional e internacional e possui nível 3 do CMMI.

O segundo contexto demonstrado na Tabela 2 é para identificar os pontos fortes e fracos no processo do software, identificar o conhecimento sobre a última pratica para melhorar o processo e para saber o quanto os Gerentes de Projetos executam sua função da forma apropriada.

Tabela 2. Pontos Fortes e Fracos.

Empresa A	Descrição
Pontos Fortes	**Padronização de Processo de Software** **Melhor controle e reuso de processo** **Maior agilidade na construção de software**
Pontos Fracos	**Fraca representação do processo** **Falta de boas praticas em Gerência de Configuração, Garantia da Qualidade e Medição.**
Empresa B	**Descrição**
Pontos Fortes	**Habilidade em Gerencia de Projetos**
Pontos Fracos	**Processo burocrático**
Empresa C	**Descrição**
Pontos Fortes	**Nível alto em Gerencia de Projetos, Gerência de Configuração e Garantia da Qualidade.** **Diversos guias de processo disponível para os colaborados.** **A organização conhece o processo.**
Pontos Fracos	**O Processo não é aplicado em todos os projetos.** **Falta de ferramentas de automação para o processo.** **Faltam praticas ágeis no processo atual.**

O comportamento dos Gerentes de Projetos é apresentado no gráfico na Figura 2, que caracteriza o nível de responsabilidade e de comprometimento com a função. A escala vertical é para o comprometimento onde o número 4 é Totalmente, o número 3 Muito, o numero 2 Mediano, o numero 1 Pouco e o numero 0 Baixíssimo.

Figura 2. Comportamento dos Gerentes de Projetos.

Os pontos horizontais são: 1) satisfação do cliente; 2) requisitos do software; 3) controle das mudanças; 4) marco dos projetos; 5) orçamento e custo; 6) qualidade de software; 7) produtividade; 8) satisfação da equipe e 9) imagem de companhia. Os dados apresentados nas tabelas e no gráfico foram coletados nos questionários das duas primeiras fases.

6. Características Inovadoras

A proposta de investigar os aspectos relevantes em MPS é inovador e de certa forma ousado, pois as pretensões implícitas nos objetivos, tendência a uma exploração mais aprofundada do tema com a expectativa de criar um novo conhecimento para definir um modelo de estimativa de ROI em MPS. Por enquanto, a implementação do FROISPI esta fortemente associada à expertise de Engenheiro de Software. Pois as dificuldades durante a pesquisa não se limitaram somente na demora do preenchimento dos questionários, mas um certo grau de resistência das equipes envolvidas.

O contexto dos resultados esperados é personalizado por cada empresa, pois na pesquisa os problemas encontrados nas empresas não foram similares. Além disso, o resultado do projeto piloto passar ser o grande diferencial, pois irá determinar se a empresa conseguiu ou não melhorar seu processo de software.

Pôde-se observar que a empresa que possuía uma equipe mais experiente em processo de software conduziu o experimento com maior objetividade, mesmo no envolvimento de mais de um processo no projeto piloto de MPS. Acredita-se que os benefícios intangíveis serão identificados pelas empresas, compreendendo a importância desses benefícios: houve aumento de conhecimento; e lições aprendidas.

7. Conclusões e Perspectivas Futuras

O contexto dos resultados esperados é personalizado por cada empresa, pois na pesquisa os problemas encontrados nas empresas não foram similares. Além disso, o resultado do projeto piloto passar ser o grande diferencial, pois irá determinar se a empresa conseguiu ou não melhorar seu processo de software.

Pôde-se observar que a empresa que possuía uma equipe mais experiente em processo de software conduziu o experimento com maior objetividade, mesmo no envolvimento de mais de um processo no projeto piloto de MPS. Acredita-se que os benefícios intangíveis serão identificados pelas empresas, compreendendo a importância desses benefícios: houve aumento de conhecimento; e lições aprendidas Como trabalho futuro, O FROISPI será submetido a um novo experimento com dez empresas, observando um número maior de projetos pilotos para comparar os resultados.

8. Referências

CECOA. ELearning Quality for SME's: Guidance and Counseling. Nova Formação Magazine 6.Ed.2005

ISO – International Organization for Standardization. *Software Process Assessment*. ISO/IEC TR-15504. 1998.

CMU/SEI. *Capability Maturity Model® Integration (CMMI-SM), Version 1.2. CMMISM for Software Engineering - Staged Representation.* Carnegie Mellon University – Software Engineering Institute, 2007.

Fuggetta, A. *Software process: a roadmap. In: The Future of Software Engineering*, A. Finkkelstein (ed), 2000

Hamblin, A. Avaliação e Controle do Treinamento. São Paulo: McGraw Hill do Brasil, 1978

Kirkpatrick, Donald. *Evaluating Training Programs: The Four Levels*. San Francisco: Berrett-Koehler, 1994

Phillips, Jack J., Phillips, P. *Return on Investment Measures Success*. Industrial Management Magazine. Published at [WWW document]. URL http://www.roiinstitute.net

Phillips, Jack J. *Return on Investment; in training and Performance Improvement Programs*, 2.ed. Amsterdam: Butterworrth Heinemann, 2003

Rico, David. *ROI of Software Process Improvement: metrics for project managers and software engineers*. J. Ross Publishing, 2004.

Raytheon. Presentation. *CMMI Level 5 Return on Investment for Raytheon N TX,* 2004 SOFTEX, *MPS.BR-Melhoria de Processo do Software Brasileiro – Guia Geral*, Versão 1.2, 2007. Disponível em: http://www.softex.br/mpsbr

SOFTEX, iMPS : resultados de desempenho de organizações que adotaram o modelo MPS / Guilherme Horta Travassos e Marcos Kalinowski. – Campinas, SP :

Associação para Promoção da Excelência do Software Brasileiro – SOFTEX, 2008. Disponível em: http://www.softex.br/mpsbr

Solingen, R. e Berghout, E. *The Goal/Question/Metric Method; a practical guide for quality improvement of software development*. London: McGraw-Hill. 1999.

Suwardy, T., Ratbatunga, J., Sohal, A. S. e Speight, G. *IT projects: evaluation, outcomes and impediments*. *Benchmarking: An International Journal,* v.10, n.4, p.325-42, 2003.

Viana, P.W.P., Vasconcelos A.M.L. *FROISPI – Framework Return on Investment of Software Process Improvement.* VI Workshop de Teses e Dissertações em Qualidade de Software (WTDQS) no Simpósio Brasileiro de Qualidade de Software (SBQS) 2008.

Viana, P.W.P., Vasconcelos A.M.L. *Aspectos de Retorno de Investimento em Melhoria de Processo de Software*. I Escola Regional de Informática – Regional Norte (ERIN) 2009.

Gestão Quantitativa de Pessoas em Desenvolvimento Distribuído de Software

Rafael Prikladnicki

Faculdade de Informática – PUCRS – POA – RS – Brasil

rafaelp@pucrs.br

Resumo. *Neste artigo apresenta-se o relatório final do projeto 6.12, cujo objetivo foi estudar uma forma de quantificar a distância percebida em equipes distribuídas de desenvolvimento de software. Acredita-se que o modelo proposto, a partir dos resultados alcançados até então, pode ser útil para auxiliar na gestão de projetos distribuídos de desenvolvimento de software.*

1. Introdução

Desenvolver software com qualidade, dentro dos prazos e recursos estipulados, tornou-se um dos grandes desafios para as organizações. A concorrência no mercado aumenta gradativamente e necessita-se diminuir os custos de desenvolvimento, mantendo a qualidade. Quando se fala sobre gerência de projetos, um tópico fundamental é a gestão de pessoas. Em um cenário de gerência de projetos distribuídos, gerenciar as expectativas e conflitos nas equipes torna-se ainda fundamental, dada a distribuição geográfica e conseqüente diminuição de contato físico.

Times distribuídos de desenvolvimento de software estão se tornando comuns nos atuais projetos de software. Mas segundo Gutwin & Greenberg (2002) é fundamental reconhecer o fator humano de equipes distribuídas. Os autores defendem que manter colaboradores geograficamente distantes envolvidos e engajados no projeto aumenta muito as chances do projeto dar certo. Utilizando as ferramentas disponíveis e mantendo vivo o espírito de trabalho em equipe, times distribuídos podem ser tão bem sucedidos quanto os times alocados em um mesmo espaço físico.

Sendo assim, gestão de pessoas, do ponto de vista de gerência de projetos distribuídos, é um assunto que tem sido cada vez mais explorado. Muito se tem comentado sobre os desafios técnicos e não técnicos do Desenvolvimento Distribuído de Software (DDS), mas pouco tem sido feito para minimizar os efeitos dos fatores da distância na gestão das pessoas – dentre os quais podemos citar, principalmente, a cultura, a confiança e a comunicação (Carmel, 1999; Herbsleb & Moitra, 2001). Em parte, tanto a academia como a indústria têm concentrado esforços no desenvolvimento de soluções técnicas para aliviar alguns dos problemas encontrados em projetos de DDS. Por outro lado, boa parte das dificuldades associadas a questões principalmente não técnicas tem sido resolvidas a partir da experiência e bom senso dos profissionais responsáveis pelo gerenciamento das equipes neste contexto.

Para minimizar o lado negativo dos fatores supracitados, tem-se utilizado abordagens com foco em dados qualitativos e análise subjetiva das situações. Mas muitas vezes isto não tem se mostrado suficiente. É neste contexto que surgiu este projeto, a partir de uma idéia de explorar formas quantitativas de apoiar a gestão de pessoas em um cenário de DDS. Sendo assim, o objetivo deste artigo é reportar os resultados alcançados com o projeto 6.11, cujo objetivo foi estudar uma forma de quantificar a distância percebida em equipes distribuídas de desenvolvimento de software. Na seqüência do artigo apresentam-se os objetivos e a justificativa do projeto, a metodologia de execução, os resultados obtidos, a aplicabilidade dos resultados e as conclusões.

2. Objetivos e Justificativa

O Desenvolvimento Distribuído de Software (DDS) é uma área recente do ponto de vista da Engenharia de Software, mas tem se firmado como uma grande oportunidade atualmente. Diversas empresas têm optado por desenvolver software contando com equipes distribuídas em diversos locais, seja no mesmo país, ou em um cenário global, visando, entre outros fatores, redução de custos, maior flexibilidade e ganhos de escala (Carmel, 1999). Ao mesmo tempo em que esta é uma oportunidade de negócio interessante, são necessárias adaptações na forma de trabalho, principalmente no uso de técnicas de Engenharia de Software e Gerência de Projetos (Damian & Moitra, 2006).

Sendo assim, as dificuldades impostas pelo DDS têm motivado pesquisadores e profissionais tanto na academia como na indústria a buscarem soluções para minimizar o efeito do DDS (Audy & Prikladnicki, 2007). Adaptação do processo de desenvolvimento, investimento em gerência de risco, treinamentos em gestão de projetos distribuídos, reconhecimento da importância de fatores tais como diferenças culturais, comunicação, confiança, são alguns exemplos do que tem sido feito.

Do ponto de vista de gestão de equipes de DDS, Carmel (1999) sugere a existência de cinco forças, chamadas de forças centrífugas, que devem ser bem gerenciadas para garantir o sucesso de uma equipe de DDS. Evaristo et al (2004) por sua vez enfatizam que um aspecto importante na avaliação da dispersão é a distância percebida entre as equipes distribuídas (além da distância física). Neste contexto, o objetivo deste projeto foi

explorar e propor uma forma quantitativa de auxiliar os gerentes de projetos na gestão das equipes de DDS. Desta forma, foi proposto um modelo chamado PDI (*Perceived Distance Index*) para apoio na avaliação quantitativa da distância percebida definida por Evaristo et al (2004), a partir das cinco forças centrífugas propostas por Carmel (1999). O modelo está baseado na aplicação de questionários e fórmulas matemáticas para calcular o índice da distância percebida em equipes de DDS, representando o fator de percepção de distância de cada colaborador em relação à equipe, e da própria equipe. A seção a seguir apresenta a metodologia de execução seguida durante este projeto.

3. Metodologia de Execução

O projeto foi dividido em cinco fases, quais sejam:

Fase 1 - Revisão bibliográfica: inicialmente, foi desenvolvida uma revisão bibliográfica destinada a ampliar a visão sobre o assunto, permitindo a formação de um conhecimento coerente e embasado. Como fontes bibliográficas de sustentação do estudo foram utilizados artigos publicados em congressos e periódicos nacionais e internacionais, com foco especial no workshop em Desenvolvimento Distribuído de Software (WDDS) e na *International Conference on Global Software Engineering* (ICGSE). Parte deste estudo foi documentada em Prikladnicki & Audy (2007), Prikladnicki & Audy (2008a) e Prikladnicki & Audy (2008b).

Fase 2 - Proposta de modelo preliminar: a partir da revisão de literatura, um modelo preliminar de avaliação de distância percebida foi proposto em meados de 2007.

Este modelo está documentado em Prikladnicki & Audy (2007).

Fase 3 - Avaliação do modelo preliminar: o modelo preliminar foi avaliado de duas formas. Inicialmente, ele foi posto em prática, sendo avaliado com dados reais em uma empresa multinacional com matriz na Índia. A avaliação ocorreu na filial em Curitiba e está documentada em Prikladnicki & Audy (2008a). Além disso, o modelo foi avaliado por especialistas da área de DDS (Desenvolvimento Distribuído de Software), sendo um profissional de empresa multinacional (nos Estados Unidos), um pesquisador (Irlanda), e uma empresa que atua em DDS (na Índia). Estas avaliações foram realizadas de forma presencial, sendo que as avaliações com o profissional americano e o pesquisador Irlandês ocorreram em Bangalore, quando este autor teve a possibilidade de participar de um evento na área de DDS organizado naquele local (ICGSE 2008). As avaliações estão documentadas em Prikladnicki & Audy (2008a) e Prikladnicki & Audy (2008b).

Fase 4 - Modelo consolidado: a partir do retorno recebido, melhorias foram incorporadas ao modelo PDI, e a coleta de dados na empresa em Curitiba foi atualizada para refletir algumas mudanças. O modelo consolidado está documentado em Prikladnicki & Audy (2008b).

Fase 5 - Ferramenta de apoio: para facilitar a coleta e análise de dados, duas ferramentas foram planejadas, a partir de trabalhos de conclusão de curso na Faculdade de Informática da PUCRS. A ferramenta para apoio a coleta de dados foi planejada no contexto de um trabalho de conclusão no Bacharelado em Ciência da Computação e está sendo finalizada. Já a ferramenta para análise de dados (QUAMP) foi desenvolvida no contexto de uma monografia de conclusão do curso de especialização em Gerenciamento de Projetos com Ênfase em TI (DalSotto, 2009). Futuramente as ferramentas serão integradas, o que não foi possível no ciclo 2008 devido ao tempo limitado para o desenvolvimento dos trabalhos por parte dos alunos.

4. Resultados Obtidos

O projeto produziu os seguintes resultados:

- **produtos de software gerados (módulos ou programas de computador resultantes, disponibilizados para o mercado):** duas ferramentas de apoio ao modelo proposto, sendo uma para a coleta e a outra para a análise dos dados.

- **outros produtos gerados (que foram disponibilizados para o mercado):** desenvolveu-se o modelo PDI, disponibilizado para avaliação em três empresas multinacionais, sendo uma em Bangalore (Índia), uma em Porto Alegre e uma em Curitiba.

- **métodos ou algoritmos desenvolvidos:** foram desenvolvidos algoritmos para apoio a análise dos dados gerados pelo modelo, a partir do modelo matemático desenvolvido e documentado nas publicações resultantes do projeto.

- **artigos publicados:** este projeto gerou diversos artigos. O primeiro deles foi publicado em 2007 no WDDS (Workshop em Desenvolvimento Distribuído de Software), organizado como evento paralelo ao Simpósio Brasileiro de Engenharia de Software (Prikladnicki & Audy, 2007). Este artigo documentou a idéia inicial do modelo PDI. Especificamente no ciclo 2008, outro artigo foi publicado no WDDS 2008, documentando a primeira aplicação do modelo PDI (Prikladnicki & Audy, 2008a).

Além disso, este artigo foi estendido e publicado em um periódico de circulação nacional (Prikladnicki & Audy, 2008b). Mais recentemente, um pôster foi aceito para apresentação no CHASE 2009 (Workshop on the Cooperative and Human Aspects of Software Engineering), propondo a aplicação do PDI para equipes de desenvolvimento de software em geral, e não apenas para contexto de projetos distribuídos (Priklandicki,

2009a). Ainda, um artigo foi submetido e está em avaliação na ICGSE 2009 (International Conference on Global Software Engineering). Este artigo formaliza o modelo para a comunidade internacional de Engenharia de Software, contento as últimas atualizações e avaliações (Prikladnici, 2009b). Por fim, um último artigo está sendo preparado para ser submetido para uma edição especial de um periódico de circulação internacional. Neste artigo serão incorporados os resultados do artigo publicado no CHASE e uma avaliação realizada no final de 2008 com equipes de três projetos de uma das empresas multinacionais onde o modelo foi disponibilizado.

- **recursos humanos capacitados:** no escopo deste projeto, um trabalho de especialização foi finalizado no início de 2009 (DalSotto, 2009). Outro trabalho foi desenvolvido no contexto do trabalho de conclusão de curso de um aluno do Bacharelado em Ciência da Computação da PUCRS. Na oportunidade, ambos os alunos faziam parte do quadro de colaboradores de empresas que possuem projetos distribuídos. Sendo assim, além da capacitação gerada pela execução dos trabalhos em si, foi uma oportunidade de ter contato com o estado da arte de DDS, melhorando o conhecimento dos alunos no tema. Adicionalmente, as palestras e apresentações organizadas nas três empresas multinacionais permitiram a capacitação dos profissionais destas empresas em DDS, especificamente na gestão de projetos distribuídos.

- **dissertações e/ou teses geradas:** foi gerado um volume de monografia de conclusão de curso de especialização da PUCRS (DalSotto, 2009) e está para ser publicado um volume de trabalho de conclusão de curso de graduação.

- **eventuais parcerias ou programas de transferência de tecnologia efetuados:** tanto o modelo como as ferramentas serão utilizados em pesquisas desenvolvidas no grupo MuNDDoS da Faculdade de Informática da PUCRS (www.inf.pucrs.br/munddos). Além disso, o modelo será avaliado de forma mais abrangente, a partir do interesse demonstrado por profissionais atuantes em DDS quando da participação do coordenador do projeto em alguns eventos durante o ano de 2008 (EQPS Florianópolis e ICGSE). Por fim, além das parcerias já estabelecidas, novas parcerias estão sendo prospectadas de forma a avaliar o modelo em larga escala.

5. Aplicabilidade dos resultados

Esta versão do modelo PDI foi proposta com foco em equipes de projeto de DDS.

Alguns estudos já foram iniciados e indicam a possibilidade de usar o modelo para qualquer equipe de desenvolvimento de software (não necessariamente distribuída).

Naturalmente, o modelo possui algumas limitações, documentadas nas publicações realizadas, e que estão sendo tratadas dentro do objetivo de mantê-lo em constante refinamento.

As avaliações realizadas até o momento indicam que o modelo pode ajudar a identificar comportamentos não tão visíveis nas equipes dos projetos, auxiliando os gerentes na gestão das equipes. Isto foi percebido a partir da apresentação dos resultados para as equipes onde os dados foram coletados. Como exemplo, descobriu-se que, em um determinado projeto com parte da equipe no Brasil e parte da equipe na Índia, o principal desafio no Brasil era relacionado comunicação, enquanto que na Índia era relacionado à confiança, o que foi corroborado com os integrantes destas equipes.

6. Características inovadoras

Da mesma forma como em qualquer projeto, um projeto distribuído é executado por pessoas. A distância percebida é um aspecto importante que precisa ser considerado neste contexto, mas a quantificação da percepção de distância (psicológica, emocional e sensorial) muitas vezes é difícil e pode não refletir a realidade das equipes. Ao mesmo tempo, a gestão dos fatores humanos é um fator crítico de sucesso em DDS. Por este motivo, este projeto inova ao incorporar a quantificação da distância percebida como um insumo adicional para os gerentes de projeto e para a organização como um todo.

Para evitar replicação de conteúdo já publicado, informações específicas sobre o modelo PDI e como ele foi proposto podem ser encontradas nas publicações geradas ao longo do projeto.

Os dados gerados a partir do modelo abrem a possibilidade de explorar os comportamentos das equipes de projeto, diferentes percepções das pessoas, correlação dos índices de distância percebida com dados de projeto (atrasos, custo, número de defeitos), entre outros. Não se tem conhecimento de iniciativas similares sendo desenvolvidas. Existem apenas estudos que analisam a importância de se considerar a percepção de distância ou proximidade entre as equipes, e não apenas a distância física efetivamente existente (Wilson et al, 2008).

7. Conclusões e Perspectivas Futuras

Neste artigo apresentou-se o relatório final do projeto submetido ao ciclo 2008 do PBQP Software, cujo objetivo foi desenvolver um modelo para apoiar a gestão de pessoas em projetos de DDS, a partir de uma abordagem quantitativa. O modelo foi desenvolvido a partir da experiência da equipe de pesquisa em projetos de DDS e estudos anteriores desenvolvidos (Prikladnicki & Audy, 2006; Prikladnicki & Audy, 2004).

Entende-se que o uso do modelo proposto pode ajudar no melhor gerenciamento de uma equipe distribuída (seja ela local ou global), avaliando como cada colaborador percebe a distância existente e os fatores que causam a maior percepção de distância.

Uma versão preliminar do modelo foi apresentada no WDDS 2007 (Prikladnicki & Audy, 2007), enquanto que uma primeira avaliação foi discutida no WDDS 2008 (Prikladnicki & Audy, 2008). Neste relatório foram acrescentadas informações acerca da consolidação das atividades realizadas ao longo do ciclo 2008.

Em relação a perspectivas futuras, este modelo apresenta-se como um instrumento importante para que os gerentes de projetos possam ter mais insumos para identificar fraquezas em suas equipes distribuídas e diferentes perspectivas que não são claramente percebidas através de uma avaliação puramente qualitativa. Os próximos passos são a consolidação das ferramentas desenvolvidas para apoiar a coleta e análise dos dados, novas replicações e análise de dados, além de estudos mais aprofundados para avaliar a efetividade do modelo na gestão das equipes de DDS. Pretende-se também disponibilizar uma versão completa do modelo e das ferramentas para uso na indústria.

8. Agradecimentos

Este projeto foi realizado no contexto do grupo de pesquisa em DDS do PDTI, financiado pela Dell Computadores do Brasil Ltda., com recursos da Lei Federal Brasileira nº 8.248/91. Agradecemos as empresas que forneceram dados para as avaliações do modelo, bem como os alunos envolvidos no desenvolvimento das ferramentas de apoio

9. Referências Bibliográficas

Audy, J., Prikladnicki, R. (2007) “Desenvolvimento Distribuído de Software: desenvolvimento de software com equipes distribuídas”, Série livros didáticos Campus-SBC, Rio de Janeiro: Campus/Elsevier.

Carmel, E. (1999). “Global Software Teams – Collaborating Across Borders and Time- Zones”, EUA: Prentice Hall.

DalSotto, F. R. (2009). “Quamp Analyzer – Analisando A Distância Percebida em Equipes Distribuídas de Desenvolvimento de Software”, Monografia de Conclusão, Especialização em Gerenciamento de Projetos com Ênfase em TI, Faculdade de Informática, PUCRS.

Damian, D., Moitra, D. (2006). “Guest Editors' Introduction: Global Software Development: How far Have We Come?”, IEEE Software, 23(5), pp.17-19.

Evaristo, R., Scudder, R., Desouza, K., Sato, O. (2004). “A Dimensional Analysis of Geographically Distributed Project Teams: A Case Study”, Journal of Engineering and Technology Management. 21(3), pp. 75-189.

Gutwin, C., Greenberg, S. (2002) “A Descriptive Framework of Workspace Awareness for Real-Time Groupware”. *JCSCW*, *Issue 3-4, p. 411-446.*

Herbsleb, J. D., Moitra, D. (2001). “Guest Editors' Introduction: Global Software Development”, IEEE Software, 18(2), pp. 16-20.

Prikladnicki, R., Audy, J. L. N. (2008a). “Gestão Quantitativa de Pessoas em DDS: primeira aplicação de um modelo para o cálculo da distância percebida relativa em equipes distribuídas de desenvolvimento de software”, II Workshop em DDS (WDDS), Campinas, Brasil.

Prikladnicki, R., Audy, J. L. N. (2008b). “Uma Abordagem Quantitativa para Gerenciar a Distância Percebida em Equipes Distribuídas de Desenvolvimento de Software”, Infocomp Journal of Computer Science, Special Edition, Nov 2008, p. 51-60.

Prikladnicki, R., Audy, J. L. N. (2007). “Um Modelo para o Cálculo da Distância Percebida Relativa em Equipes Distribuídas de Desenvolvimento de Software”, I Workshop em DDS (WDDS), João Pessoa, Brasil.

Prikladnicki, R., Audy, J. L. N. (2006). “Uma Análise Comparativa de Práticas de Desenvolvimento Distribuído de Software no Brasil e no exterior”, XX SBES, Florianópolis.

Prikladnicki, R., Audy. Jorge L. N. (2004). “MuNDDoS: Um Modelo de Referência para Desenvolvimento Distribuído de Software”, XVIII SBES, Brasília.

Wilson, J. M., O’Leary, M. B., Metiu, A., Jett, Q. R. (2008). “Perceived Proximity in Virtual Work: Explaining the Paradox of Far-but-Close,” Organization Studies, 29(07), pp. 979-1001

Artigos dos Projetos Concluídos
Ciclo 2009

Processo de Contratação de Software e Serviços Correlatos para Entes Governamentais.

Cláudio Silva da Cruz[1], Edméia Leonor Pereira de Andrade[2], Rejane Maria da Costa Figueiredo[3]

[1]Tribunal de Contas da União (TCU)

[2]Empresa Brasileira de Pesquisa Agropecuária (Embrapa)

[3] Faculdade FGA, Campus Gama – Universidade de Brasília (UnB).

cscruz@cscruz.org, edmeia.andrade@embrapa.br, rejanecosta@unb.br

Abstract. *The Brazilian Federal Public Administration is a large IT services contractor and must adopt formal processes for IT outsourcing. MPS.BR-Acquisition Guide outsourcing process model do not comprehend legal requirements for the public sector. This research aimed to propose an outsourcing process model derived from MPS.BR-Acquisition Guide and aligned to legal requirements. The developed model has four phases: IT planning, contracting planning, provider selection and contractual management. The phases were divided onto activities and tasks and two examples of tasks are presented in this article.*

Resumo. *A Administração Pública Federal brasileira é grande contratadora de serviços de tecnologia da informação (TI) e deve adotar processos formais para esse tipo de contratação. O modelo de processo de contratação do MPS.BR-Guia de Aquisição:2009 não contempla as exigências do atual marco legal. O objetivo desse projeto foi propor um modelo de processo de contratações de serviços de TI derivado do Guia de Aquisição e alinhado ao marco legal. O modelo desenvolvido possui quatro fases: planejamento de TI, planejamento da contratação, seleção do fornecedor e gestão contratual. As fases foram subdivididas em atividades e tarefas e dois exemplos de tarefas são apresentados no presente artigo.*

1 - Introdução

O processo de contratação de software é considerado fundamental no modelo de ciclo de vida de software definido pela ISO/IEC (ABNT, 2009).

No entanto, trata-se de disciplina ainda bastante desconhecida dos profissionais de tecnologia da informação (TI) do setor público, provavelmente em decorrência da elevada demanda por conhecimento jurídico e administrativo que envolve (CRUZ, 2008).

Para Cruz (2008), a capacitação dos gestores de contratos de TI e a existência de um processo formal de contratação são precondições para que as contratações públicas se dêem com boa aderência legal, entre outras.

2 - Objetivo e justificativa

O objetivo do presente projeto foi 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.

Justificativa: a Administração Pública Federal gasta pelo menos R$ 6 bilhões por no em tecnologia da informação (Acórdão nº 1.934/2007-TCU-Plenário), sendo parte significativa desse orçamento consumido com serviços relacionados a software. No controle do gasto desse orçamento, o Tribunal de Contas da União detectou elevada frequência de irregularidades e impropriedades, pelo que recomendou a adoção de processos de contratação mais maduros (Acórdão nº 786/2006-TCU-Plenário). De fato, ainda há carência de um processo governamental de aquisição de software e serviços correlatos, pois mesmo o modelo de processo do MPS.BR-Guia de Aquisição (SOFTEX, 2009) ainda não contempla as exigências do atual marco legal, como a Instrução Normativa SLTI/MP nº 04/2008 (Brasil, 2008). Portanto, é oportuna a realização de projeto de pesquisa para obter um processo que facilite a adoção das boas práticas do MPS-BR-Guia de Aquisição pelos entes públicos, cujas aquisições representam hoje parcela importante do mercado brasileiro de serviços de software.

3 - Metodologia de execução

Para alcance do objetivo geral, foram definidos os seguintes objetivos específicos: (a) identificar o marco legal relevante para as aquisições de software e serviços correlatos; (b) identificar as necessidades de adequação do MPS.BR-Guia de Aquisição ao marco legal, segundo o COBIT ME3 (ITGI, 2007); e (c) ajustar o processo com base na versão do MPS.BR-Guia de Aquisição (SOFTEX, 2009) e no marco legal e definir artefatos para sua aplicação prática.

A identificação do marco legal tomou por base o levantamento de Cruz (2008, p. 90), que incluiu a análise de 64 documentos legais e 57 documentos jurisprudenciais. A esses ainda foram acrescidas normas recentes, como a Instrução Normativa SLTI/MP nº 02/2008 (Brasil, 2008a) e a nº 04/2008 (Brasil, 2008b), e jurisprudência recente, como os acórdãos 1.603 e 2.471/2008-TCU-Plenário.

A identificação das necessidades de adequação do Guia de Aquisição e o ajuste do processo foram realizados com base em comparações entre os elementos extraídos do marco legal, especialmente o Quadro Referencial Normativo (QRN) (CRUZ, 2008) e a Instrução Normativa SLTI/MP nº 04/2008 (Brasil, 2008), e o Guia de Aquisição do MPS.BR (SOFTEX, 2009), à semelhança das pesquisas concluídas pelos pesquisadores (ANDRADE; SCALET, 2007; CRUZ, 2008; GILLIS; ANDRADE, 2009; SCALET; ANDRADE, 2009; SOUSA et al., 2005), e também com base no conhecimento advindo de atividades de ensino (CRUZ, **Erro! Fonte de referência não encontrada.**, **Erro! Fonte de referência não encontrada.**, **Erro! Fonte de referência não encontrada.**, **Erro! Fonte de referência não encontrada.**) e em experiências práticas dos autores deste trabalho.

Para a definição do modelo de processo foi empregada a estrutura de processo definida pela norma ABNT NBR ISO/IEC 12207:2009 (ABNT, 2009), que é padrão internacional para processos do ciclo de vida de software, e pelo Guia de Aquisição do MPS.BR (SOFTEX, 2009). Para cada fase foram descritos o propósito, as atividades e as respectivas tarefas.

4 - Resultados obtidos

Mais de cem documentos do marco legal brasileiro foram identificados e analisados. Ao comparar o Guia de Aquisição do MPS.BR com o marco legal levantado, as principais lacunas identificadas foram:

- ausência de clara vinculação entre a aquisição e as diretrizes estabelecidas no planejamento estratégico de TI;
- ausência de identificação clara de papéis relevantes e seus atores, tais como gestor do contrato, requisitante da contratação, autoridade competente, fiscal do contrato etc.;
- ausência de previsão da fase licitatória;
- menor rigor com a formalização e a motivação do que o exigido pela legislação para o setor público;
- ênfase na negociação com fornecedores, o que não é possível no setor público;
- ausência de rigor formal na fase de gestão do contrato (p.ex. uso da ordem de serviço).

Essas e outras lacunas foram tratadas no modelo proposto. O **Erro! Fonte de referência não encontrada.** apresenta a estrutura do processo proposto, que consiste de 4 fases, subdivididas em atividades e tarefas, que podem ser adotadas para contratação de qualquer tipo de serviço de TI no setor público.

Quadro 1. Estrutura geral do modelo de processo de contratação proposto

ATIVIDADE	TAREFA
FASE 1. PLANEJAMENTO DE TI	
1.1 Identificar as necessidades de negócio 1.2 Definir diretrizes de TI alinhadas ao negócio 1.3 Definir macro-ações de TI 1.4 Identificar as principais contratações de TI 1.5 Reservar os recursos (RH, orçamento, prioridades etc.)	
FASE 2. PLANEJAMENTO DA CONTRATAÇÃO	
2.1 Analisar a viabilidade da contratação	2.1.1 Analisar e justificar a necessidade 2.1.2 Fundamentar a motivação 2.1.3 Levantar e especificar requisitos 2.1.4 Analisar soluções de TI 2.1.5 Analisar preliminarmente riscos e impactos 2.1.6 Selecionar a solução de TI e justificar
2.2 Elaborar o plano de sustentação	2.2.1 Elaborar proced. de segurança da informação 2.2.2 Elaborar proced. de gestão de recursos 2.2.3 Elaborar proced. de transfer. de conhecimento 2.2.4 Elaborar proced. de transição contratual 2.2.5 Elaborar proced. de continuidade de negócio
2.3 Elaborar a estratégia de contratação	2.3.1 Indicar o tipo de contratação 2.3.2 Definir termos contratuais 2.3.3 Definir a estratégia de independência 2.3.4 Indicar o gestor do contrato 2.3.5 Definir as responsabilidades da contratada

	2.3.6 Definir orçam. detalhado e fonte de recursos 2.3.7 Definir os critérios técnicos de seleção
2.4 Analisar e tratar riscos	2.4.1 Identificar riscos da licitação 2.4.2 Identificar riscos dos serviços 2.4.3 Analisar probabilidade e impacto dos riscos 2.4.4 Planejar ações de mitigação de riscos 2.4.5 Planejar ações em contingências
2.5 Analisar e demonstrar a economicidade da contratação 2.6 Elaborar e encaminhar a minuta do termo de referência (ou do projeto básico)	
FASE 3. SELEÇÃO DO FORNECEDOR	
3.1 Formalizar e aprovar o termo de referência (ou o projeto básico)	
3.2 Selecionar fornecedor (por contratação direta)	3.2.1 Justificar a contratação direta 3.2.2 Negociar o contrato 3.2.3 Obter o exame e a aprovação jurídicos 3.2.4 Obter a aprovação da contratação
3.3 Selecionar fornecedor (por licitação)	3.3.1 Elaborar o edital 3.3.2 Obter o exame e a aprovação jurídicos 3.3.3 Realizar consulta ou audiência pública 3.3.4 Realizar ajustes no edital 3.3.5 Publicar o edital 3.3.6 Responder a questionamentos/recursos 3.3.7 Receber e julgar as propostas 3.3.8 Adjudicar o objeto 3.3.9 Homologar a licitação
3.4 Ajustar a minuta do contrato 3.5 Assinar o contrato	
FASE 4. GESTÃO DO CONTRATO	
4.1 Iniciar o contrato	4.1.1 Elaborar o plano de inserção 4.1.2 Repassar conhecimento 4.1.3 Alocar infraestrutura 4.1.4 Designar as pessoas para papéis 4.1.5 Realizar a reunião de iniciação
4.2 Encaminhar demandas	4.2.1 Receber demandas de negócio 4.2.2 Elaborar a ordem de serviço 4.2.3 Assinar a ordem de serviço
4.3 Realizar o monitoramento técnico	4.3.1 Acompanhar a execução da ordem de serviço 4.3.2 Gerenciar riscos 4.3.3 Determinar ação corretiva 4.3.4 Providenciar alterações da ordem de serviço
4.4 Executar a atestação técnica	4.4.1 Receber os serviços concluídos 4.4.2 Avaliar os serviços 4.4.3 Rejeitar serviços inadequados 4.4.4 Aceitar serviços adequados
4.5 Realizar o monitoramento administrativo	4.5.1 Acompanhar os serviços em execução 4.5.2 Avaliar cumprimento do contrato 4.5.3 Aplicar sanções 4.5.4 Corrigir desatendimento contratual 4.5.5 Calcular valores dos serviços e impostos 4.5.6 Liquidar a despesa 4.5.7 Autorizar emissão de fatura/nota fiscal 4.5.8 Verificar regularid.fiscal/comercial/trabalh./previd. 4.5.9 Efetuar pagamento dos serviços
4.6 Tratar as demandas por alterações contratuais	4.6.1 Tratar alteração de quantitativos 4.6.2 Tratar alteração de prazos de execução 4.6.3 Tratar alteração de cláusulas contratuais 4.6.4 Tratar alteração técnica 4.6.5 Tratar readequação de preços 4.6.5.1 Tratar reequilíbrio econômico-financeiro 4.6.5.2 Tratar reajuste de preço 4.6.5.3 Tratar repactuação 4.6.6 Tratar prorrogação do contrato

	4.6.7 Tratar suspensão do contrato 4.6.8 Tratar rescisão do contrato
4.7 Realizar o encerramento contratual e a transição	4.7.1 Realizar proced. de transferência de conhecimento 4.7.2 Realizar proced. de transferência de recursos 4.7.3 Realizar proced. de segurança da informação 4.7.4 Verificar a quitação de encargos trabalh./previd.

4.1 - Exemplo de descrição de tarefa

Sucintamente são apresentados dois exemplos de descrição de tarefa. Elegeram-se as tarefas "analisar e justificar a necessidade" e "levantar e especificar os requisitos" da atividade "analisar a viabilidade da contratação" da fase de "planejamento da contratação" para facilitar ao leitor a comparação com a versão original do Guia de Aquisição.

Para cada tarefa foi definido um identificador (Id) que representa as três letras iniciais da fase do processo "planejamento da contratação" (Pla), seguida da letra t, indicando uma tarefa, e a sua numeração, conforme apresentado a seguir.

Quadro 2. Exemplos de detalhamento de tarefa

Id.	Tarefa
Pla-t1	**Analisar e justificar a necessidade** **Descrição:** Estabelecer as necessidades a serem atendidas por meio da contratação, desenvolvimento ou melhoria de um produto ou serviço de software. Durante essa tarefa são analisadas o efetivo escopo das necessidades e resultados que a organização pretende atingir com o projeto de contratação. Esta tarefa é fundamental, pois indica a primeira tomada de decisão quanto ao prosseguimento do projeto e que resultados são esperados pela organização após a efetivação da contratação. **Produtos requeridos:** Pti-p1 – Identificação das necessidades de negócio Pti-p2 – Ações de TI priorizadas **Responsável:** Área de Negócio **Participantes:** Área de TI **Produtos gerados:** Pla-p1 – Documento da análise e justificativa da necessidade da contratação **Legislação relacionada:** Decreto nº 2.271/1997, art. 2º (Brasil, 1997); Decreto nº 3.555/2000, art. 8º, III, "b" (Brasil, 2000); Instrução Normativa SLTI/MP nº 04/2008, art. 10, I e II (Brasil, 2008b).
Pla-t2	**Levantar e especificar os requisitos** **Descrição:** Identificar e especificar os requisitos do cliente para a contratação do produto ou serviço de software. Se necessário, as organizações poderão solicitar informações de fornecedores ou realizar pesquisas para identificar as melhores práticas de outras organizações, que adquiriram produtos e serviços semelhantes, visando identificar os requisitos a partir de soluções disponíveis no mercado. Durante esta tarefa devem ser especificados os seguintes requisitos a serem considerados no projeto de contratação: • **dos interessados** (*stakeholders*): as necessidades devem ser transformadas em requisitos mais específicos que contemplem os diversos tipos de interessados, tais como, usuários, planejadores, gestores, desenvolvedores e beneficiários do produto de software; • **do sistema**: requisitos envolvendo processos, hardware, software, integrações, ambiente e pessoas que irão compor a solução que atenderá as necessidades estabelecidas; • **do software**: requisitos do(s) produto(s) de software que irá(ão) compor o(s) sistema(s) a ser(em) implementado(s). Devem ser especificados os requisitos funcionais; • **de qualidade**: alguns requisitos de qualidade são: usabilidade - estilo ou princípios de diálogo que são aplicáveis; tipo de documentação a ser entregue (on-line, manuais de usuário); portabilidade - regras de portabilidade que deverão ser adotadas (tanto para a parte de servidores quanto para acesso via estações de trabalho); interoperabilidade - integração das aplicações novas com os bancos de dados e aplicações legadas; manutenibilidade: tipos e características dos artefatos gerados, de modo a permitir a manutenção por parte do contratado e facilitar eventuais repasses de conhecimento; • **de projeto**: ciclo de vida a ser adotado, técnicas, metodologias, forma de gestão e de documentação do projeto; • **de manutenção**: requisitos de manutenção do software após a sua entrega; • **de treinamento**: características esperadas e quantidade de treinamento relacionado ao produto de software a ser entregue; e • **de implantação**: descrição dos procedimentos necessários para a implantação do software no ambiente de operação, como, por exemplo, a carga do banco de dados, a implementação numa

	configuração distribuída, entre outros. Além destes requisitos, podem ser considerados outros requisitos e restrições que afetam diretamente o projeto de aquisição como, por exemplo, restrições legais, financeiras, de prazo do projeto e de número de usuários do sistema em operação. O adquirente poderá definir e analisar os requisitos com sua própria equipe ou contratar um fornecedor para executar estas atividades. Neste caso, o adquirente deverá manter a responsabilidade pela aprovação do resultado da especificação dos requisitos. NOTA: Esses requisitos variam de acordo com o tipo da contratação. No caso de contratação de serviços de desenvolvimento e manutenção de escopo aberto, sabe-se que será adquirida uma certa quantidade de Pontos de Função, baseado na lista de necessidades identificadas no Plano Diretor de Tecnologia da Informação (PDTI), na capacidade de gerenciamento e na disponibilidade de recursos financeiros da organização, mas a especificação dos requisitos de cada software será realizada posteriormente em cada ordem de serviço a ser aberta durante a execução do contrato, ou pela equipe interna. **Produtos requeridos:** Pla-p1 – Documento da análise e justificativa da necessidade da contratação Pla-p2 – Documento da análise de mercado e soluções de TI NOTA: Este documento é, em geral, elaborado como parte do processo de análise de viabilidade de iniciar o projeto de aquisição, avaliando-se o problema da organização e as alternativas possíveis de solução que o mercado oferece. **Responsável:** Área de TI **Participantes:** Área de Negócio **Produtos gerados:** Pla-p1 – Documento da análise e justificativa da necessidade da contratação revisado. NOTA: Esta tarefa pode provocar a revisão do escopo das necessidades a serem atendidas em função das características dos requisitos a serem contemplados para atender as necessidades, que podem causar impactos em custo e prazo. Pla-p3 – Documento de especificação de requisitos **Legislação relacionada:** Lei nº 8.666/1993, arts. 6º, IX, e 12 (Brasil, 1993); Decreto nº 3.555/2000, art. 8º, I, II e III, "a" (Brasil, 2000); Instrução Normativa SLTI/MP nº 04/2008, art. 10, III (Brasil, 2008b).

5 - Aplicabilidade dos Resultados

Relevância: A Instrução Normativa SLTI/MP nº 04/2008 (Brasil 2008b) e os acórdãos 1.603 (Brasil, 2008c) e 2.471/2008-TCU-Plenário (Brasil, 2008d) estabelecem a obrigatoriedade de os entes públicos adotarem processos formais de contratação de serviços de TI, em particular em processos de *software*.

O presente modelo de processo pode auxiliar os governantes e os gestores públicos de TI no atendimento dessa demanda legal, visto que o processo em si já é aderente aos requisitos da legislação e pode favorecer:

- à ampliação do valor da TI aos processos de negócio dos entes públicos;
- à ampliação do estímulo à adoção da melhoria da qualidade do software brasileiro, em linha com o PBQP-SW/MCT (Brasil, 2009);
- à ampliação da participação de pequenas e médias empresas na comercialização de software e serviços correlatos para Governo, em linha com a Lei Complementar 123/2006 (Brasil, 2006) e também com o PBQP-SW/MCT, uma vez que foram incluídos mecanismos de divisão de objeto, de exclusão de exigência desnecessárias e de aumento do uso do pregão eletrônico.

Impacto: Espera-se que o modelo seja discutido e aperfeiçoado no âmbito da Administração Pública e contribua:

- com a Softex e com a SLTI/MP na produção e aperfeiçoamento de modelos e normas sobre o assunto;
- com os entes públicos na capacitação de gestores de contratos de TI e na facilitação do atendimento da Instrução Normativa SLTI/MP nº 04/2008 (Brasil, 2008b), permitindo aumento da eficácia, eficiência, efetividade, economicidade, legalidade e legitimidade dos projetos de TI e aumentando o valor agregado pela TI ao negócio institucional, em termos de aumento de benefícios produzidos, de redução de riscos legais e contratuais e de aumento da integração entre as áreas de TI, de licitação, jurídica e os gestores de negócio; e
- com o MCT na política de melhoria da qualidade do software brasileiro (PBQP-SW/MCT) e na ampliação da participação de pequenas e médias empresas nas compras públicas (Lei Complementar nº 123/2006) (Brasil, 2006).

Abrangência: O modelo proposto foi concebido com base na legislação da Administração Pública Federal. Porém, como a legislação de licitações e contratos de estados e municípios também está vinculada à Lei nº 8.666/1993 (Brasil, 1993), considera-se que o modelo é perfeitamente aplicável a todas as esferas, sem

distinção, com pouca necessidade de adaptação.

Inovação: A inovação do presente modelo reside na integração de aspectos técnicos da Engenharia de Software com os aspectos legais em um só modelo de processo. Desse modo, abre-se à oportunidade de interação mais produtiva entre os gestores públicos de TI e as áreas jurídica, administrativa e de controle (interno ou externo), com base em contextos mais compartilhados e uma linguagem comum. Com isso, torna-se possível à redução do retrabalho, pois os documentos de contratação podem ser produzidos já aderentes à legislação da primeira vez, a redução do tempo de contratação, mesmo utilizando-se dos rigorosos procedimentos licitatórios, e o aumento da eficiência, seja pela redução de prazos de contratação ou pela redução dos valores pagos pela Administração em suas contratações em virtude do aumento da concorrência.

6 - Conclusão e perspectivas futuras

No ciclo 2009, foi possível construir um modelo de processo de contratação de software e serviços correlatos com base no marco legal do setor público e no Guia de Aquisição do MPS.BR. Esse modelo alia aspectos técnicos e jurídicos em um único modelo de processo. Por essa razão, o modelo tem potencial para reduzir as divergências entre as áreas envolvidas nas contratações (TI, jurídica, administrativa, orçamentária e controle), acelerando o processo sem perda de segurança jurídica e reduzindo preços em razão do aumento da competição.

Em novo ciclo será possível:

- definir modelos de documento que implementem o presente modelo de processo;
- realizar testes de campo, com medição da melhoria alcançada;
- validar o modelo de processo e de documentos por meio de estudo de caso;
- publicar os resultados da pesquisa;
- encaminhar sugestão de análise do presente modelo à Softex e à SLTI/MP;
- incorporar o conhecimento produzido nesta pesquisa ao projeto pedagógico do curso de graduação Engenharia de Software da UNB-Gama.

7 - Referências bibliográficas

ANDRADE, Edméia Leonor Pereira de; SCALET, Danilo. **RELATÓRIO TÉCNICO: Personalização do Processo de Aquisição do MPS.BR para o Ministério da Agricultura, Pecuária e Abastecimento - MAPA.** Disponível em: <http://www.softex.br/portal/softexweb/uploadDocuments/_mpsbr/RELATORIO_TECNICO_Consultor_edmeia_220107.pdf>. Acesso em: 26 fev. 2010.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - ABNT. **ABNT NBR ISO/IEC 12207:2009 - Engenharia de sistemas e software - Processos de ciclo de vida de software**Disponível em: <http://www.abntcatalogo.com.br/norma.aspx?ID=38643>. Acesso em: 26 fev. 2010.

BRASIL. **Lei n° 8.666, 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. 1993. Disponível em: <http://www.planalto.gov.br/ccivil_03/Leis/L8666cons.htm>. Acesso em: 26 fev. 2010.

_____. **Decreto n° 2.271, de 7 de julho de 1997.** Dispõe sobre a contratação de serviços pela Administração Pública Federal direta, autárquica e fundacional e dá outras providências. 1997. Disponível em: <http://www.planalto.gov.br/ccivil_03/decreto/D2271.htm>. Acesso em: 26 fev. 2010.

_____. **Decreto n° 3.555, de 8 de agosto de 2000.** Aprova o Regulamento para a modalidade de licitação denominada pregão, para aquisição de bens e serviços comuns. 2000. Disponível em: <http://www.planalto.gov.br/ccivil_03/decreto/D3555compilado.htm>. Acesso em: 26 fev. 2010.

_____. **Lei Complementar 123, de 14 de dezembro de 2006.** Institui o Estatuto Nacional da Microempresa e da Empresa de Pequeno Porte; altera dispositivos das Leis nos 8.212 e 8.213, ambas de 24 de julho de 1991, da Consolidação das Leis do Trabalho – CLT, aprovada pelo Decreto-Lei no 5.452, de 1° de maio de 1943, da Lei no 10.189, de 14 de fevereiro de 2001, da Lei Complementar no 63, de 11 de janeiro de 1990; e revoga as Leis nos 9.317, de 5 de dezembro de 1996, e 9.841, de 5 de outubro de 2006a. Disponível em: <http://www.planalto.gov.br/ccivil_03/Leis/LCP/Lcp123.htm>. Acesso em: 26 fev. 2010.

_____. Tribunal de Contas da União. **Acórdão 786/2006-TCU-Plenário.** 2006b. Disponível em: <http://contas.tcu.gov.br/portaltextual/MostraDocumento?lnk=(acordao+adj+786/2006+adj+plenario)[idtd][b001]>. Acesso em: 26 fev. 2010.

_____. _____. **Acórdão 1.934/2007-TCU-Plenário.** 2007. Disponível em: <http://contas.tcu.gov.br/portaltextual/MostraDocumento?lnk=(acordao+adj+1934/2007+adj+plenario)[idtd][b001]>. Acesso em: 26 fev. 2010.

_____. Ministério do Planejamento, Orçamento e Gestão. Secretaria de Logística e Tecnologia da Informação. **Instrução Normativa SLTI n° 2, de 30 de abril de 2008.** Dispõe sobre regras e diretrizes para a contratação de serviços, continuados ou não. 2008a. Disponível em: <http://www.governoeletronico.gov.br/biblioteca/arquivos/instrucao-normativa-no-02>. Acesso em: 26 fev. 2010.

_____. _____. **Instrução Normativa SLTI n° 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. 2008b. Disponível em: <http://www.governoeletronico.gov.br/biblioteca/arquivos/instrucao-normativa-no-04-2>. Acesso em: 26 fev. 2010.

_____. Tribunal de Contas da União. **Acórdão 1.603/2008-TCU-Plenário.** 2008c. Disponível em: <http://contas.tcu.gov.br/portaltextual/MostraDocumento?lnk=(acordao+adj+1603/2008+adj+plenario)[idtd][b001]>. Acesso em: 26 fev. 2010.

_____. _____. **Acórdão 2.471/2008-TCU-Plenário.** 2008d. Disponível em: <http://contas.tcu.gov.br/portaltextual/MostraDocumento?lnk=(acordao+adj+2471/2008+adj+plenario)[idtd][b001]>. Acesso em: 26 fev. 2010.

_____. Ministério da Ciência e Tecnologia. **Programa Brasileiro da Qualidade e Produtividade em Software. Estratégias e ações.** Brasília: MCT, 2009. Disponível em: <http://www.mct.gov.br/index.php/content/view/4794.html>. Acesso em: 26 fev. 2010.

CRUZ, Cláudio Silva da. **Governança de TI e conformidade legal no setor público: um quadro referencial normativo para a contratação de serviços de TI.** 2008. 252f. Dissertação (Mestrado em Gestão do Conhecimento e da Tecnologia da Informação). Universidade Católica de Brasília, Brasília, 2008. Disponível em: <http://www.bdtd.ucb.br/tede/tde_arquivos/3/TDE-2008-11-25T123713Z-687/Publico/Texto Completo Cruz - 2008.pdf>. Acesso em: 26 fev. 2010.

_____. **Contratação de serviços de TI.** Curso integrante do MBA em Governança de TI. Brasília: Fundação Universa/Universidade Católica de Brasília, 2009a.

_____. **Elaboração do Plano Diretor de Tecnologia da Informação - PDTI.** Curso integrante do programa de desenvolvimento de gestores de TI da Secretaria de Logística e Tecnologia da Informação/MP. Brasília: ENAP, 2009b.

_____. **Gestão de contratos de serviços de TI - GCTI.** Curso integrante do programa de desenvolvimento de gestores de TI da Secretaria de Logística e Tecnologia da Informação/MP. Brasília: ENAP, 2009c.

_____. **Planejamento de contratação de serviços de TI - PCTI.** Curso integrante do programa de desenvolvimento de gestores de TI da Secretaria de Logística e Tecnologia da Informação/MP. Brasília: ENAP, 2009d.

GILLIS, Eric Robert; ANDRADE, Edméia Leonor Pereira de. 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-SP, 19 a 22 de outubro de 2009, p. 92-101. Disponível em: <http://www.softex.br/portal/softexweb/uploadDocuments/Softex%20WAMPS%202009%20Web.pdf>. Acesso em: 26 fev. 2010.

INFORMATION TECHNOLOGY GOVERNANCE INSTITUTE - ITGI. **COBIT - Control Objectives for Information and related Technology.** 4.1. ed. Rolling Meadows: ITGI, 2007. Disponível em: <http://www.isaca.org/AMTemplate.cfm?Section=Downloads&Template=/MembersOnly.cfm&ContentFileID=14002>. Acesso em: 26 fev. 2010.

SCALET, Danilo; ANDRADE, Edméia Leonor Pereira de. O Guia de Aquisição do MPS.BR e o Pregão Eletrônico. IN: **Workshop Anual do MPS**, 5. WAMPS 2009. Campinas-SP, 19 a 22 de outubro de 2009, p. 82-91. Disponível em: <http://www.softex.br/portal/softexweb/uploadDocuments/Softex%20WAMPS%202009%20Web.pdf>. Acesso em: 26 fev. 2010.

SOFTEX, Sociedade para Promoção da Excelência do Software Brasileiro. **MPS.BR - Melhoria de Processo do Software Brasileiro – Guia de Aquisição.** São Paulo: SOFTEX, Junho 2009. Disponível em: <http://www.softex.br/mpsbr/_guias/guias/MPS.BR_Guia_de_Aquisicao_2009.pdf>. Acesso em: 26 fev. 2010.

SOUSA, Marcelo Paiva de; SÂNDI, Vanessa Teixeira de Oliveira; OLIVEIRA, Káthia Marçal de; FIGUEIREDO, Rejane Maria da Costa. Processo de aquisição de produtos e serviços de software para uma instituição bancária. In: **SIMPROS - Simpósio Internacional de Melhoria de Processos de Software**, 7., 2005. São Paulo, SP – Brasil, 21-23 nov. 2005. Disponível em: <http://www.simpros.com.br/upload/A08_2_artigo14644.pdf>. Acesso em: 26 fev. 2010.

PCP-DEF - Método para Definição de Perfis Alvo para Melhoria de Processo em MPEs de Software

Cristiano Schwening1,2, Marcello Thiry2

1ENGSOFT Soluções em Melhoria de Processos
Rua 15 de novembro, 321/504, Ijuí – RS - Brasil
2Universidade do Vale do Itajaí (UNIVALI) – Campus São José
Mestrado em Computação Aplicada (MCA)
Rodovia SC 407, km 04, São José – SC - Brasil

cristiano@engsoft.com.br, marcello.thiry@gmail.com

Resumo*. Este trabalho apresenta um método para a definição de perfis de capacidade de processos alinhados com características específicas (modelo de negócio, estágio de crescimento e disciplina de valor) de cada micro e pequena empresa de software (MPE). O método é composto por um processo, artefatos e guias que auxiliam na definição de perfis de processos. Assim, a aplicação do método permite selecionar um conjunto de processos alinhados ao negócio oferecendo ações de melhoria mais objetivas para a MPE.*

Abstract*. This paper presents a method for the definition of process capability profiles lined up with specific characteristics (business model, business life cycle and value discipline) of each small software company. The method is made up of a process, artifacts and guides that assist to obtain process profiles. Therefore, the application of the method allows selecting a set of adjusted processes and lined up with the business model offering more objective improvement actions to a small software company*

1 Introdução

As mudanças que estão ocorrendo nos ambientes de negócios têm motivado as organizações a modificar suas estruturas organizacionais e processos produtivos, saindo da visão baseada em áreas funcionais em direção a redes de processos centrados no cliente [Softex 2007]. Portanto, projetos que objetivam realizar diversas ações voltadas para modificar aqueles processos prioritários para a organização, com o propósito de realizar uma melhoria de processo de software (MPS), têm sido tratados com grande destaque nos últimos anos por estas empresas [Oktaba e Piattini 2008].

Entretanto, micro e pequenas empresas (MPEs) do setor de software (força de trabalho inferior a 50 pessoas) demandam uma melhor efetividade destas iniciativas de MPS e ainda por um maior alinhamento com seus objetivos de negócio [Salviano 2006]. Realizar uma iniciativa de MPS requer esforço, conhecimento em engenharia de software, investimento financeiro direto e indireto, alto apoio da alta direção, adaptação correta das práticas solicitadas e a disponibilidade de alocação de recursos humanos para a implementação. Estas características nem sempre estão presentes nas MPEs de software [Anacleto *et al* 2004]. Desta forma, uma vez que as MPEs representam 85% das empresas de software no Brasil [MCT 2005], faz sentido buscar alternativas para apoiar programas de MPS em MPEs.

Este projeto foi baseado no estudo de três características observáveis em uma MPE de software: seu estágio de crescimento, seu modelo de negócio e a sua principal disciplina de valor. Ou seja, as características específicas das MPEs devem fazem com que as ações destinadas à melhoria de processo sejam aplicadas de uma forma especial para elas e visivelmente diferente de como isso é realizado nas grandes organizações [Salviano 2006].

Os estágios de crescimento definem conjuntos de características e práticas pelas quais as organizações devem evoluir para aumentarem o seu nível de maturidade [Daft 2003]. O modelo de negócio expressa a visão estratégica do negócio, sendo este um ponto relevante quando inicia-se uma iniciativa de MPS [Rajala e Tuunainen 2003]. A disciplina de valor fornece orientações relacionadas ao tipo de estratégia pretendida pela organização para ser competitiva e gerar valor para os clientes [Treacy e Wiersema, 1993].

Nas últimas décadas foram desenvolvidos diversos modelos e normas com o propósito de auxiliar as organizações envolvidas em melhoria dos processos de software. Uma relação parcial dos modelos mais relevantes inclui: CMMI-DEV [Sei 2006], NBR ISO/IEC 15504 [NBR ISO 2008], MR-MPS [Softex 2007] e COMPETISOFT [Oktaba e Piattini 2008].

Porém, estes modelos não procuram fornecer um alinhamento com as características da organização, deixando esta tarefa a cargo daqueles que executarão a melhoria na organização [Salviano 2006]. Com o propósito de reduzir a deficiência que os atuais modelos apresentam em relação ao alinhamento com as características de uma MPE foram desenvolvidas abordagens para MPS. Alguns exemplos de abordagens que trabalham com perfis de processo são: MARES [Anacleto *et al* 2004], PRO2PI [Salviano 2006], CMMI *in Small Settings Toolkit* [Garcia 2006], FAME [Beitz e Wieczorek 2000] e COMPETISOFT [Oktaba e Piattini 2008].

Neste contexto, este trabalho apresenta um método para apoiar a definição de perfis de capacidade de processo1 (PCP), proporcionando às abordagens para MPS uma sistemática objetiva de seleção de processos através de critérios claros e adequados para uma MPE de software.

[5]2 Objetivos e Justificativas

O objetivo principal foi o desenvolvimento de um método que apoie uma melhoria de processos de software através da definição de perfis, incluindo processos e respectivos níveis de capacidade, a partir das características observáveis nos modelos de negócios, estágios de crescimento e disciplinas de valor das micro e pequenas empresas de software.

A busca pela resolução das problemáticas relacionadas com a efetividade dos projetos para MPEs de software foi o principal foco deste projeto. Ao analisar estes problemas sob as perspectivas financeiras, de negócio, de estágio de crescimento e da disciplina de valor constata-se, ainda mais, a necessidade do desenvolvimento deste método (BEITZ et al., 1999). Redução de custos em MPEs torna-se relevante por impactar na sua capacidade de promover melhorias organizacionais com menor custo financeiro agregado ao projeto para MPS

Em relação à perspectiva de negócio, o método pode definir critérios objetivos que permitam uma análise e classificação da MPE de software de acordo com modelo de negócio adotado, fornecendo assim subsídios para a escolha dos processos que permitam que a organização atinja os objetivos de negócio estabelecidos na sua estratégia e em consequência agregar ainda mais valor para os produtos e/ou serviços.

3 Metodologia para desenvolvimento do método

Este método foi construído através da execução de atividades sistemáticas de pesquisa e da aplicação em organizações piloto. Nas atividades de pesquisa foram selecionadas e analisadas informações provenientes de outros trabalhos relacionados e também de informações provenientes do conhecimento do pesquisador e da equipe do LQPS (Laboratório de Qualidade e Produtividade em Software da Univali). Desta forma, para o desenvolvimento do PCP-DEF (Perfil de Capacidade de Processo – Método para Definição) foram planejadas e realizadas as seguintes atividades:

1. Identificar as principais características presentes em MPEs e selecionar as mais relevantes para o trabalho.

2. Mapear as práticas do CMMI-DEV [Sei 2006] de acordo as informações relacionadas com as características selecionadas.

3. Detalhar o método por meio de atividades e de passos necessários para a definição de um PCP.

4. Aplicar a versão inicial do PCP-DEF em MPEs de software.

5. Avaliar os resultados e ajustar o PCP-DEF.

6. Revisar, documentar o PCP-DEF e avaliá-lo novamente em MPEs de software.

O objetivo inicial foi identificar características aplicáveis em qualquer tipo de MPE de software. Dentre as características observadas, três foram selecionadas: estágio de crescimento, modelo de negócio e disciplina de valor. Esta seleção foi feita com base em atividades com o foco em identificar as variáveis mais importantes em qualquer MPE. Para alcançar este objetivo foi executada uma revisão da literatura e uma coleta de informações com especialistas em avaliação de processos.

Na etapa 2 as áreas de processos e as respectivas práticas dos níveis 2 e 3 do CMMI-DEV [Sei 2006] foram mapeadas com relação às três características. Para verificar a aplicabilidade, os perfis obtidos neste mapeamento foram comparados com outros perfis produzidos em 26 avaliações de MPEs do Rio Grande do Sul, Santa Catarina e Paraná utilizando o MARES [Anacleto *et al* 2004] e PRO2PI [Salviano 2006]. Estes perfis, provenientes do mapeamento, foram novamente ajustados utilizando para isso a experiência dos autores e de outros pesquisadores do LQPS da área de avaliação de processos.

Na etapa 3, por meio de pesquisa bibliográfica e das etapas anteriores foi desenvolvida uma versão piloto do PCP-DEF. Esta versão foi utilizada na etapa 4 objetivando obter indicadores para avaliar a aplicabilidade do método. Por fim, na etapa 5 foram realizados os ajustes detectados e posteriormente, na etapa 6, foi novamente aplicado o PCP-DEF em 6 MPEs com o objetivo de avaliá-lo com relação aos ajustes relatados.

O método requer inicialmente que sejam fornecidas as seguintes informações: o estágio de crescimento; o modelo de negócio adotado e a disciplina de valor. Estas informações estão diretamente relacionadas com a MPE e devem ser obtidas através de técnicas que possam detectar tais características. No caso do estágio de crescimento, a abordagem de MPS deve selecionar um estágio no qual a MPE encontra-se localizada no momento da definição do PCP. Para selecionar este estágio, pode ser utilizada a técnica de entrevistas com os

1 Um conjunto de processos que devem ser escolhidos para uma MPS e que causarão maior impacto às atividades de negócio da organização e em consequência auxiliarão na redução dos prazos de entrega e custos, melhorando assim a qualidade do produto entregue [Salviano 2006].

principais colaboradores e diretores da organização e em conjunto aplicar questionários dirigidos com perguntas que ajudem no levantamento da informação [Silva 2002]. Para selecionar o modelo de negócio podem ser aplicados questionários, para os diretores da organização, que contenha os principais modelos conhecidos na literatura [Schwening 2009]. Do mesmo modo, a disciplina de valor da organização pode ser selecionada através de questões direcionadas aos executivos que busquem identificar: quanto ao foco em inovação; quanto à política de preços; quanto ao relacionamento com o cliente; quanto a busca por novas fronteiras no produto; quanto ao reconhecimento da qualidade do produto pelo cliente; quanto ao tipo de produto entregue ao cliente; e quanto ao foco em cliente específicos [Kaplan e Norton 2004].

Mesmo não sendo o objetivo principal do PCP-DEF, são fornecidas técnicas objetivas para determinar as informações requeridas. O método disponibiliza listas de verificação com as informações mencionadas acima para ajudar na detecção das 3 características necessárias [Schwening 2009].

Com relação ao número de características a serem analisadas para obter um PCP foi necessário definir uma técnica para agrupar este conjunto de informações, principalmente os modelos de negócio. Os modelos de negócios podem se ramificar em diversos segmentos de atuação conforme as características definidas pelas organizações habitualmente quando da definição estratégica dos objetivos da organização.

Para permitir uma organização clara destas informações definiu-se um conceito para agrupar estas ideias, ou seja, classificá-las para facilitar a localização e uso. O conceito, adotado nesse artigo, para induzir o acesso às informações de forma a existir algum tipo de classificação foi o mapa mental [Buzan 1996].

4 Resultados obtidos

Dentre as possíveis contribuições que o método de definição de PCP apresentado neste projeto conseguiu proporcionar, e que estão relacionados com métodos desenvolvidos, destacam-se:

Definição objetiva de características (estágio de crescimento, disciplina de valor e modelo de negócio) que podem ser detectadas em qualquer tipo e porte de organização;

Definição de regras iniciais para realizar a seleção de perfis de capacidade de processos;

Mapeamento, mesmo que parcial, dos principais modelos de negócios das empresas de software brasileiras;

Um conjunto de ferramentas que permitam a utilização do método em qualquer abordagem para MPS.

A utilização de três características (estágio de crescimento, disciplina de valor e modelo de negócio) que permitem a analise mais rápida de uma organização.

O método desenvolvido durante o projeto deve, preferencialmente, ser utilizado em parceria com uma abordagem para MPS (ex.: MARES, PRO2PI). O foco está em fornecer um PCP inicial, orientado para os objetivos de negócio da organização que poderá ser depois adaptado pela abordagem.

Desta forma, com base no estágio de crescimento, no modelo de negócio e na disciplina de valor da MPE, o método orienta a identificação do PCP para a empresa analisada. O perfil gerado identifica as práticas que a organização deve atingir para melhorar seus processos, permitindo assim uma melhoria diretamente alinhada ao seu negócio.

A Figura 1 descreve as possíveis inter-relações entre as principais etapas e partes envolvidas na execução do método.

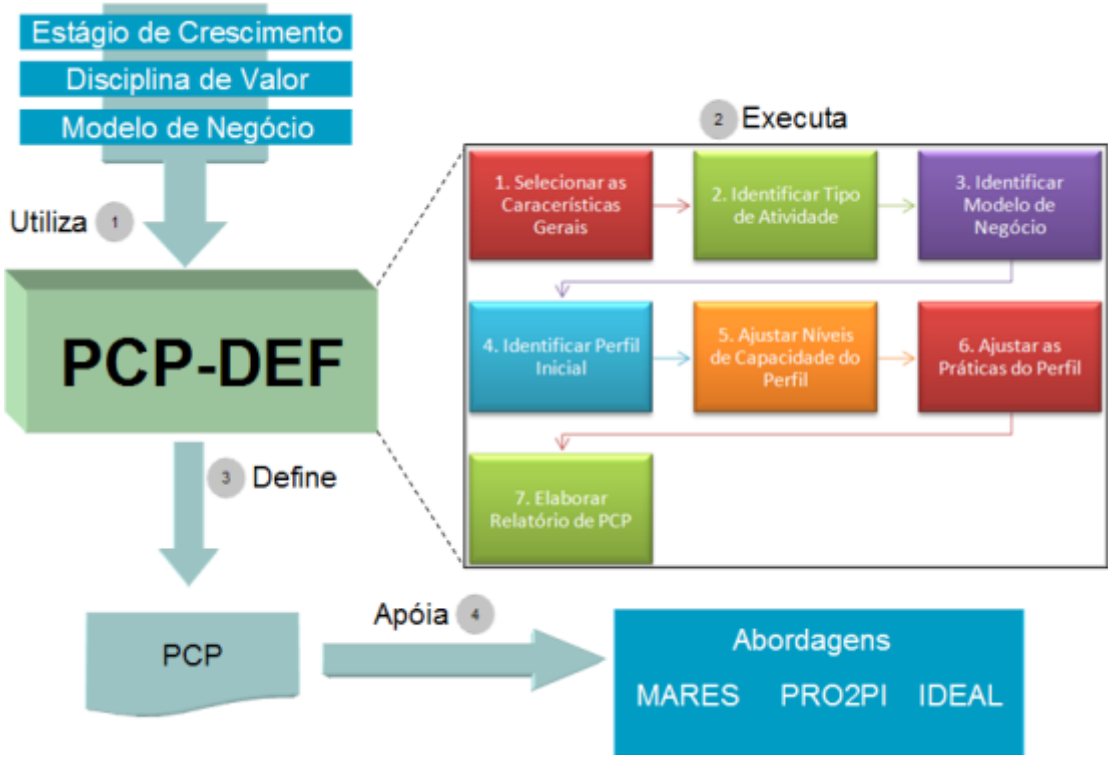


Figura 1. Principais interações do PCP-DEF

O método inicia (1) com a identificação do estágio de crescimento, o modelo de negócio e a disciplina de valor da MPE. Inicia-se então, a sequência de atividades definidas pelo método (2). O mapa mental é percorrido, iniciando-se com o estágio de crescimento e com a disciplina de valor; o próximo passo é percorrer o mapa mental a partir do modelo de negócio. Depois, navega-se nas ramificações existentes até que se chegue aos ramos finais; por fim é possível visualizar o perfil predefinido que será utilizado como perfil base para a definição do PCP final (3). Em primeiro lugar, no estágio de crescimento são analisados os fatores que definem a maturidade da organização, em seguida a disciplina de valor direciona para qual é o foco estratégico com relação aos clientes e por fim o modelo de negócio define a visão estratégia da organização.

O método ainda oferece a possibilidade de revisão do PCP inicial a partir de regras de adaptação baseadas em objetivos de negócio. De acordo com os objetivos identificados para uma MPE, as regras podem ser executadas e o PCP inicial é ajustado. Um exemplo de regra de adaptação é “Se a empresa desenvolve MOST verificar a relevância de definir a rastreabilidade do projeto”.

Na etapa final (4), um PCP é apresentado para apoiar a abordagem na avaliação dos processos da MPE. Este PCP pode ser utilizado integralmente como base para um projeto de MPS ou ainda sofrer outros ajustes conforme critérios da abordagem de MPS.

Os resultados gerados durante o projeto proporcionaram também a geração de três artigos que foram publicados ou aceitos para publicação, conforme abaixo:

SCHWENING, CRISTIANO.; THIRY, M., Definindo perfis de capacidade de processos de software a partir do modelo de negócio de micro e pequenas empresas. Anais do I Simpósio de Computação Aplicada (SCA 2009). Passo Fundo : UPF Editora, 2009. v. 001. p. 1-15.

SCHWENING, CRISTIANO. ; THIRY, M., Um método para definir perfis de capacidade de processo a partir de características de micro e pequenas empresas de software. Proceedings of International Workshop on Advanced Software Engineering (IWASE). Santiago do Chile : Editorial Universidad de Santiago de Chile, 2009. v. 1. p. 19-26.

SCHWENING, CRISTIANO. ; THIRY, M., Um método para definir perfis de capacidade de processo a partir de características de micro e pequenas empresas de software. CLEI Electronic Journal, 2010 (artigo aceito para publicação).

O projeto também serviu de base para a conclusão de uma dissertação de mestrado em Computação Aplicada realizada na Universidade do Vale do Itajaí/SC.

5 Aplicabilidade dos resultados

Os resultados deste projeto podem ser utilizados pelas abordagens para MPS. Estas abordagens possuem técnicas que ajudam na seleção de processos, porém elas carecem de um método mais sistemático que possa

automatizar a análise de questões relacionadas com características particulares das organizações e identificar de forma mais ágil e objetiva os perfis de capacidade de processos.

Neste sentido, o PCP-DEF buscou ser aplicável como apoio às carências detectadas nas abordagens para MPS. Uma das contribuições deste método será a oferta de mais uma opção de ferramenta de apoio para a definição dos processos, agilizando e tornando mais objetiva a seleção dos processos relevantes para um projeto de MPS.

6 Características inovadoras

A principal contribuição inovadora deste projeto foi o desenvolvimento de um componente adicional destinado às abordagens para MPS, com o propósito de agilizar a obtenção de perfis de capacidade de processos ajustados as características das organizações, através da análise de 3 características (estágio de crescimento, disciplina de valor e modelo de negócio) organizacionais.

7 Conclusões e Perspectivas Futuras

Em virtude das características singulares das micro e pequenas empresas de software os processos de desenvolvimento e manutenção de software, não podem estar unicamente focados em níveis de maturidade. Estas organizações precisam considerar suas características aos estabelecer iniciativas de melhoria de processos de software alinhadas com o seu negócio. Para muitas MPEs os processos de software devem estar diretamente alinhados ao modelo de negócio escolhido, sob o risco desta falta de alinhamento afetar diretamente as estratégias comerciais e financeiras destas organizações. O alinhamento dos processos pode ser considerado como uma necessidade estratégica para que todas as empresas de software, e principalmente as MPEs, manterem-se competitivas.

Este projeto apresentou um método para a definição de perfis de capacidade de processos para posterior utilização em projetos para MPS em MPEs. O objetivo é contribuir para as organizações fornecendo uma sistemática para apoiar na definição de um PCP alinhado com modelo de negócio. O método é detalhado através de uma representação de processo (atividades, passos e responsáveis), artefatos e guias que auxiliam na definição de perfis de processos focados nos processos que estejam mais alinhados ao modelo de negócio de uma organização. No transcorrer do projeto buscou-se demonstrar a importância da adequada seleção de processos para as empresas, principalmente as MPEs.

O conhecimento produzido a partir deste projeto representa o ponto de partida para a concepção de um método para definição de perfis de capacidade de processos baseado em características observáveis nas organizações. Torna-se necessário aprofundar o tema, desta forma, sugerem-se possíveis trabalhos futuros motivados no desenvolvimento do PCP-DEF.

Realizar o estudo e a definição de perfis de capacidade de processo para os níveis 4 e 5 do CMMI-DEV, B e A do MR MPS e respectivos processos da NBR ISO/IEC 15504-5, permitindo desta forma que o método possa, também, ser aplicado em organizações que estejam buscando estes níveis de capacidade.

Do ponto de vista dos artefatos desenvolvidos, existe a possibilidade de realizar a implementação de uma ferramenta que integre todos os artefatos e transformar as atividades do método em um workflow na própria ferramenta. Os *checklists* que foram construídos para a definição do estágio de crescimento e da disciplina de valor precisariam ser reescritos para conter uma linguagem mais relacionada com as empresas de software.

O método desenvolvido neste projeto ainda poderá ser complementado com a inclusão de todos os processos da norma ISO/IEC 15504-5 que permitem apoiar um maior número de práticas necessárias para o desenvolvimento e manutenção de software.

8 Referências Bibliográficas

Anacleto, A., Wangenheim, C. G. von e Salviano , C. F., “Avaliação de Processos para Início de Programas de Melhoria em Micro e Pequenas Empresas de Software”. VI Simpósio Internacional de Melhoria de Processos de Software, São Paulo, 2004.

BEITZ, A., WIECZOREK, I. Applying benchmarking to learn from best practices. Proceedings II International Conference (PROFES 2000), Ed. Springer, Oulu, Finlandia, Junho/2000.

Buzan, T., “The Mind Map Book”, Ed. Penguin Books, Londres, 1996.

Daft, R. L., “Organizações: teorias e projetos”, Ed. Pioneira: São Paulo, 2003.

Garcia, S., Graettinger, C. e Kost, K., “Proceedings of the First International Research Workshop for Process Improvement in Small Settings, 2006”. Special Report CMU/SEI-2006-SR-001, Software Engineering Institute, 2006.

MCT - Ministério da Ciência e Tecnologia. “Qualidade do setor de Software Brasileiro 2005”, Publicação eletrônica [mensagem pessoal], Mensagem recebida por <crsch@engsoft.com.br> em 18 mar. 2008.

NBR ISO/IEC 15504-5:2008, Tecnologia da informação - Avaliação de processo Parte 5: Um exemplo de Modelo de Avaliação de Processo. Rio de Janeiro, ABNT - Associação Brasileira de Normas Técnicas, 2008.

Oktaba, H.e Piattini, M. “Software process improvement for small and medium enterprises : techniques and case studies”. Information Science Reference, Hershey, New York, 2008.

Rajala, R., Rossi, M., Tuunainen, V.K. “A Framework for Analyzing Software Business Models”, 2003.

Salviano, C.F. “Uma proposta orientada a perfis de capacidade de processo para evolução da melhoria de processo de software”. Tese de doutorado pela Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação. 2006.

Schwening, C. Método para definição de perfis de capacidade de processo de software em MPEs. Dissertação de Mestrado. Universidade do Vale do Itajaí, São José, 2009.

SEI - Software Engineering Institute. “CMMI for Development, Version 1.2”, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, 2006. Disponível em: http://www.sei.cmu.edu, acessado em 12/12/09.

SILVA, J. da. Opção estratégica, aptidões dominantes e estágio no ciclo de vida das organizações: estudo de caso exploratório em uma média empresa industrial. Dissertação de Mestrado. Universidade Federal de Santa Catarina, Florianópolis, 2002.

Softex - Associação para Promoção da Excelência do Software Brasileiro. “MPS.BR – Guia Geral, v 1.2”, 2007. Disponível em: www.softex.br, acessado em: 12/12/09.

Thiry, M., Wangenheim , C. G. von, Zoucas, A. “Aplicando avaliações de contextualização em processos de software alinhados ao nível F do MR-MPS v1.2”. ProQuality (UFLA), v. 4, p. 27-33, 2008.

Treacy, M. e Wiersema, F., Customer Intimacy and Other Value Disciplines, Harvard Business Review, Jan/Fev, pp. 84-93, 1993.

Um Modelo de Capacidade para Desenvolvimento Distribuído de Software em Ambientes de Internal Offshoring

Rafael Prikladnicki

Faculdade de Informática – PUCRS – POA – RS – Brasil

rafaelp@pucrs.br

Resumo. *Neste artigo apresenta-se o relatório final do projeto 2.12, cujo objetivo foi desenvolver um modelo de capacidade (modelo WAVE) visando documentar os padrões de evolução das práticas das organizações envolvidas com o internal offshoring, um tipo específico de DDS. O modelo proposto como resultado do projeto é útil para aquelas empresas iniciando operações de DDS com captive centers, que poderão se beneficiar do conhecimento de práticas que foram aplicadas em outras organizações no passado.*

1. Introdução

A necessidade de obter vantagens competitivas no Desenvolvimento Distribuído de Software (DDS) tem levado às organizações a buscarem soluções externas em outros países, gerando o DDS em escala global – *offshore sourcing* ou *offshoring* (Carmel & Tjia, 2005). Neste caso, os dois principais modelos de atuação são o *offshore outsourcing* (contratação de uma organização externa, localizada em outro país) e o *internal offshoring* (criação de uma subsidiária – conhecida também como *captive center* – da própria organização, localizada em outro país).

Apesar de o DDS ser uma área recente, o seu próprio amadurecimento passa pelo entendimento de como suas práticas tem evoluído ao longo do tempo. Mas boa parte da literatura existente sobre a evolução das práticas de DDS em escala global se concentra em aspectos estratégicos, tais como decisões de se estabelecer centros de DDS (Carmel & Agarwal, 2002) e as relações entre cliente e fornecedor (Mirani, 2007). Além disso, a maioria dos estudos tem se concentrado no modelo de *offshore outsourcing* (Prikladnicki et al 2008), e na perspectiva das contratantes (Smite et al, 2008). Apesar disto, é crescente o número de organizações que tem optado pelo *internal offshoring* como estratégia de DDS. Como exemplo, das 900 organizações que são membros da NASSCOM (Associação Nacional das Empresas de Software) na Índia, mais de 300 delas está atuando no modelo de *internal offshoring*, com *captive centers* instalados naquele país (Ramamani, 2006). Além disso, um estudo recente sobre o desenvolvimento *offshore* no Brasil apontou que os *captive centers* representam 36% do mercado brasileiro de serviços *offshore* (Peres, 2009). Mas além de existir um crescimento na estratégia de *internal offshoring*, as organizações têm enfrentado diversos desafios do ponto de vista técnico. Assim, alguns estudos destacam a importância de se pensar na evolução do DDS de forma unificada, incluindo aspectos técnicos, não-técnicos e organizacionais (Meyer, 2006; Ramasubbu et al, 2005) Dentro deste contexto, este projeto visou desenvolvimento um modelo de capacidade para DDS com *captive centers*, proposto a partir de um extenso estudo da prática de DDS. O projeto foi desenvolvido ao longo de quatro anos, como parte da tese de doutorado do autor deste artigo.

2. Objetivos e Justificativa

Se o *offshore outsourcing* é bastante conhecido e adotado em muitas empresas para cortar custos operacionais e ganhar vantagens competitivas (Carmel & Tjia, 2005), o *internal offshoring* tem surgido como uma opção para a operacionalização do DDS. O uso de centros de desenvolvimento de software da própria empresa, localizados em países de baixo custo tem aumentado significativamente. Desta forma, as empresas conseguem manter suas operações "*in-house*".

Ramamani (2006) diz que o modelo de *internal offshoring* é um modelo de integração vertical, onde a operação é "in–house" e não envolve dependência em contratos complexos com agente externos. A lógica por trás de ter uma subsidiária *offshore* é justamente a integração vertical (onde é desejável manter os direitos em casa, ao invés de confiar em contratos que aumenta a complexidade da relação de sourcing).

Entretanto, a integração das operações não garante um sucesso automático para as empresas. As capacidades que uma subsidiária desenvolve ao longo do tempo e como estas capacidades agregam valor para a matriz é que governa a efetividade de uma relação de sourcing.

De acordo com Herbsleb (2007), o processo utilizado em uma relação de *offshore outsourcing* pode ser diferente daqueles utilizados numa relação de *internal offshoring*, e a caracterização neste caso pode fazer diferença para a prática de DDS.

Além disso, a pesquisa conduzida em um modelo de DDS não é automaticamente válida para todos os tipos de DDS. E a pesquisa atual em DDS não tem explorado o modelo de *internal offshoring* e seus possíveis desafios nas atividades de ES de forma suficiente na literatura (Madlberger & Roztocki, 2008) .

Sendo assim, este projeto teve como objetivo principal buscar respostas para uma crescente demanda na área de DDS, que são os centros de desenvolvimento de software do tipo *captive centers.* Com isso, foi proposto um modelo de capacidade documentando práticas de DDS em ambientes de *internal offshoring* (ou com *captive centers*), visando ajudar organizações inseridas neste contexto.

3. Metodologia de Execução

Este projeto teve início no ano de 2005. O desenvolvimento do modelo de capacidade envolveu diversos colaboradores e parceiros durante este período, quais sejam:

- aluno de mestrado em Ciência da Computação da PUCRS entre 2005 e 2006;

- aluno de doutorado em Ciência da Computação da PUCRS entre 2005 e 2009;

- pesquisador localizado nos Estados Unidos, entre 2005 e 2007;

- pesquisador localizado no Canadá, entre 2006 e 2009;

- o grupo de pesquisa em DDS da PUCRS (MuNDDoS), durante todo o período;

- empresas colaboradoras e interessadas no resultado do estudo, durante todo o período;

Além disso, diversas atividades foram realizadas, entre as quais se destacam:

- revisão inicial da literatura da área, em 2005;

- condução de um estudo de caso exploratório com duas empresas no Canadá e três no Brasil, em 2006;

- revisão sistemática da literatura da área, em 2007;

- condução de um estudo de caso com duas empresas no Brasil e uma na Índia, em 2007;

- publicação do primeiro livro em português na área de DDS, em 2007;

- análise de dados e proposta do modelo de capacidade, em 2008 e 2009;

- publicação de diversos artigos em eventos e periódicos nacionais e internacionais, durante todo o período.

Para garantir a riqueza das informações e manter a visão geral de todo o projeto, apresenta-se neste relatório todas as atividades desenvolvidas, destacando quais foram especificamente planejadas e executadas no ciclo de 2009. O projeto foi dividido em duas etapas (Figura 1), quais sejam:

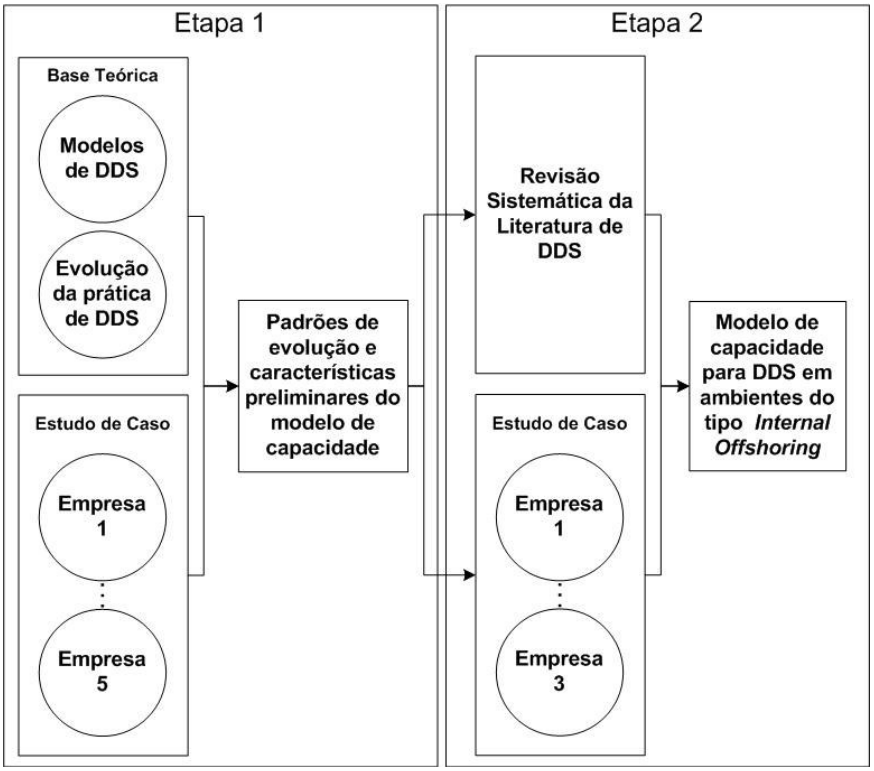


Figura 1. Desenho de pesquisa

Etapa 1: incluiu a definição da idéia, o problema de pesquisa, bem como sua estratégia, o próprio desenho de pesquisa, além de uma revisão inicial da literatura e o desenvolvimento de um estudo de caso em cinco subsidiárias de empresas atuando em DDS (Oates, 2006; Yin, 2003). Esta etapa foi importante na medida em que formou um referencial teórico consistente para a continuidade do estudo. Além disso, a revisão da base teórica permitiu um estudo do conhecimento existente na área (estado da arte). O objetivo do estudo de caso desenvolvido nesta etapa foi identificar atributos críticos para organizações que atuam em ambientes de DDS, no modelo de *internal offshoring*, sob os pontos de vista técnico (tais como processo de desenvolvimento software, gestão de projetos), não-técnico (tais como gestão de pessoas e habilidades interpessoais) e organizacional (tais como estrutura e modelo de gestão de organizações de DDS). Como resultado desta fase, identificou-se um conjunto de atributos candidatos para fazer parte do modelo de capacidade para *internal offshoring*, incluindo suas possíveis seqüências (padrões) de evolução.

Etapa 2: envolveu uma revisão sistemática da literatura de DDS e um segundo estudo de caso. Uma vez que os atributos de capacidade foram definidos, na segunda etapa planejou-se a execução de uma revisão sistemática da literatura de DDS (Dias Neto et al, 2007) e um estudo de caso múltiplo. O objetivo da revisão sistemática foi de aprofundar o entendimento nos modelos de evolução da prática de DDS existentes, de forma a contribuir para o desenvolvimento do modelo de capacidade proposto. A revisão sistemática foi incluída visando ampliar a cobertura da revisão de literatura inicial, que foi realizada em 2005. Além disso, o rápido desenvolvimento da área de DDS é razão suficiente para buscar novos estudos relacionados sendo desenvolvidos, e confirmar que estudos em *internal offshoring* têm sido pouco explorados na literatura, apesar do crescimento da sua prática nas empresas (Ramamani, 2006). Em paralelo, planejou-se o desenvolvimento de um estudo de caso múltiplo, com o objetivo de avaliar os atributos e padrões de evolução identificados na etapa inicial e exploratória da pesquisa, de forma a fornecer mais subsídios para o desenvolvimento do modelo de capacidade. O objetivo do estudo foi avaliar os atributos de capacidade identificados na primeira etapa da pesquisa. Como resultado, os padrões de evolução na prática de DDS em ambientes de *internal offshoring* foram identificados, apresentados no formato de um modelo de capacidade. A análise dos dados e a proposta do modelo de capacidade fizeram parte do projeto submetido ao ciclo 2009 do PBQP-Software.

4. Resultados Obtidos

O projeto produziu os seguintes resultados:

- **processos que contribuem para a melhoria da qualidade de software, descritos e institucionalizados:** foi desenvolvido o WAVE, modelo de capacidade para organizações inseridas em um contexto de Desenvolvimento Distribuído de Software em ambientes de *internal offshoring*. A figura 2 ilustra a estrutura do modelo proposto.

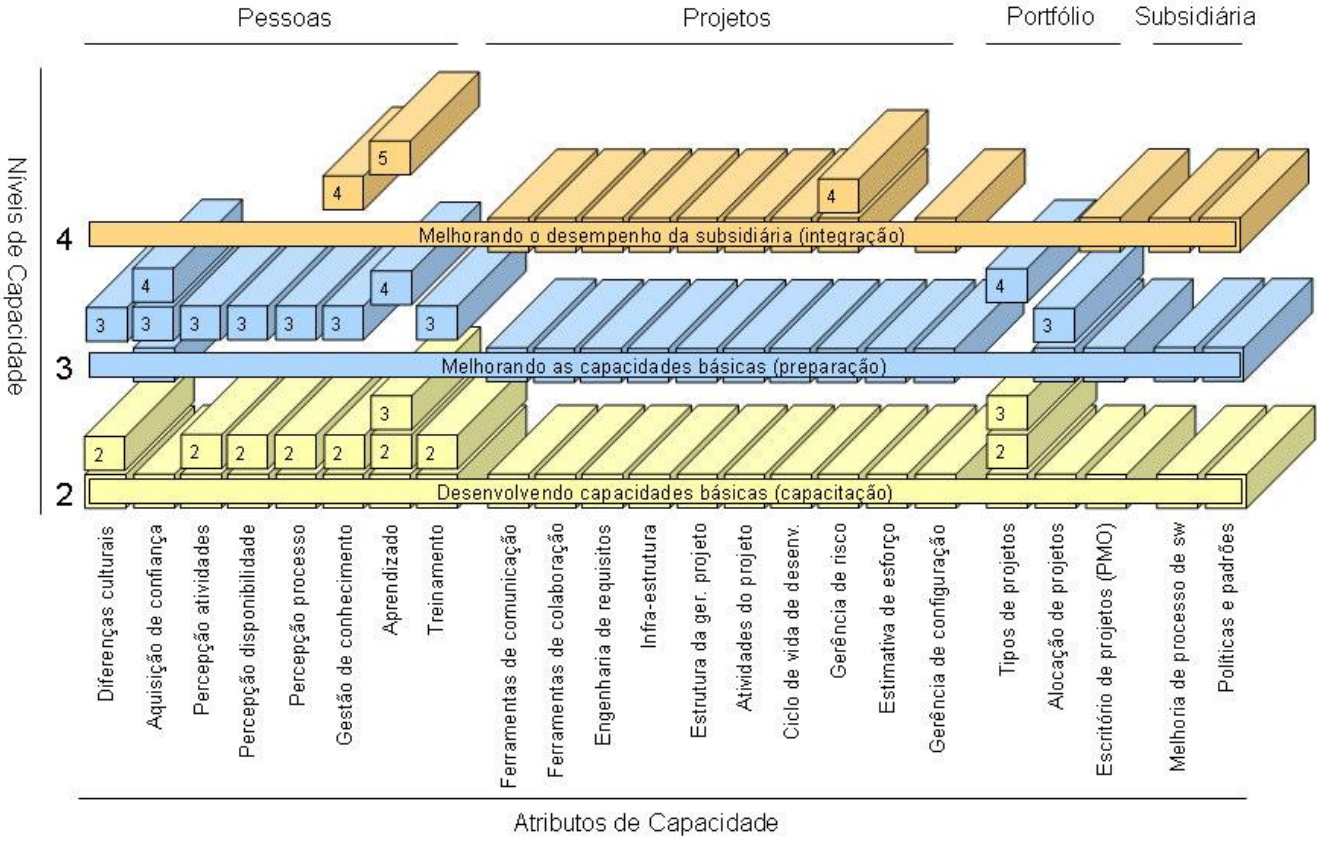


Figura 2. O modelo de capacidade proposto

- **artigos publicados:** este projeto gerou um total de 27 publicações, sendo um livro nacional - o primeiro livro em Língua Portuguesa sobre DDS, publicado pela Série livros didáticos Campus/SBC (Audy & Prikladnicki, 2007), um capítulo de livro internacional, dois periódicos internacionais, três periódicos nacionais, dez artigos publicados em conferências internacionais, dois em workshops de conferências internacionais, um em conferência nacional e oito em workshops de conferências nacionais. sobrediversos artigos ao longo da sua execução. O primeiro deles foi publicado em 2004, na AMCIS (*Americas Conference on Information Systems*) ainda com a proposta da pesquisa (Avritchir et al, 2004). Especificamente no ciclo 2009, foram publicados os seguintes artigos: um artigo na Infocomp *Journal of Computer Science* (periódico nacional), como um dos melhores artigos do Workshop Brasileiro de DDS (Prikladnicki & Audy, 2010); um artigo técnico no *Journal of* Software *Maintenance and Evolution: Research and Practice*, como um dos melhores artigos da *International Conference on Global Software Engineering* - ICGSE (Prikladnicki, 2010); um artigo na IEEE Software (Prikladnicki et al, 2010); um artigo na AMCIS (Prikladnicki & Audy, 2009a); um artigo no workshop CHASE (Prikladnicki, 2009b); um artigo no WDDS (Prikladnicki & Audy, 2009b); e um artigo na ICGSE (Prikladnicki, 2009a).

- **recursos humanos capacitados:** no escopo deste projeto, uma dissertação de mestrado e uma tese de doutorado foram concluídas (esta última como parte do ciclo 2009). Além disso, um trabalho de especialização foi finalizado no ano de 2009. Com isso, foi possível capacitar os alunos envolvidos em uma área nova dentro da Engenharia de Software em escala mundial.

- **dissertações e/ou teses geradas:** em 2006 foi gerado um volume de dissertação de mestrado no curso de Pós-Graduação em Ciência da Computação da PUCRS (Pilatti, 2006). No ciclo de 2009 foi gerado um trabalho de especialização em Gerência de Projetos com Ênfase em TI (DalSotto, 2009) e um volume de tese de doutorado (Prikladnicki, 2009c).

5. Aplicabilidade dos resultados

O modelo WAVE tem como objetivo principal ajudar empresas num melhor entendimento dos diversos passos e evolução necessários no modelo de negócio de *internal offshoring*, envolvendo decisões, processos, padrões, forma de trabalho e outros aspectos que outras organizações mais experientes já vivenciaram no passado. A avaliação da capacidade de cada subsidiária desempenha um papel importante para determinar níveis de evolução em cada empresa e eventualmente comparar com outras empresas inseridas no mesmo modelo de negócio de DDS. É importante ainda destacar que o modelo proposto não tem como objetivo avaliar a maturidade das empresas.

Do ponto de vista prático, na medida em que a estratégia da organização evolui, o modelo proposto pode ser combinado com práticas de modelos conhecidos na indústria, tal como o CMMI-SW e o MPS.BR. Além disso, entende-se que a adoção das práticas propostas no WAVE por parte de novas empresas inseridas no modelo de *internal offshoring* pode resultar em um menor impacto dos desafios encontrados por estas empresas em ambientes de DDS, ou então na melhoria da produtividade e da qualidade do desenvolvimento de software. Isto é reforçado por estudos recentes, identificados, que apresentam indícios nesta direção (Nguyen et al, 2008; Bird et al, 2009; Cataldo & Nambiar, 2009).

Estes estudos, combinados com a proposta apresentada para o modelo de capacidade para *internal offshoring*, indicam que o modelo WAVE pode contribuir de forma significativa para que, independente da distância existente, organizações inseridas no modelo de *internal offshoring*, com uma ou mais subsidiárias, podem ter sucesso no desenvolvimento dos seus projetos, evitando atrasos ou impactos na qualidade do produto final, conforme já documentado na literatura de DDS.

6. Características inovadoras

Até o presente momento, este é o primeiro modelo de que se tem conhecimento neste cenário, agregando contribuições para a teoria e prática do DDS. Sendo assim, a principal contribuição deste projeto é a proposta de padrões de evolução, organizados em um modelo de capacidade, para guiar uma ou mais subsidiárias em diversas atividades do ponto de vista da Engenharia de Software e integrando as perspectivas técnica, não-técnica e organizacional.

Este projeto contribuiu significativamente para aprofundar o estudo na área de DDS, com foco específico no modelo de *internal offshoring*. Foi possível aprofundar o entendimento nos principais modelos de negócio existentes no Desenvolvimento Distribuído de Software (*offshore outsourcing*, *internal offshoring*, *onshore insourcing* e *onshore outsourcing*), caracterizando o modelo de *internal offshoring* e suas diferenças. Como resultado, gerou-se um corpo de conhecimento contendo informações descritivas e comparativas contendo desafios associados a cada um dos quatro modelos estudados, com foco especial nos dois principais modelos na dimensão *offshore* (*offshore outsourcing* e *internal offshoring*).

Além disso, este projeto contribuiu no sentido de reforçar a importância de se identificar corretamente o modelo de negócio de DDS em que se está trabalhando. É comum encontrar pesquisas na área de DDS que não se preocupam em identificar corretamente a relação entre as equipes distribuídas e as empresas parceiras. Quando isto ocorre, existe um risco que deve ser levado em conta no que diz respeito à distorção dos

resultados, pois os desafios do DDS podem ser diferentes dependendo do modelo em que uma determinada empresa está inserida.

Do ponto de vista de contribuição teórica, a pesquisa proposta contribui para a área de Engenharia de Software no sentido de identificar as implicações técnicas, não técnicas e organizacionais da complexidade das atividades em ambientes de DDS. Além disso, contribui no sentido de avançar os estudos atualmente existentes e prover um modelo de capacidade baseado em estudos de base qualitativa. Também contribui para a criação de um corpo de conhecimento específico para esta área de DDS, incluindo uma taxonomia para projetos de desenvolvimento distribuído, um guia de práticas utilizadas nas empresas, modelos de maturidade e capacidade específicos para DDS existentes na literatura e o próprio modelo de capacidade proposta. Parte deste corpo de conhecimento fez parte do livro já destacado em seções anteriores (Audy & Prikladnicki, 2007).

Do ponto de vista prático, o modelo proposto visa ajudar as subsidiárias das organizações que estão iniciando operações globais no modelo de negócio de *internal offshoring*. Ainda, para empresas estabelecidas o estudo contribuiu fornecendo um modelo onde elas possam comparar as práticas utilizadas com os níveis de capacidade propostos e passos a serem seguidos.

Finalmente, ao longo deste projeto foi possível estabelecer uma interação com os principais grupos de pesquisa em DDS no Brasil e no exterior, além de participar da organização do primeiro evento internacional na área de DDS (ICGSE – *International Conference on Global Software Engineering*), realizado no Brasil em 2006, e do primeiro workshop nacional de DDS (WDDS), organizado junto ao Simpósio Brasileiro de Engenharia de Software em 2007.

7. Conclusões e Perspectivas Futuras

Neste artigo apresentou-se o relatório final do projeto submetido ao ciclo 2009 do PBQP Software, cujo objetivo foi desenvolver um modelo de capacidade para organizações inseridas em um contexto de DDS, em ambientes de *internal offshoring*. O modelo WAVE foi desenvolvido a partir de uma extensa pesquisa de base qualitativa, buscando respostas para o crescimento do *internal offshoring* (também conhecido como desenvolvimento de software com *captive centers*).

Em relação a perspectivas futuras, identifica-se grande potencial de crescimento nesta linha de pesquisa, onde os pontos fortes envolvem uma parceria estável entre a academia e a indústria, criando condições de experimentação e aprendizagem únicas, decorrentes de uma sinergia positiva entre os parceiros. Especificamente, entende-se que este projeto pode se desdobrar em alguns temas de interesse não apenas da comunidade científica, mas também de empresas e do governo. O modelo deve ser melhorado e seu uso incentivado para desenvolver iniciativas nacionais de melhoria de processo de DDS em empresas que adotam o *internal offshoring* como modelo de negócio. Além disso, entende-se que o modelo deva ser utilizado de forma contínua nas empresas, identificando como cada uma se comporta em relação às práticas propostas. Para isso, sugere-se a elaboração de um guia específico de avaliação das práticas do modelo, de forma a orientar as empresas que atuam no modelo de *internal offshoring* a avaliarem periodicamente sua capacidade.

8. Referências Bibliográficas

Audy, J. L. N., Prikladnicki, R. (2007). “Desenvolvimento Distribuído de Software: Desenvolvimento de Software com Equipes Distribuídas”, Série Campus-SBC, Editora Campus-Elsevier.

Avritchir, J., Prikladnicki, R., Evaristo, R., Audy, J. L. N. (2004). "A Maturity Model for Offshore Insourcing: A Research Proposal", Proc. AMCIS 2004 - Americas Conference on Information Systems, Nova York.

Bird, C., Nagappan, N., Devanbu, P., Gall, H., Murphy, B. (2009). “Does Distributed Development Affect Software Quality? An Empirical Case Study of Windows Vista,” International Conference on Software Engineering, 2009.

Carmel, E., Tjia, P. (2005). “Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce”, UK: Cambridge.

Carmel, E., Agarwal, R. (2002). “The Maturation of Offshore Outsourcing of Information Technology Work”, MIS Quarterly Executive, 1(2), pp. 65-77.

Cataldo, M., Nambiar, S. (2009). “On the Relationship Between Process Maturity and Geographic Distribution: an Empirical Analysis of their Impact on Software Quality,” ACM/SIGSOFT Symposium on the Foundations of Software Engineering (FSE).

DalSotto, F. R. (2009). “QUAMP Analyzer – Analisando a Distância Percebida em Equipes Distribuídas de Desenvolvimento de Software”, Monografia de Conclusão, Especialização em GPTI, PUCRS, FACIN, PPGCC.

Dias Neto, A. C., Subramanyan, R., Vieira, M., Travassos, G. H. (2007). “Characterization of Model-based Software Testing Approaches”, Technical Report TR – ES 713 / 07, COPPE/UFRJ.

Herbsleb, J. D. (2007). “Global Software Engineering: The Future of Socio-technical Coordination,” 29th International Conference on Software Engineering, 188-198, Minneapolis, EUA.

Madlberger, M., Roztocki, N. (2008). “Cross-Organizational and Cross-Border IS/IT Collaboration: A Literature Review,” Proc. of the Americas Conf on Information Systems.

Meyer, B. (2006)., “The Unspoken Revolution in Software Engineering,” IEEE Computer, 39(1), 124, 121-123.

Mirani, R. (2006). “Client-vendor Relationship in Offshore Applications Development: an Evolutionary Framework,” Information Resources Mgmt Journal, 19(4), pp. 71-86.

Nguyen, T., Wolf, T., Damian, D. (2008). “Global Software Development and Delay: Does Distance Still Matter?,” Int’l Conf on Global Software Engineering, Bangalore.

Oates, B. J. (2006). “Researching Information Systems and Computing”, Sage.

Peres, M. (2009). “O Mercado de Serviços Offshore Brasileiro em 2008,” disponível em www2.convergenciadigital.com.br/wordpress/wp-content/uploads/2009/04/idc.pdf, IDC, acesso em Maio de 2009.

Pilatti, L. (2006). “Estrutura e Características para Análise de Ambientes de Desenvolvimento Global de Software em Organizações Offshore Insourcing”, Dissertação de Mestrado, PUCRS, FACIN, PPGCC.

Prikladnicki, R., Audy, J. L. N., Shull, F. (2010). "Patterns in Effective Distributed Software Development," *IEEE Software*, vol. 27, no. 2, pp. 12-15, Mar./Apr. 2010, doi:10.1109/MS.2010.48.

Prikladnicki, R., Audy, J. L. N. (2010). “Distributed Software Development with Captive Centers,” Infocomp Journal of Computer Science, aceito para publicação.

Prikladnicki, R. (2010). "Propinquity in Global Software Engineering: Examining Perceived Distance in Globally Distributed Project Teams," *Journal of* Software *Maintenance and Evolution: Research and Practice*, aceito para publicação.

Prikladnicki, R. (2009a). "Exploring Propinquity in Global Software Engineering," Proc. IEEE Int'l Conference on Global Software Engineering (ICGSE 09), Irlanda.

Prikladnicki, R. (2009b). "QUASE – A Quantitative Approach to Analyze the Human Aspects of Software Development Projects," CHASE Workshop (Collaborative and Human Aspects of Software Engineering) no ICSE, Vancouver, Canadá.

Prikladnicki, R. (2009c). “Padrões de evolução na prática de Desenvolvimento Distribuído de Software em ambientes de *Internal Offshoring*: um Modelo de Capacidade”, Tese de Doutorado, PUCRS, FACIN, PPGCC.

Prikladnicki, R., Audy, J. L. N. (2009a). "Comparing Offshore Outsourcing and the Internal Offshoring of Software Development: A Qualitative Study," Proc. AMCIS 2009 – American Conference on Information Systems, São Francisco.

Prikladnicki, R., Audy, J. L. N. (2009b). "Desenvolvimento Distribuído de Software com *Captive Centers*," Proc. Terceiro Workshop em Desenvolvimento Distribuído de Software (coalocado com o XXIII Simpósio Brasileiro de Engenharia de Software), Fortaleza, Brasil.

Prikladnicki, R., Damian, D., Audy, J. L. N. (2008). “Patterns of Evolution in the Practice of Distributed Software Development: Quantitative Results from a Systematic Review,” Proc. of Evaluation and Assessment in Software Engineering, Bari.

Ramamani, M. (2006). “Offshore Subsidiary Engagement Effectiveness: The Role of Subsidiary Capabilities and Parent – Subsidiary Interdependence,” Conference of Midwest United States Association for IS, pp. 75-80.

Ramasubbu, N., Krishnan, M. S., Kompalli, P. (2005). “Leveraging Global Resources: A Process Maturity Framework for Managing Distributed Development”, IEEE Software, 22(3), pp. 80-86.

Smite, D., Wohlin, C., Feldt, R., Gorschek, T. (2008). “Reporting Empirical Research in Global Software Engineering: a Classification Scheme,” Int’l Conf on Global Software Engineering, Bangalore.

Yin, R. K. (2003). “Case Study Research: Design and Methods”, 3rd edition, Sage.

Construção de um Portal de Conhecimentos de Engenharia de Requisitos

Judith Pavón1 , Erika do Nascimento1 , Sidney Viana2

1 Escola de Engenharia e Tecnologia – Universidade Anhembi-Morumbi São Paulo, SP

2 Vertical Group Solutions Porto Alegre, RS

judith.pavon@gmail.com, erika.sistran@gmail.com, sidney.viana@gmail.com

Abstract. This paper describes a project whose goal is to create an environment, which allows sharing the knowledge generated by the Requirement Engineering area, of open form and free. This environment corresponds to a web portal of Software Requirement Engineering, available in Spanish and Portuguese, for iberoamerican people. This web portal was developed in order to satisfy the needs of the professionals in finding a unique and collaborative space, where academic and corporative people can communicate, discuss, query and share information related to Requirement Engineering.

Resumo. Este artigo descreve um projeto que tem como objetivo criar um ambiente, que permita compartir o conhecimento gerado na área de Engenharia de Requisitos, de forma aberta e gratuita. Esse ambiente corresponde a um portal de Engenharia de Requisitos de Software, disponibilizado em espanhol e português, para o público de ibero-América. Este portal foi construído para atender às necessidades dos profissionais em encontrar um espaço único e colaborativo, onde pessoas do ambiente acadêmico e corporativo possam se comunicar, discutir, consultar e compartilhar informações relacionadas à área Engenharia de Requisitos.

1. Introdução

Portais de conhecimento tem se mostrado um forte aliado na difusão de conhecimentos, independente de qual assunto esteja sendo pesquisado. São muitos portais, sites e até mesmo blogs distribuindo informações que muitas vezes não correspondem ao objetivo principal da procura do usuário, que não consegue entender ou absorver a informação de maneira eficiente. Analisando esse aspecto constatou-se que a internet possui uma variedade de informações que muitas vezes, ao invés de esclarecer, confunde o usuário em sua pesquisa. Ao se realizar buscas por assuntos mais específicos e complexos, é necessário entrar em vários sites, e em muitos casos acessar vários links para se achar o que é preciso, com isso a busca se torna lenta, custosa e em algumas ocasiões inviável. Nos últimos anos tem aumentado muito a procura por informação na área de Engenharia de Requisitos, pois a maioria dos profissionais tem constatado a importância desta área, por ser base para as demais fases do processo de desenvolvimento de software [Sommerville 2007]. Porém, existem poucos espaços na internet que concentrem ou reúnam informações sobre a área de requisitos e permita a troca de conhecimentos em espanhol e português. Este cenário motivou a realização deste projeto.

Este artigo apresenta primeiramente a justificativa na qual o projeto está fundamentado e o objetivo do mesmo, logo após, a metodologia utilizada para elaboração do projeto, logo após os resultados obtidos, descrição das aplicabilidades dos resultados, quais as características inovadoras e para finalizar conclusão e perspectivas futuras.

2. Objetivo e Justificativa

A importância dada à Engenharia de Requisitos (ER) se deve ao fato de que a identificação e documentação correta e precisa dos requisitos influencia diretamente no sucesso de um projeto de software [Kontoya and Sommerville 1998]. Pessoas do ambiente acadêmico, profissionais da área de requisitos e pesquisadores geralmente procuram um espaço onde possam encontrar materiais relevantes relacionado à área de Engenharia de Requisitos, bem como, um ambiente onde possam discutir e compartilhar conhecimentos. Atualmente existem sites que oferecem publicações, evento e modelos de referência, porém, não especificamente da área de Engenharia de Requisitos.

Contudo existe a necessidade de um espaço que possua a consolidação de vários serviços, tais como busca/troca de conhecimentos da área, espaço para discutir e consultar sobre técnicas, ferramentas e conceitos relacionados ao assunto, entre outros.

Este projeto tem como objetivo criar um ambiente, que permita compartilhar o conhecimento gerado na área de Engenharia de Requisitos, de forma aberta e gratuita.

Esse ambiente corresponde a um portal de Engenharia de Requisitos de Software, que utiliza tecnologia Web 2.0 [Lewis 2006] e está disponibilizado em espanhol e português para estudantes e profissionais ibero-americanos da área.

3. Metodologia de Execução

Para o cumprimento das metas estabelecidas para o desenvolvimento deste projeto, foi definida uma metodologia de execução. A Figura 1 ilustra as fases desta metodologia.

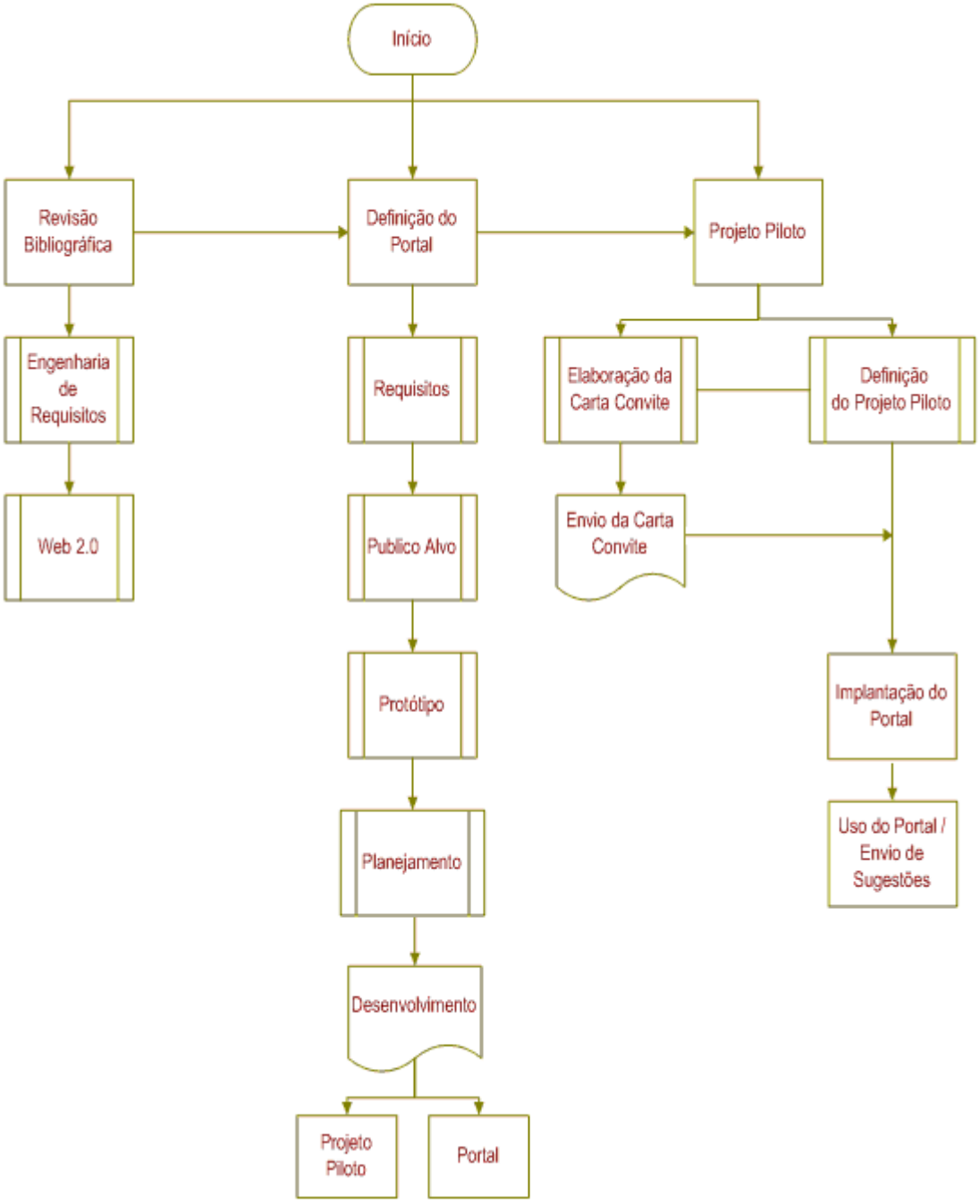


Figura 1 - Estrutura da Metodologia de Execução

Em seqüência serão detalhados objetivamente todos os itens apresentados na Figura 1.

Este trabalho iniciou-se com uma breve revisão bibliográfica sobre Engenharia de Requisitos, no qual foram analisados e discutidos livros, trabalhos acadêmicos e artigos científicos para uma compreensão clara e objetiva sobre o tema. O foco deste estudo foi pesquisar a área para entender seus principais conceitos e aplicabilidade, de forma a ajudar na seleção dos materiais a serem publicados inicialmente no portal.

Ainda nesta fase, foram realizados os primeiros contatos com pesquisadores de universidades ibero-americanas, com as áreas de treinamento de empresas, com estudantes e profissionais interessados na área, com a finalidade de convidá-los a participar do projeto piloto.

Na revisão bibliográfica subseqüente, realizou-se uma análise sobre a Web 2.0 e seus principais conceitos e padrões de projeto. O objetivo dessa análise foi conhecer todos os recursos oferecidos por esse conceito denominado Web 2.0, para auxiliar no desenvolvimento do portal. Portais de conhecimento similares foram analisados, com o intuito de elaborar um quadro comparativo dos mesmos, e assim extrair as melhores características destes.

Na etapa de Definição do Portal realizou-se um levantamento para a definição da estrutura do portal, bem como para a escolha da ferramenta a ser utilizada para o desenvolvimento do portal.

No item Requisitos, foram identificados todos os requisitos necessários para a construção do portal, considerando inclusive aspectos relacionados com usabilidade.

Em Público Alvo, determinou-se que o portal seria voltado para especialistas da área de Tecnologia da Informação, visto que essa área tem uma grande diversidade de profissionais que atuam ou têm interesse na área de requisitos de software. Considera-se de extrema importância que vários níveis hierárquicos de profissionais colaborem com a inclusão de conteúdo no portal, além de sugerir mudanças e avaliar os recursos disponíveis.

Para viabilizar o desenvolvimento, construíram-se protótipos para ter uma visualização inicial da interface, considerando conceitos de usabilidade e de arquitetura da informação. Logo após, foi elaborado o planejamento e a própria construção do portal.

As atividades realizadas durante o projeto piloto são descritas abaixo:

a) Elaboração da Carta Convite: foi elaborado um Convite Oficial a um grupo de profissionais representativo de diferentes universidades e empresas de ibero-América (usuários relacionados à área de Tecnologia da Informação), convidando a participar do projeto piloto. A idéia é que a construção do portal também seja colaborativa, isto é, durante a etapa do projeto piloto, os usuários podem entrar em contato com os administradores do portal, para informar sugestões de mudanças do próprio portal.

b) Definição do Projeto Piloto: nesta etapa foram identificados todos os serviços que deveriam ser disponibilizados nesta fase, e as formas de colaboração dos usuários do portal na carga inicial de informações e no envio de sugestões de mudanças. Esta fase do projeto piloto tem como objetivo validar com os usuários o uso do próprio portal e coletar informações para a melhoria deste portal.

c) Envio da Carta Convite: refere-se ao envio da carta para as pessoas escolhidas a participar no projeto piloto do portal. Foram enviados aproximadamente mil convites.

Uma vez finalizada a etapa do projeto piloto, as melhorias solicitadas para o portal foram implantadas, foram realizados os testes correspondentes, e posteriormente, o portal foi liberado para o seu uso por profissionais relacionados com a área.

4. Resultados Obtidos

Os resultados obtidos neste projeto são dois produtos gerados: o lançamento do portal na internet e a geração de um trabalho de conclusão de curso intitulado “Portal de Conhecimento de Engenharia de Requisitos utilizando Web 2.0” [Nascimento 2009], no curso de Sistemas de Informação da Universidade Anhembi Morumbi.

Os usuários enviaram um conjunto de sugestões para melhoria do portal e avaliaram os recursos disponíveis, durante o projeto piloto. A partir deste conjunto de sugestões, foram implantadas algumas melhorias no portal. Além dos recursos definidos no início, foi incluído no portal um novo serviço denominado “Especialista Responde”, que consiste em um serviço de consultoria de um conjunto de especialistas da área. O usuário envia a sua pergunta, e o especialista responde no máximo em quatro dias. Os especialistas disponíveis são professores universitários que atuam também como consultores da área no mercado de trabalho. Também foi reestruturado o portal, depois de finalizar o projeto piloto, seguindo as sugestões de usabilidade do sistema fornecidas pelos usuários. A etapa do projeto piloto foi fundamental para melhoria das funcionalidades e da interface do portal.

Em relação ao fórum de discussão, foi sinalizado pelos usuários, que não é prático utilizar o serviço de fórum de outro sitio web, que o ideal seria que o portal tivesse seu próprio fórum, de forma a evitar outro cadastro por separado para usar o recurso de fórum. Porém, esta sugestão ainda não foi implantada nesta versão do portal.

Em relação à contribuição na melhoria do processo de software, os profissionais ibero-americanos contam atualmente com um espaço na internet que está no seu idioma nativo e que disponibiliza modelos de referência, melhores práticas, artigos e cursos de Engenharia de Requisitos ou assuntos relacionados, que permitem uma atualização profissional constante e uma troca de experiências com outros colegas da área.

5. Aplicabilidade dos Resultados

A aplicabilidade dos resultados foi analisada em três aspectos: abrangência, relevância e impacto dos resultados. A abrangência do projeto foi cumprida, e inclusive teve um produto adicional, que foi o trabalho de conclusão de curso gerado. Todos os recursos planejados para o portal foram disponibilizados e avaliados pelos usuários que participaram do projeto piloto. Uma vez finalizada a etapa do projeto piloto, foram adicionados alguns recursos e foi melhorada a interface do portal.

A relevância do projeto se reflete na variedade de perfis e de nacionalidades das pessoas e instituições que colaboraram no projeto piloto do portal. Participaram estudantes, analistas de requisitos, programadores, professores, consultores e profissionais que tinham interesse em aprender sobre esta área. Até o momento, o portal teve o acesso de pessoas de 14 nacionalidades diferentes, considerando que ibero-América tem 23 países membros.

O impacto causado pelo portal afetou tanto profissionais já formados como estudantes da área de TI que tinham interesse no assunto, e teve uma maior procura pelo recurso de consulta do que de publicação de materiais até o momento. No projeto piloto, foi elogiado o espaço dedicado a relatos de experiência, pois é uma forma de aprendizado tendo como base experiências de colegas de trabalho. Na etapa do projeto piloto o portal teve uma aceitação maior em português do que em espanhol. Uma das preocupações do portal atualmente é conhecer melhor o seu público, pois entendendo melhor o perfil dos usuários será possível atender melhor a suas expectativas.

6. Características Inovadoras

O projeto criou um ambiente colaborativo, onde o usuário pode buscar, compartilhar e construir conhecimentos de forma coletiva. Além disso, não somente o portal é de natureza colaborativa, como a própria construção do portal foi de forma colaborativa, contando com a opinião de um grupo de usuários que participou ativamente do projeto piloto e permitiu a melhoria do portal, de forma a atender melhor as necessidades e as expectativas dos usuários.

No portal são destacadas as melhores práticas relacionadas com a área de Engenharia de Requisitos, com a possibilidade de discutir, analisar e trocar experiências práticas entre colegas de diferentes culturas. O serviço "Especialista Responde" considera-se também um elemento inovador dentro do portal, tendo em conta que o usuário conta com uma consultoria especializada na internet, sem precisar sair do seu local de trabalho.

Além disso, o próprio usuário poderá indicar fontes de informações adicionais sobre melhores práticas e pode também adicionar materiais complementares. O portal está em português e espanhol, mas os materiais disponíveis para download estão em espanhol, português e inglês.

7. Conclusão e Perspectivas Futuras

O projeto descrito neste artigo atingiu o seu objetivo, pois foi criado um espaço específico na internet, para compartilhar e discutir informações relacionadas à Engenharia de Requisitos (www.engenhariarequisitos.com), tanto em espanhol como em português. Ressalta-se também a importância da colaboração e interatividade dos usuários que participaram do projeto piloto, permitindo validar o uso e a importância do portal, nos ambientes acadêmico e corporativo.

Uma das conclusões detectadas até o momento, é que a maioria dos profissionais que não tinham contato faz algum tempo com alguma instituição acadêmica desconhecia o guia BABoK (Business Analyst Body of Knowledge), sendo um material de consulta muito utilizado nas aulas da área de requisitos nas universidades. Esta experiência demonstra uma troca de conhecimentos entre o ambiente acadêmico e o corporativo.

A proposta para trabalhos futuros consiste em incluir alguns recursos adicionais no portal, principalmente no que se refere ao gerenciamento de ofertas/demandas de profissionais da área de requisitos, personalização da interface do portal para cada usuário específico, bem como, a construção de um fórum próprio do portal. Alem disso, pretende-se realizar um estudo detalhado sobre a possibilidade de incluir a tecnologia Web 3.0 no portal.

Referências Bibliográficas

IIBA (2006). BABoK - IIBA Guide to the Business Analysis Body of Knowledge.

BABoK", Version 1.6. International Institute of Business Analysis.

Kontoya, G. and Sommerville, I. (1998). Requirements Engineering: Processes and Techniques. John Willey & Sons.

Nascimento, E. (2009). Portal de Conhecimento de Engenharia de Requisitos utilizando Web 2.0. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação).

Universidade Anhembi Morumbi. 127p.

Lewis, D. (2006). "What is Web 2.0?", ACM Portal. v. 13, set. 2006. Disponível em: <http://portal.acm.org/citation.cfm?id=1217666.1217669&coll=Portal&dl=GUIDE&C FID=38339728&CFTOKEN=73087676 >. Acesso em: 01 mai. 2009.

Sommerville, I. (2007). Engenharia de Software. São Paulo: Addilson Wesley.

FireScrum: Ferramenta Open-Source

Eric Cavalcanti3, Teresa M. de Medeiros Maciel123, Jones Albuquerque13

1Universidade Federal Rural de Pernambuco – Recife, Brasil
2Universidade Federal de Pernambuco – Recife, Brasil
3CESAR – Centro de Estudos e Sistemas Avançados do Recife – Recife, Brasil

1. Introdução

A adoção de abordagens ágeis tem sido crescente na indústria de software, apresentando resultados positivos em termos de prazo, custo e qualidade (pesquisa version one). O termo "Metodologias Ágeis" tornou-se popular em 2001 quando especialistas em processos de desenvolvimento de software representando as metodologias Scrum, Extreme Programming (XP) e outros, estabeleceram princípios comuns compartilhados por todos esses métodos. Foi então criada a Aliança Ágil e o estabelecimento do "Manifesto Ágil" (AGILEMANIFESTO 2001).

Alinhado aos princípios ágeis, o Scrum compreende atividades de monitoramento e *feedback*, em geral, através de reuniões rápidas e diárias com toda a equipe, visando à identificação e correção de deficiências e/ou impedimentos no desenvolvimento (SCHWABER 2004). A proposta é baseada em um ciclo iterativo e incremental, onde cada iteração é planejada de acordo com a prioridade definida pelo cliente. Ele reúne atividades de monitoramento e feedback, reuniões rápidas e constantes com toda a equipe, visando à identificação e correção de impedimentos no processo de desenvolvimento.
O Scrum tem se apresentado como uma das abordagens ágeis mais adotadas por organizações de software, tanto no contexto mundial como no setor brasileiro de software. Exemplos de organizações que adotam Scrum no Brasil reúne empresas de diferentes portes e cultura, tais como Globo.com, Yahoo Brasil, Pitang S/A, Provider, HP/EDS, Instituto Nokia de Desenvolvimento, C.E.S.A.R, Google Brasil, Power Logic, entre muitas outras. Vale salientar que uma parte substancial destas instituições adotam modelos de maturidade de processo como MPS BR (www.softex.br) e CMMI(www.sei.cmu.edu).

O uso de artefatos simples e não automatizados como cartões e murais recomendados pelo Scrum podem representar um desafio para sua utilização. Dados históricos, por exemplo, podem ser comprometidos pela falta do foco em estimativas baseadas em experiências passadas. Por outro lado, ferramentas automatizadas são consideradas por muitos um fator que pode comprometer a interação do time promovida, em parte, pelo visual e simplista defendido pela metodologia especialmente por equipes remotas (MOUNTAIN 2009).

Inserido neste contexto, este artigo apresenta o Fire Scrum, a primeira ferramenta brasileira open-source e totalmente *free* para suporte a projetos que usam o Scrum como referência. A ferramenta pode ser utilizada por qualquer equipe Scrum, sendo especialmente útil por organizações que pretendem se utilizar de dados históricos e registros sobre o desenvolvimento, assim como por equipes que trabalham total ou parcialmente de forma remota.

A idealização do Fire Scrum nasceu como um projeto de uma equipe no mestrado profissionalizante do C.E.S.A.R em 2008. Seu projeto lógico e arquitetura foi fruto de uma dissertação de mestrado defendida em 2009 e a implementação dos requisitos de sua primeira versão foi realizada em 2009 por cerca de 60 estudantes de mestrado e doutorado da Universidade Federal de Pernambuco, tendo o envolvimento também de professores e profissionais de outras instituições, tais como a Universidade Federal Rural de Pernambuco (www.ufrpe.br) e o C.E.S.A.R (www.cesar.org.br). No segundo semestre de 2009 foi lançada a primeira versão do FireScrum disponível para download no site da ferramenta (www.firescrum.com).

O Fire Scrum contempla o suporte ao planejamento, execução e controle de projetos com Scrum, utilizando-se de design centrado no usuário, interfaces ricas e tecnologias RIA.

2. Objetivos e Justificativa

O Objetivo do projeto Fire Scrum concentra-se na viabilização de uma ferramenta para suportar o planejamento e gerenciamento de projetos de software que adotam metodologias alinhadas à abordagem Scrum, fornecendo facilidades especialmente para equipes remotas, com a premissa de não comprometer os princípios ágeis fundamentais e por outro lado garantir a recuperação de registros e dados históricos.

A idéia do Fire Scrum demandou um trabalho de dissertação de mestrado do C.E.S.A.R, tendo sido motivada especialmente por lacunas existentes em propostas similares e pela demanda por ferramentas de apoio ao Scrum, vivenciada pela própria instituição.

O Fire Scrum, então, foi concebido para ser a primeira ferramenta brasileira open-source e totalmente *free* para suporte a projetos que usam o Scrum como referência, de forma gratuita, e ainda com o código aberto para que a comunidade adicionasse funcionalidades de interesse comum ou espeífico. A ferramenta poderia ser

utilizada por qualquer equipe Scrum, sendo especialmente útil por organizações que pretendem se utilizar de dados históricos e registros sobre o desenvolvimento, assim como por equipes que trabalham total ou parcialmente de forma remota.

Com foco nos objetivos, o projeto FireScrum foi desenvolvimento alinhado às seguintes premissas:

- Construído colaborativamente o FireScrum não seria uma ferramenta de propriedade de nenhuma instituição específica, tendo que contar com um trabalho colaborativo da comunidade para sua construção.

- Foco em usabilidade e simplicidade as práticas e princípios ágeis exigem uma interface que promova interação e que não enfraqueça princípios como comunicação “face-to-face”.

- Foco no Scrum a ferramenta seria concebida com foco no Scrum, no sentido de garantir foco e especialização no que se propõe. Plug-ins e integração com outras ferramentas serão alvo de próximas fases.

- Foco no Time pelos próprios princípios ágeis, que fortalecem muito a iniciativa pessoal, o FireScrum deveria estar atento para a interação do time, daí a escolha por utilizar design centrado no usuário.

Adicionalmente seria fundamental para o projeto que a ferramenta possuísse características inovadoras que atendessem as expectativas da pesquisa dos professores envolvidos, do trabalho de dissertação de mestrado, dos alunos de pós-graduação, assim como da própria comunidade de software.

3. Metodologia de Execução

Conforme abordado na Seção 1, a idéia do Fire Scrum nasceu como um projeto de uma equipe no mestrado profissionalizante do C.E.S.A.R em 2008. Apesar da idéia da ferramenta ter nascido em 2008, devido a limitações financeiras para financia seu desenvolvimento, só em 2009 o projeto lógico e arquitetura do FireScrum fora desenvolvidos e a implementação de sua primeira versão realizada.

Para viabilizar o desenvolvimento da ferramenta sem nenhuma verba disponível, era preciso comprometer instituições e pessoas de forma voluntária e colaborativa. A estratégia de desenvolvimento definida, para atingir a meta de lançamento da primeira versão da ferramenta ainda em 2009, foi a seguinte:

- O projeto do FireScrum (projeto lógico, requisitos, arquitetura e orientação do desenvolvimento) foi definido como produto da dissertação de Eric Cavalcanti, estudante do mestrado profissionalizante do C.E.S.A.R, o qual fez parte também da idealização da ferramenta em 2008. A orientação do projeto ficou a cargo dos professores Jones Albuquerque e Teresa Maciel, ambos alocados como docentes nas instituições UFRPE e C.E.S.A.R.

- Devido ao interesse da comunidade no negócio envolvido (gerenciamento ágil de projetos) e considerando os aspectos inovadores envolvidos, o FireScrum foi escolhido para ser o produto foco do aprendizado de 60 alunos da disciplina Engenharia de Software da pós-graduaçao (mestrado e doutorado) do Centro de Informática da UFPE, curso conduzido pelo Prof.Silvio Meira, cientista que lidera as instituições CESAR e INES (www.ines.org.br). O FireScrum também daria suporte a outras dissertações de mestrado ou doutorado, o que se concretizou ao longo do ano (atualmente o FireScrum é parte dos trabalhos de 2 teses de mestrado e 1 de doutorado em andamento).

Inicialmente o planejamento foi realizado considerando a estratégia supra-exposta, quando foi estabelecido o cronograma macro do ano de 2009 (vide Tabela 1).

A metodologia de desenvolvimento foi fundamentada no Scrum, no sentido de fortalecer o conhecimento das pessoas no próprio negócio da ferramenta. Pelo relevância da usabilidade para a ferramenta, foi adotada práticas de design centrado no usuário, fortalecido com usos de protótipos. Toda a documentação e código esteve sob gerência de configuração utilizando-se ferramentas de controle de versões (SVN), controle de mudanças (Mantis). Todo o gerenciamento foi documentado utilizando ferrramentas do pacote Microsoft Office. A comunicação do time, o qual trabalhou parcialmente de forma remota, foi realizada através da ferramenta de redes sociais Oro-Aro (www.oro-aro.com), também desenvolvida por estudantes de da UFPE.

Tabela 1: Cronograma Geral de Atividades

Atividade	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Visão, Arquit. e Tec.adotada							X					
Anál.de ferramentas smilares	X	X										
Def. da Visão	X	X										
Def. da tecnologia	X	X										
Def. da Arquitetura	X	X	X									
Desenvolvimento												X
Def. papéis			X	X	X	X	X	X	X	X	X	
Def. metodologia												
Formação dos Times												
Desenvolvimento em sprints												
Desenvolvimento em sprints												
Divulgação						X						X
Publicação de artigos						X						X
Desenvolv. do novo site do FireScrum					X	X	X					
Capacitação de pessoas							X					X
Cond. da discipl.ESW.			X	X	X	X	X					
Cond. da discipl. TAES								X	X	X	X	X
Defesa de mestrado							X					

4. Resultados Obtidos

Os resultados obtidos pelo projeto estão descritos a seguir.

Produtos de software gerados

Disponibilização, para a comunidade de software, de uma ferramenta totalmente *free* e *open-source* de suporte a projetos que se utilizam do Scrum como base para o planejamento e gestão, possibilitando seu uso por equipes distribuídas geograficamente e por equipes que precisam resgatar automaticamente dados históricos e registros de atividades (disponível em www.firescrum.com).

As funcionalidades do FireScrum estão organizadas em módulos integrados, mas independentes entre si, os quais estão relacionados a seguir:

Core: responsável por toda a parte operacional básica do Scrum, incluindo controle de acesso, cadastro de usuários e projetos, criação e priorização de *itens de backlog*, criação de *sprints* e tarefas, associação de *itens de backlog* a *sprints*, geração de gráficos de *burndown*, entre outros.

TaskBoard: reproduz o quadro físico ou mural utilizado em projetos reais pelo Scrum. O intuito é permitir que, através do quadro, os usuários da aplicação possam manipular post-its virtuais de forma simples. Inclui visualizar os *itens de backlog* por *sprint*, priorizar itens de *backlog*, cadastrar tarefas (representadas por post-its virtuais), mudar o estado das tarefas arrastando-as pelo quadro, cadastrar impedimentos, entre outros.

Planning Poker: viabiliza o uso remoto da técnica de estimativa Planning Poker. Disponibiliza recursos multimídia para reuniões virtuais de estimativas utilizando comunicação via *chat*, vídeo ou texto.

Além dos módulos supra relacionados, o FireScrum disponibiliza também aplicações de apoio para facilitar a automatização de atividades que não estão diretamente citadas no ciclo Scrum, mas que são relevantes ao desenvolvimento. Os módulos de apoio são o Test Management (gerenciamento de teste), o Bug Tracking (gerenciamento de mudanças) e o Desktop Agent, que facilita o acesso as funcionalidades do FireScrum, uma vez em que o mesmo quando executado estará disponível através da bandeja do

Sistema Operacional, onde o usuário terá acesso a tais funcionalidades através de um menu suspenso.

Capacitação de pessoal

- Cerca de 60 estudantes de pós-graduação (cujos nomes estão relacionados em http://www.firescrum.com/team/) capacitados em gestão ágil de projetos, aplicação do conceito RIA -Rich Internet Application (ALLAIRE 2002) e tecnologias relacionadas.

- Formação de 1 mestre, Eric Cavalcanti, com a dissertação FireScrum - Ferramenta de Apoio à Gestão de Projetos utilizando Scrum, defendida em julho de 2009.

Artigos publicados e palestras convidadas

- Ágiles 2009, Florianópolis, out/2009 - FireScrum - Ferramenta de Apoio à Projetos Scrum (http://www.agiles2009.org/pt/session.php?id=20).

-WDDS'2009, Fortaleza, out/2009 - Ferramenta Open-Source para Apoio ao Uso do Scrum por Equipes Distribuídas (http://www.wdds.ufpb.br/2009/index.php?s=artigos).

- SpinRecife2009: Agilidade na Prática, Recife, Nov/2009 – FireScrum, Ontem, Hoje e o Futuro, (http://agilidadenapratica.wordpress.com/palestras/).

5. Aplicabilidade dos Resultados

O FireScrum está disponível para download em www.firescrum.com, podendo ser utilizado por quaisquer organização que se interessar. Hoje está dentro do escopo de 2 dissertaçoes de mestrado e de uma tese de doutorado do CIn-UFPE.

Dentre as contribuições do projeto, pode-se destacar:

- Disponibilizar via internet de uma ferramenta brasileira de suporte ao processo Scrum, permitindo seu uso por equipes distribuídas geograficamente, possibilitando, por exemplo, a realização de reuniões de planejamento e estimativas utilizando a técnica de Planning Poker (GRENNING 2002) por meio de recursos de áudio e vídeo em tempo real;

- Facilitar a adoção do Scrum por empresas de desenvolvimento de software, especialmente as pequenas e médias empresas que não querem onerar a adoção de práticas ágeis de desenvolvimento de software com custo adicional de ferramentas;

- Prover, através da ferramenta, a geração automática de gráficos, métricas e histórico, evitando assim o esforço necessário para o levantamento e consolidação de tais informações.

- Aplicar tecnologia inovadora, como RIA (Rich Internet Application) e tecnologias relacionadas, uma vez que tal conceito está sendo cada vez evidente no desenvolvimento de aplicações, e arquitetura de uma aplicação, utilizando a tecnologia RIA Adobe Flex, uma vez que a referida tecnologia tem conquistado uma fatia considerável do mercado, e este trabalho pode servir de referência na definição de uma arquitetura para uma aplicação que utiliza o Adobe Flex;

- Produção e distribuição de uma ferramenta de forma colaborativa no modelo open source, envolvendo estudantes, professores e profissionais.

Até o final de 2009 o FireScrum contava com mais de 21 mil acessos provenientes de diversos países e 4.286 downloads provenientes do Brasil e de outros países (vide Figura 1, extraída do Google Statistics.



Figura 1: Acessos ao site FireScrum até dezembro de 2009.

Não se tem informações concretas de quais empresas utilizam o FireScrum, pois seu download é livre por qualquer usuário que acesse o site, mas dentre elas pode-se ressaltar o C.E.S.A.R (Recife) e o Instituto Nokia de Tecnologia (Manaus e Recife).

Na comunidade ágil mundial o FireScrum também teve um papel de divulgação ou mesmo de fortalecimento da imagem brasileira no que diz respeito a desenvolvimento de software e gestão ágil de projetos. Dentre os comentários que podem ser encontrados em sites sobre o assunto, pode-se destacar a afirmação do blog Scrum4you, www.borisgloger.com, “the most impressive open source tool I found: FireScrum”.

6. Características Inovadoras

Dentre as características inovadoras, podem se ressaltar o próprio negócio envolvido (gerenciamento ágil de projeto) ainda carente de ferramentas, especialmente open-source e sem custo.

O FireScrum, por ser uma aplicação de internet rica (RIA), combina o melhor das funcionalidades, interatividade e multimídia das interfaces de aplicações desktop com a alta disponibilidade, baixo custo de distribuição de aplicações web. Utiliza o Adobe Flex pela Adobe, ma tecnologia lançada em Março de 2004 pela Macromedia, que suporta o desenvolvimento de aplicações ricas para a Internet, baseadas na plataforma do Macromedia Flash (TAPPER 2008).

Através de uma arquitetura modular e extensível é possível a criação de novos módulos, expandindo assim suas funcionalidades. Com o uso de APIs de integração (Application Program Interface) é possível realizar a integração do FireScrum com ferramentas e aplicações de terceiros.

7. Conclusão e Perspectivas Futuras

O desenvolvimento de novos módulos está sendo previsto como trabalhos futuros. Algumas funcionalidades estão sendo planejadas como o uso de uma tela projetada com a tecnologia multi-touch, pode simular ainda mais o uso do quadro de tarefas (taskboard) físico, o que evitaria o uso de cartões de papel, contribuindo assim também com um posicionamento ecologicamente correto. O uso dos recursos de áudio e vídeo disponíveis no módulo Planning Poker, também poderá ser utilizado para viabilizar as reuniões diárias com times geograficamente distribuídos, como também adicionados ao módulo Desktop Agent para facilitar a comunicação entre os integrantes de um projeto.

Com o lançamento da primeira versão oficial pública, o FireScrum pretende, cada vez mais junto com a comunidade, evoluir para se tornar a melhor ferramenta open source de apoio ao Scrum disponível no mercado mundial.

Atualmente participantes do desenvolvimento do FireScrum estão analisando possíveis modelos para garantir a continuidade da ferramenta de forma sustentável.

Referências Bibliográficas

AGILEMANIFESTO, 2001, “Manifesto for Agile Software Development” http://agilemanifesto.org/. Acesso em: 02/04/2008.

ALLAIRE, J. Macromedia Flash MX – A Next-Generation Rich Client. Macromedia white paper, 2002.

MOUNTAIN GOAT SOFTWARE. Task Board. Disponível em: http://www.mountaingoatsoftware.com/task-boards. Acesso em: 23/02/2009.

SCHWABER, Ken. Agile Project Management with Scrum. Redmond: Microsoft Press, 2004.

TAPPER, Jeff. ADOBE FLEX 3 Training from the Source. Adobe Press, 2008.

Implantação de Gerência do Conhecimento no CTIC-UFPA com o Ambiente WebAPSEE

Jadielly F. Oliveira1, Gabriela F. Andrade1, Lorena C. Tavares1 Antonio J. G. Abelém1, 2, Rodrigo Quites Reis1, Carla A. Lima Reis1

1 Programa de Pós-Graduação em Ciência da Computação (PPGCC) 2 Centro de Tecnologia da Informação e Comunicação (CTIC) Universidade Federal do Pará (UFPA) Rua Augusto Correa 01 – Belém – PA – Brasil

{jady, gfa}@ufpa.br, lorena@webapsee.com, {abelem, quites,clima}@ufpa.br

Abstract. Software development processes produce a large amount of knowledge that should be used by organizations to improve both software process and product quality. This report presents the experience on deployment of a software infrastructure to support knowledge management planning and execution, which is integrated to a Process-Centered Software Engineering Environment in the context of Information Technology Center at Federal University of Pará.

Resumo. Processos de desenvolvimento de software geram grande quantidade de conhecimento que deve ser aproveitado pela organização para melhoria da qualidade do processo e consequentemente do produto a ser desenvolvido.

Este artigo apresenta o relato da implantação de uma infra-estrutura de apoio ao planejamento e execução de estratégias para gerenciar conhecimento, integrada a um ambiente de desenvolvimento de software centrado em processos no Centro de Tecnologia da Informação e Comunicação da Universidade Federal do Pará.

1. Introdução

Organizações de desenvolvimento de software requerem intenso fluxo de conhecimento para a execução das suas atividades. Muitas vezes esse conhecimento é perdido, seja com a saída de membros da equipe, seja com o esquecimento ao passar o tempo. Com isso a organização perde o que vem sendo considerado, atualmente, um dos seus principais ativos que é o capital intelectual [Rus e Lindvall 2002]. Essa perda faz com que erros que poderiam ser evitados voltam a acontecer, assim como problemas já solucionados voltam a ser repensados por não existirem registros da solução. Neste cenário, a qualidade e produtividade da organização são comprometidas e, conseqüentemente, aumentam o tempo e os custos de seus projetos, entre outros problemas. Por tal motivo, preservar e utilizar o conhecimento organizacional com eficiência tornou-se um fator decisivo para que organizações de software possam atingir níveis mais altos de maturidade e qualidade.

Gerenciar o conhecimento é um processo que ajuda as organizações a encontrar, selecionar, organizar, disseminar e transferir as informações importantes e as experiências necessárias para as várias atividades da organização, tais como, resolução de problemas, aprendizagem dinâmica, planejamento estratégico e tomada de decisões [Probst, Raub e Romhardt 1999]. Tendo em vista essa perspectiva, modelos de maturidade de software como o MPS.BR - Melhoria do Processo de Software Brasileiro [MPS.BR 2007] e normas como a ISO/IEC 12207 [2008] já possuem recomendações e práticas que evidenciam a gerência do conhecimento como atividade obrigatória para que uma organização atinja maior maturidade e capacidade de desenvolvimento de software.

Muitas organizações desenvolvedoras de software utilizam Ambientes de Desenvolvimento de Software Centrados em Processos (do inglês Process-Centered Software Engineering Environment - PSEE) [Arbaoui et al., 2002] para modelagem e execução de processos de software permitindo apoio às atividades planejadas. Um PSEE armazena as informações sobre o processo e seus envolvidos o que pode auxiliar a gerência do conhecimento organizacional.

De acordo com Fisher (2001), a integração com o ambiente de trabalho dos membros da organização é um dos fatores determinantes para o sucesso da implantação de Gerência de Conhecimento (GC) em uma organização. No contexto do processo de software, esse ambiente de trabalho é, frequentemente, um PSEE. Dessa forma, o conhecimento é melhor aproveitado se existir no PSEE uma infra-estrutura que apóie o planejamento e execução de estratégias de Gerência de Conhecimento [Natali, 2003].

Apesar da existência de diversas aplicações, poucos estudos relatando a aplicação prática de GC são realizados em organizações de pequeno porte, e menos ainda relatando os benefícios da integração com PSEEs. Com isso torna-se inviável afirmar que a implantação das mesmas proporciona vantagens em relação a outras abordagens, bem como ainda comprovar os reais benefícios obtidos pelas organizações após a implantação das mesmas. Por isso a implantação de GC integrado em um PSEE no contexto da Divisão de Desenvolvimento de Software do Centro de Tecnologia da Informação e Comunicação (CTIC) da UFPA, pode

trazer benefícios não somente para a organização, mas para o incremento na pesquisa em GC no desenvolvimento de software.

Adicionalmente a esta parte inicial, são apresentados os objetivos e justificativas do projeto na seção 2, a metodologia de execução na seção 3, os resultados obtidos na seção 4, aplicabilidade dos resultados na seção 5, características inovadoras do projeto na seção 6 e por fim as considerações finais e perspectivas futuras na seção 7.

2. Objetivos e Justificativa

Segundo Davenport e Prusak (1999) conhecimento é o "conjunto de experiências, valores, informações contextuais e percepção de especialistas que provêm uma forma para avaliar e incorporar experiências e informação. É originado e aplicado na mente das pessoas. Nas organizações, geralmente é encontrado não apenas em documentos e repositórios, mas também nas rotinas, processos, práticas e normas organizacionais".

Para organizar esse conhecimento é necessário estabelecer procedimentos, técnicas e ferramentas de forma a gerenciar esse ativo de forma eficiente.

A divisão de desenvolvimento do CTIC/UFPA – Centro de Tecnologia da Informação e Comunicação da Universidade Federal do Pará – é uma unidade organizacional avaliada no nível G do MPS.BR em dezembro de 2008, composta treze membros, dos quais oito são bolsistas. Por conta do alto número de bolsistas a unidade é caracterizada pela alta rotatividade de pessoas, fazendo com que haja perda do conhecimento adquirido pelos mesmos. Embora o CTIC utilize o ambiente de desenvolvimento de software WebAPSEE[1] - que concentra grandes quantidades de informações sobre os processos - para gerenciar a execução dos processos de software, nenhuma solução era aplicada para gerenciar o conhecimento criado pelos membros da organização. Dessa forma, a organização não conseguia reter o conhecimento, dificultando o aprendizado da equipe. Considerando o cenário exposto, houve a motivação para a implantação da Gerência do Conhecimento na divisão de desenvolvimento com apoio ferramental do ambiente WebAPSEE e definição de uma estratégia adequada à realidade da organização. Ressalta-se ainda que esta ação está associada com a iniciativa do CTIC em ser avaliada no nível E do MPS.BR até o ano de 2011.

O objetivo central do projeto foi obter a melhoria da qualidade dos produtos e serviços ofertados pelo CTIC, através da definição e implantação do processo de Gerência de Conhecimento com ferramenta integrado ao WebAPSEE. Além disso, o projeto teve como objetivo estimular o grupo de gerentes e desenvolvedores do CTIC a realizarem as práticas de GC, através de treinamentos e palestras de motivação.

Os objetivos específicos do projeto estão descritos a seguir.

[6]2.1. Objetivos Específicos

- Avaliação da ferramenta e da estratégia de GC propostos, através do Obtenção de um diagnóstico que permitiu direcionar o trabalho de implantação de GC no CTIC. Tal diagnóstico foi constituído de um levantamento da situação atual do CTIC, das equipes de desenvolvimento, e das práticas adotadas;
- Conscientização das pessoas envolvidas com desenvolvimento de software e da alta direção do CTIC acerca da necessidade da implantação da GC;
- Definição de uma estratégia de GC adequada com a realidade e as necessidades do empresa e de acordo com normas e modelos de maturidade, especificamente em relação ao Modelo MPS;
- Implantação da estratégia em projetos piloto utilizando um Wiki – disponível em http://mps.ctic.ufpa.br (com acesso restrito);
- Desenvolvimento de uma nova versão do ambiente WebAPSEE com o módulo de apoio à GC e a Agenda Web integradas com a base de processos do CTIC;
- Apoio na implantação de GC através de treinamentos e acompanhamento na utilização da ferramenta;
- Acompanhamento de projetos;
- Elaboração de um relatório final com os resultados obtidos.

3. Metodologia de Execução

A metodologia executada para obtenção de cada objetivo citado acima está descrita nos tópicos a seguir:

[1] WebAPSEE é um Software Livre para gestão do processo de software desenvolvido pelo Laboratório de Engenharia de Software da UFPA que alcançou em 2006 o Segundo lugar no Prêmio Dorgival Brandão Júnior da Qualidade e Produtividade em Software. Na avaliação formal MPS.BR do CTIC/UFPA em 2008 o WebAPSEE foi reconhecido como ponto forte da implementação.

1.Diagnóstico da organização: Foi conduzido a partir da aplicação de entrevistas semi-estruturadas abertas em toda a organização, com os perfis Gerentes, Desenvolvedores Novatos e Desenvolvedores Experientes. Ao todo foram realizadas dezesseis entrevistas, de em média trinta minutos cada. O objetivo de utilizar essa divisão de perfis foi observar as dificuldades que cada um enfrenta em relação ao fluxo de conhecimento na organização, bem como identificar as principais necessidades de cada perfil;

2.Conscientização e motivação: Foram realizadas duas palestras com a equipe da Divisão de Desenvolvimento contando também com a participação de membros de outros setores da organização, em maio de 2009. Durante as palestras foram apresentados os conceitos, benefícios e desafios da GC;

3.Elaboração da estratégia para GC: A partir da análise das entrevistas, do perfil da organização, e dos resultados obtidos com a palestra motivacional, foi definida a estratégia de GC a ser implantada na organização, que contempla os tipos de conhecimento a serem coletados, os responsáveis pela gerência do conhecimento e os processos nos quais será implantada a estratégia;

4.Implantação do processo de GC com o Wiki: Para coletar os requisitos para a ferramenta, bem como realizar uma pré-avaliação da estratégia, o processo foi implantado em dois projetos piloto com o apoio de um Wiki institucional, para que seja possível a avaliação do antes e depois da implantação na organização;

5.Apresentação e refinamento da estratégia definida: Reunião com a equipe de desenvolvimento para apresentar a estratégia definida para discussão/avaliação;

6.Desenvolvimento da ferramenta: a partir da avaliação da implantação com o Wiki foi iniciado o desenvolvimento da ferramenta com o módulo de GC integrado à versão 1.5 ambiente WebAPSEE;

7.Treinamento sobre o módulo de GC do WebAPSEE: Foram realizados treinamentos práticos acerca da utilização da ferramenta no CTIC;

8.Observação e acompanhamento na execução dos processos: Os processos nos quais estiverem sendo realizados a GC estão sendo monitorados e acompanhados pela equipe de implantação com o objetivo de observar a execução das atividades e a quantidade de itens que estão sendo inseridos;

9.Análise dos resultados e elaboração do relatório final: Ao final de todo o projeto, haverá a elaboração de um relatório final com a avaliação do processo em relação à melhoria da qualidade do processo do CTIC;

4. Resultados Obtidos

Os produtos gerados com a execução do projeto foram: 1) um processo-padrão para implantação de GC, 2) um software de apoio à criação e execução de estratégias de GC (integrado a um PSEE), 3) desenvolvimento de uma dissertação de mestrado no PPGCC-UFPA [Oliveira, 2010], 4) duas publicações técnicas em conferência da área de Engenharia de Software [Oliveira et al, 2009 a e b], e 5) também qualificação formal de profissionais através dos treinamentos e palestras realizadas no CTIC-UFPA. Além disto, o projeto contou com também com dois alunos de Iniciação Científica e um aluno do Curso de Residência em Arquitetura de Software do LABES-UFPA que atuaram como auxiliares de pesquisa e desempenharam papel importante no levantamento da situação do CTIC, na configuração do Wiki, e na construção do ferramental proposto.

4.1. Processo-padrão para Implantação de GC

O processo desenvolvido tem como objetivo guiar a implantação de gerência do conhecimento em organizações de desenvolvimento de software, no entanto, o mesmo pode ser adaptado para diferentes contextos e pode ser aplicado em outras áreas de negócio. Assim como em outros processos da engenharia de software, foram definidas atividades com sequências lógicas, artefatos de entrada e saída foram gerados e papéis foram criados e alocados para execução das atividades. As macro-atividades do processo de GC são ilustradas[2] na Figura 1 a seguir na notação do WebAPSEE [Lima Reis, 2007].

7

[2] A figura 1 apresenta apenas uma visão geral do modelo de processo implantado. A sua descrição completa é fornecida em [Oliveira, 2010].

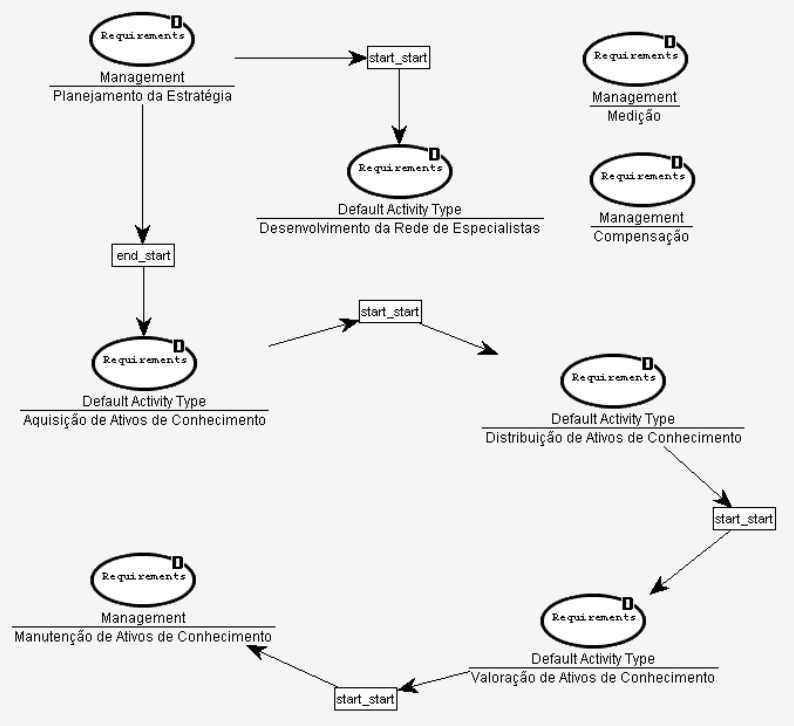


Figura 1. Visão Geral do Processo de Gerência do Conhecimento

4.2. Ferramenta: WebAPSEE Knowledge Manager (WKM)

A integração de funcionalidades do módulo WKM foi realizada nos dois principais componentes do WebAPSEE: no Manager Console (usado na definição e acompanhamento dos processos) e na TaskAgenda (usado pelos desenvolvedores que atuam nos projetos). O Manager e Agenda interagem entre si através de um servidor que gerencia a execução do processo e o fluxo de conhecimento na organização. Os desenvolvedores também podem se comunicar através de suas agendas. Todos os itens de conhecimento registrados são armazenados em uma base e os anexos e artefatos relacionados estão em um repositório de dados. A figura 2 ilustra uma visão geral do funcionamento do sistema.



Figura 2. Visão geral do funcionamento do sistema.

Durante o desenvolvimento da ferramenta primou-se pelo atendimento do modelo MPS com o objetivo de proporcionar apoio ferramental para que as organizações pudessem alcançar mais níveis de maturidade do modelo. Como exemplo disso, a ferramenta permite a formação de uma rede de especialistas, bem como a coleta automática de ativos de conhecimento na Agenda Web do Desenvolvedor (Figura 3) conforme estabelecido pelo gerente no planejamento do projeto. Assim, o conhecimento coletado é automaticamente associado ao contexto da atividade do projeto, dos desenvolvedores envolvidos, e dos artefatos consultados ou produzidos.

5. Aplicabilidade dos Resultados

Com a implantação da Gerência do Conhecimento no CTIC-UFPA, foi possível disseminar as boas práticas e benefícios alcançados no meio acadêmico, através de publicações de artigos e trabalhos a nível nacional.

Palestras motivacionais e treinamentos foram ministrados para a Unidade Organizacional, e embora a implantação tenha sido realizada apenas em uma das sub-gerências do CTIC, a divulgação dos resultados foi feita para a organização como um todo, incentivando as outras gerências - não ligadas ao desenvolvimento de software - a realizar GC.

O processo-padrão proposto tem caráter geral e pode ser adaptado para implantação em outras organizações que desejem alcançar o nível E do MPS.BR que exige como resultados esperados do processo de gerência de recursos humanos aspectos relacionados à GC, independente da técnica ou ferramenta que a organização utilize. Por sua vez, a ferramenta é útil para apoiar a criação e execução da estratégia de GC.

Além disso, no meio acadêmico, as publicações técnicas e o trabalho de mestrado desenvolvidos, podem incrementar os estudos e pesquisa na área de GC na engenharia de software, servindo de embasamento para a realização de outros estudos na linha de pesquisa na utilização do processo e da ferramenta, por exemplo.

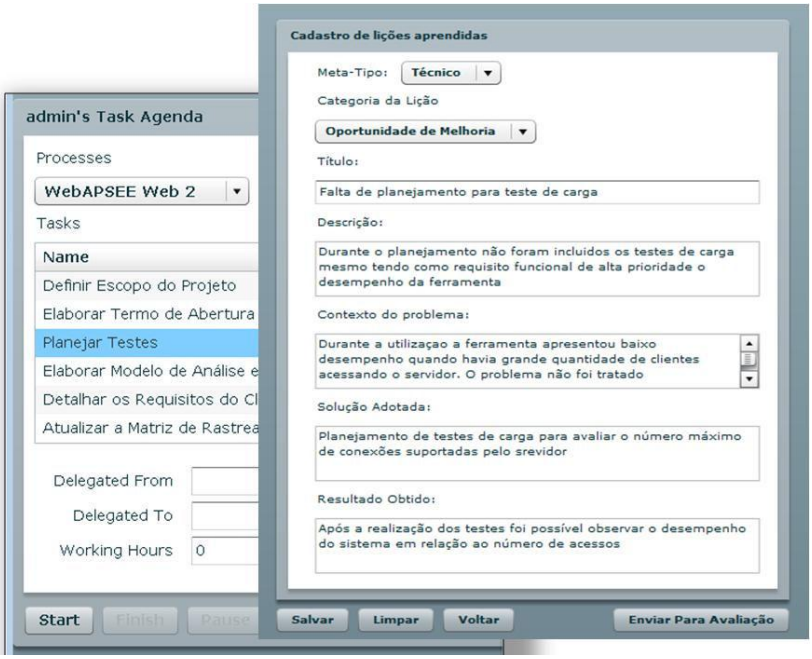


Figura 3. Exemplo de interface de coleta de conhecimento na Agenda Web.

6. Características Inovadoras

Como aspectos de inovação relacionados ao projeto, pode-se enfatizar a elaboração de um processo-padrão que prima pelas exigências de normas e modelos de maturidade, além do desenvolvimento de uma ferramenta integrada a um PSEE o que permite uma rápida implantação do processo de GC aliada a flexibilidade na sua evolução para atender novos requisitos do projeto ou da organização. A coleta automática de conhecimento proporcionada pela ferramenta – com base no planejamento prévio - fornece um controle ágil acerca das contribuições dos envolvidos, além de facilitar a estruturação da informação em relação aos demais elementos do processo executado.

7. Conclusões e Perspectivas Futuras

Este artigo apresentou o projeto de implantação da GC no CTIC-UFPA, detalhando seus principais objetivos, a metodologia utilizada no decorrer do projeto, e os principais resultados obtidos com a realização do projeto.

A principal contribuição deste trabalho consiste em proporcionar flexibilidade e automação na implantação de estratégias de GC, através do processo-padrão e da ferramenta, respectivamente. Além disso, será possível avaliar a utilização a partir do acompanhamento da implantação, ou até mesmo em uma avaliação oficial MPS.BR.

Como perspectiva futura pretende-se implantar GC utilizando o processo padrão elaborado, juntamente com a ferramenta em mais organizações para avaliar sob diferentes aspectos e em diferentes contextos, quão útil a GC pode ser no desenvolvimento de processos de software.

Outra perspectiva de trabalho futuro com a utilização da ferramenta é estabelecer um mecanismo de troca de conhecimento entre diferentes sistemas dentro da perspectiva do projeto “Modelo Interoperável de Informações de Processos de Software” desenvolvido no contexto dos Projetos em Interoperabilidade de Software desenvolvidos no LABES-UFPA [Reis, 2010]. Assim, pode ser criado um vocabulário genérico, o que viabilizaria a utilização de ontologias de conhecimento para compartilhamento de conhecimento sobre o processo de software em nível global.

8. Referências

Arbaoui, S.; Derniame, (2002) “A comparative review of Process-Centered Software Engineering Environments”. Annals of Software Engineering, vol. 14, p. 311-340.

Davenport, T., Prusak, L. (1999), “Conhecimento Empresarial”. Rio de Janeiro: Ed. Campus.

Fischer, G., Ostwald, J., (2001), “Knowledge Management: Problems, Promises, realities, and Challenges”, IEEE Intelligent Systems, v. 16, n. 1 (Jan/Feb), pp. 60-72.

ISO/IEC 12207 International Organization for Standardization and International Electrotechnical Commission (2008) “ISO/IEC 12207:2008 Systems and Software Engineering -Software Life Cycle Processes”. Genebra: ISO, Março.

LABES (2007), “Versão para Web 2.0 do WebAPSEE”. Disponível em http://www.mct.gov.br/upd_blob/0019/19585.pdf.

Lima Reis, C. A.; Reis, R. Q. (2007) “Laboratório de Engenharia de Software e Inteligência Artificial: Construção do Ambiente WebAPSEE”. ProQualiti – Qualidade na Produção de Software. v. 3, n. 1, junho de 2007. p. 43-48.

MPS.BR Associação para Promoção da Excelência do Software Brasileiro – SOFTEX (2007) “MPS.BR – Guia Geral”, versão 1.2, Junho. Disponível em: www.softex.br.

Natali, A. C., (2003) “Uma Infra-Estrutura para Gerência de Conhecimento em um Ambiente de Desenvolvimento de Software”. Dissertação de Mestrado.UFES.

Oliveira, J. F. ; Andrade, G. F. ; Tavares, L. C. ; Reis, C. A. L. (2009a). Planejamento e Execução de Gerência do Conhecimento em um Ambiente de Desenvolvimento de

Software. In: VIII Simpósio Brasileiro de Qualidade de Software, Ouro Preto.

Oliveira, J. F. ; Reis, C. A. L. (2009b) . Apoio Automatizado à Elaboração de Planos de Gerência de Conhecimento para Processos de Software. In: XII Conferência Iberoamericana de Ingenieria de Requisitos y Ambientes de Software, 2009, Medelín.

Oliveira, J.F. (2010). Apoio a Gerência do Conhecimento em um Ambiemte de Desenvolvimento de Software Centrado em Processos. 152f. Dissertação (Mestrado em Ciência da Computação) – PPGCC-UFPA, Belém.

Probst, G. J. B.; Raub, S.; Romhardt, K. (1999) “Managing Knowledge: Building Blocks for Success”, 368 pp, Ed. Wiley.

Reis, R.Q. (2010) Modelo Interoperável de Informações de Processos de Software. http://miips.codeplex.com/

Rus, I.; Lindvall, M., (2002) “Knowledge management in software engineering”, IEEE Software, vol.19, no.3, pp.26-38.

Padrões e Processos para Arquitetura de Software do Governo do Estado do Pará: uma Parceria PRODEPA e UFPA

[1]Cláudio R. de L. Martins1, Adailton M. Lima, [2]Carla A. Lima Reis2, Rodrigo Quites Reis

1 Processamento de Dados do Estado do Pará (PRODEPA) Belém – Pará – 66820-000
http://www.prodepa.pa.gov.br - claudio.martins@prodepa.pa.gov.br

2 Laboratório de Engenharia de Software (LABES) – Instituto de Ciências Exatas Universidade Federal do Pará (UFPA) - Belém – Pará - 66075-110
http://www.labes.ufpa.br - adailton@webapsee.com, clima@ufpa.br, quites@ufpa.br

Abstract. *This paper describes the main results of the technical cooperation between PRODEPA and UFPA. The main goals of this cooperation are the creation of directives, patterns and processes related to Software Architecture on the context of the State Government of Pará in Brazil. The main results are related to the advance on the software development methodologies and the development of directives that are applied to software acquisition on the State Government of Pará.*

Resumo. *Este artigo descreve os principais resultados da cooperação técnica entre PRODEPA e UFPA, cujo objetivo é a criação de diretivas, padrões e processos voltados para Arquitetura de Software no contexto do Governo do Estado do Pará. Este projeto tem como principais resultados o avanço na metodologia de desenvolvimento e teste de software público na PRODEPA e a definição de diretrizes de aquisição de software no contexto do Governo Estado do Pará.*

1. Introdução

O investimento em Tecnologia da Informação é um dos principais objetivos dos Governos em todo mundo para prover a melhoria na qualidade dos serviços para o cidadão. No contexto específico do Pará, a PRODEPA – Empresa de Processamento de Dados do Estado do Pará, é a principal responsável pelo desenvolvimento de soluções de software e em telecomunicações para os diversos setores da Administração Pública.

A Diretoria de Desenvolvimento de Sistemas é a unidade gestora responsável pelo setor de software nesta instituição e alcançou em 2007 o nível F do MPS.BR.

Ciente da importância de se lidar de maneira estruturada com o tema Arquitetura de Software, desde 2007 a PRODEPA tem buscado parcerias com instituições de ensino superior da região para capacitar seu pessoal e implantar as melhores práticas no cotidiano da empresa. Neste contexto, em 2008 foi iniciado o Curso de Especialização na modalidade de Residência em Arquitetura de Software onde a PRODEPA foi a principal parceira de um projeto envolvendo a UFPA, a Universidade da Amazônia - UNAMA e o Centro de Ensino Superior do Estado do Pará - CESUPA na formação de quatorze bolsistas que desenvolvem projetos vinculados aos problemas da empresa.

O projeto aqui apresentado é resultado de uma cooperação técnico-científica entre o Laboratório de Engenharia de Software da UFPA e a PRODEPA, com o objetivo de estabelecer e manter padrões de arquitetura de software para estabelecer soluções para problemas recorrentes na PRODEPA, com base tanto em soluções já descritas na literatura quanto em soluções inéditas.

Para tanto, foram examinados 696 sistemas desenvolvidos pela PRODEPA e utilizados por 55 órgãos do Governo. Deste total, 64,95% está hospedado em Windows, enquanto que 21,75% está em IBM/390 - Mainframe, e 13,3% está em Linux e diferentes tipos de Unix. A maioria dos sistemas são desktop - 43,65%, enquanto sistemas Web com páginas dinâmicas constituem 29,71%, e o sistemas baseados em caracteres – terminais ficam com 21,20%.

Além da definição de modelos arquiteturais que estejam adequados ao contexto organizacional da empresa, também inclui a definição de metodologias que permitam a institucionalização de padrões e política de melhoria contínua. No contexto governamental existe a necessidade de acesso integrado às informações do governo, em virtude da descentralização dos órgãos e seus setores de Tecnologia da Informação (TI).

Dentre os principais problemas existentes podem-se citar: interoperabilidade com sistemas legados; migração de sistemas legados; adequação à padrões de integração nacionais; políticas de gestão para aquisições de recursos de TI.

Este projeto também trata da necessidade dos setores que desenvolvem e compram software no Governo. Um dos principais problemas neste contexto é a falta de interoperabilidade entre os sistemas do Estado e retrabalho em sistemas adquiridos sem padronização compatível aos sistemas já existentes.

2. Objetivos e Justificativa

Este projeto se justifica pela constante busca de melhorias nos serviços públicos. Neste caso, existe a necessidade de políticas que estejam alinhadas com a melhoria dos serviços de desenvolvimento e aquisição de software no contexto do Estado do Pará.

Atualmente existe alta rotatividade de pessoal, causando grande impacto no processo de desenvolvimento quando há dependência do conhecimento dos profissionais envolvidos. Neste caso, existe a necessidade de padronização de técnicas e metodologias como forma de apoiar a gerência da empresa e facilitar treinamento e adequação de funcionários contratados.

O processo de melhoria da qualidade iniciou com a obtenção do nível F do MPS.BR na PRODEPA em 2007. Foi percebida necessidade de avançar em melhorias com apoio mais direcionado aos produtos construídos na empresa. Portanto, o Projeto tem como objetivo avançar na institucionalização de padrões arquiteturais e metodologias de desenvolvimento de software na PRODEPA, visando abrangência e benefícios a todo Governo do Estado do Pará.

As frentes de trabalho definidas para realização desse projeto incluem:

- Consolidação de políticas e diretrizes para a área de TIC no que concerne ao desenvolvimento e aquisição de software;
- A especificação dos padrões arquiteturais de software visando interoperabilidade, envolvendo premissas, políticas e especificações técnicas que regulamentem a utilização da Tecnologia de Informação e Comunicação (TIC) no governo do Pará;
- Aumento da maturidade na área de software na PRODEPA, a partir da padronização de metodologias, técnicas e ferramentas.

O projeto tem como principal justificativa técnica a necessidade de padronização da Arquiteturas de Software principalmente pela demanda de maior interoperabilidade de sistemas construídos internamente ou adquiridos de fornecedores externos e, por conseguinte, diminuição do retrabalho e facilidade na integração, quando necessário.

3. Metodologia de Execução

Durante a inicialização do projeto foi realizado o levantamento de necessidades de padronização arquitetural de software com o objetivo de compreender a arquitetura geral e o funcionamento interno dos sistemas já existentes. Nesta etapa foi realizado um estudo de campo envolvendo os principais órgãos-clientes de Tecnologia da Informação no Governo do Pará – a saber: o Departamento de Trânsito (DETRAN), a Secretaria de Estado de Educação (SEDUC), e a Secretaria de Estado de Saúde Pública (SESPA) - representando uma amostra dos principais problemas técnicos de arquitetura de software no contexto governamental analisado. A partir desse levantamento de necessidades foi realizado o planejamento das atividades de catalogação de padrões arquiteturais e desenvolvimento de processos e metodologias para desenvolvimento de software no contexto da PRODEPA.

O projeto foi continuamente avaliado a cada quatro meses em reuniões que envolviam os gestores da PRODEPA e representantes da equipe técnica da UFPA.

Nestas reuniões eram definidas prioridades para próximas etapas e avaliação dos resultados alcançados até aquele momento. As mudanças no andamento do projeto eram acordadas entre o líder do projeto pela UFPA e representantes da PRODEPA, para direcionar as ações do projeto na obtenção dos objetivos definidos no plano do projeto para o período quadrimestral. Todos os resultados do projeto são compartilhados a partir da utilização de um wiki organizacional, onde são disponibilizados documentos e atividades durante o andamento do projeto.

4. Resultados Obtidos

As subseções a seguir apresentam os produtos e resultados relevantes desde o início em 2008 até o período atual. O projeto se estende até o final de 2010, embora a maioria dos seus resultados esperados já tenham sido alcançados no momento da sua submissão deste relatório.

4.1. Processos que contribuem para a melhoria da qualidade de software descritos e institucionalizados

Os processos e práticas elaborados foram institucionalizados a partir da construção de catálogos direcionados às principais etapas do processo de desenvolvimento de software e constituição de grupos de trabalho direcionados à problemas de arquitetura de software. A seguir são descritos os principais documentos e grupos de trabalho institucionalizados para melhoria da qualidade do software produzido na PRODEPA.

4.1.1. Apoio à Análise e Projeto de Desenvolvimento de Software

Os processos de análise e projeto de software foram melhorados a partir da identificação de *gap* das práticas que eram até então aplicadas. Neste sentido foram institucionalizadas práticas de Desenvolvimento de Requisitos e Projeto Arquitetural de Software, os quais foram consolidados no documento institucional denominado **Catálogo de Apoio à Análise e Projeto (CAPS)**. Este documento contém definições técnicas e recomendações aplicadas ao contexto do Governo, onde os exemplos e práticas recomendadas estão diretamente relacionadas à problemas identificados em projetos passados ou atuais dentro da organização.

Os seguintes tópicos são cobertos pelo CAPS: documentação de requisitos, diagramas de caso de uso, diagramas de classes, diagramas de estados, diagramas de atividades, diagramas de seqüência e diagramas de componentes. Estes tópicos são continuamente revisados e melhoradas a partir do *feedback* dos projetos que são executados na organização.

4.1.2. Apoio a Testes de Software

Originalmente fora do escopo do projeto, a questão do teste de software acabou sendo incorporada no projeto como iniciativa de melhoria de qualidade de software. Um processo para guiar o teste de software foi elaborado para institucionalizar atividades mais criteriosas verificação dos produtos desenvolvidos. As práticas de teste de software elaboradas foram consolidadas no documento institucional denominado **Catálogo de Apoio à Testes de Software (CATS)**. Da mesma forma que o CAPS, o CATS possui definições e recomendações direcionadas aos problemas recorrentes em projetos passados ou atuais dentro da organização.

4.1.3. Padrões de Projeto e Grupo de Trabalho criados para melhoria de Arquitetura de Software

Em relação à problemas de arquitetura de software foram desenvolvidas duas ações principais: definição de um **Catálogo de Padrões de Arquitetura de Software (CPAS)** – ilustrado pela figura 1 - e constituição de um grupo de específico para discussão contínua de problemas e melhorias de Arquitetura de Software no contexto da PRODEPA. O CPAS é constituído por um conjunto de padrões arquiteturais que devem ser seguidos como referência na fase de projeto de software. Dos cerca de 116 padrões candidatos identificados na literatura, 13 padrões considerados essenciais ao domínio do governo estão catalogados na versão atual do CPAS. Este documento é disponibilizado aos projetistas de software da PRODEPA.

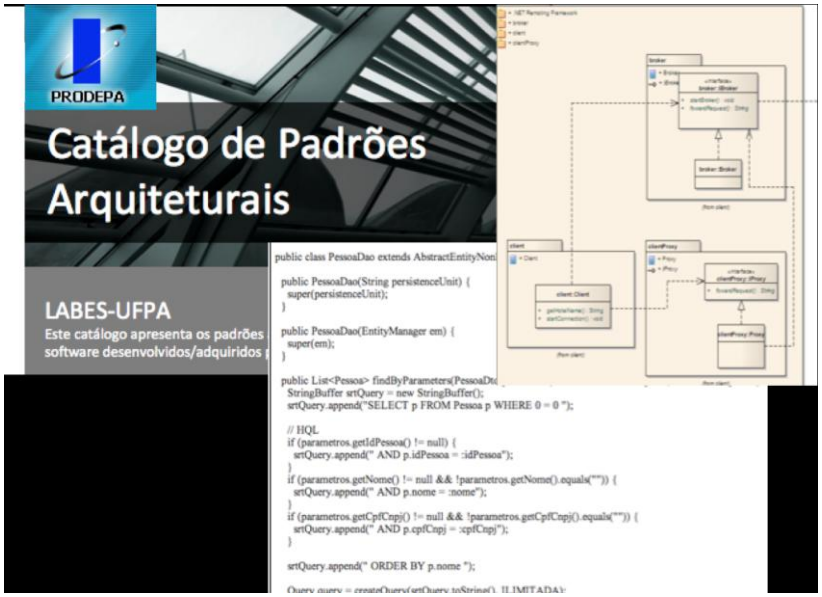


Figura 1 Ilustração do catálogo de padrões arquiteturais

Um aspecto importante e inovador deste catálogo consistiu na elaboração de exemplos da aplicação dos padrões de projeto usando trechos de sistemas desenvolvidos internamente na PRODEPA, com perfeito alinhamento com o contexto da empresa e das soluções corporativas do Governo do Estado.

O grupo de melhorias de arquitetura de software é constituído por membros de diferentes cargos dentro da organização (analistas de negócio, projetistas de software e programadores), onde são discutidos problemas atuais e perspectivas de melhoria na arquitetura de software da PRODEPA. A iniciativa de padronizar a arquitetura de software já existia antes do convênio de cooperação científica (evidenciada pela publicação de [Sizo 2009][1], e atualmente avança para flexibilização e otimização da arquitetura para contextos mais específicos (como integração com sistemas legados e sistemas de diferentes órgãos do governo).[8]

[1] Esta publicação foi premiada durante **I Workshop de Computação Aplicada em Governo Eletrônico**. Maiores detalhes em http://www.prodepa.pa.gov.br/index.php?q=node/411

4.1.4. Diretrizes para Aquisição de Software

Para atender a necessidade de consolidação de políticas e diretrizes para a área de TIC no que concerne ao desenvolvimento e aquisição de software foi desenvolvimento um documento contendo diretrizes gerais que foi encaminhada à COSIT[2]. Estas diretrizes devem ser utilizadas para dar suporte técnico à avaliação de pedidos de compra relativos à software no contexto do Estado do Pará, e estão alinhadas às políticas e diretrizes nacionais já aplicadas no governo federal (como exemplo os padrões e-MAG e e-PING).

4.2. Recursos humanos capacitados

Foram realizados 20 tutoriais de capacitação técnica em assuntos pertinentes ao projeto (com participação por volta de 100 colaboradores distribuídos entre as diferentes sessões técnicas), tanto para consultores do projeto quanto para colaboradores da PRODEPA. Os assuntos ministrados foram divididos em Formação em Análise de Software, Projeto de Software e Formação em Testes de Software. Além de sessões técnicas, foram realizadas atividades de *mentoring* continuamente aos membros da DDS, com a atuação diária de consultores *in loco* no ambiente da PRODEPA.

No contexto de análise de software foram ministrados tutoriais referentes aos seguintes tópicos relevantes ao projeto:

- UML 2.0;
- Requisitos não funcionas;
- Casos de Uso;
- Diagramas de Seqüência.

No contexto de projeto de software foram ministrados tutoriais referentes aos seguintes tópicos relevantes ao projeto:

- Processo PCP e ITP do MPS.BR;
- [9]Linguagens de Definição de Arquiteturas de Software;
- Definição de *Frameworks*;
- Arquitetura Orientada a Modelos (MDA);
- Métodos de Avaliação de Arquitetura de Software;
- Arquitetura Java EE;
- Padrões de Projeto [Gamma et AL 2000].

No contexto de teste de software foram ministrados tutoriais referentes aos seguintes tópicos relevantes ao projeto:

- Testes Unitários;
- Metodologia de teste segundo o processo definido no CATS.

4.6. Outros resultados

Como resultado geral do projeto foi realizado no dia 24 de fevereiro de 2010 o I Workshop sobre Desenvolvimento de Software na PRODEPA (intitulado DDS Day[3]).

Neste evento foram apresentados os principais resultados deste convênio de cooperação científica, além da apresentação de diferentes assuntos relacionados ao ambiente de desenvolvimento de software na PRODEPA. A figura 2 evidencia a participação dos membros da PRODEPA neste evento. Um dos resultados da institucionalização do processo de teste de software foi a submissão do relato de experiência “Uma Experiência de Implantação de Processo de Testes - Caso PRODEPA” ao Simpósio Brasileiro de Engenharia de Software 2010 [Martins 2010].

---

2 A COSIT (Comissão dos Sistemas de Informação e Telecomunicações) foi instituída no Estado do Pará por intermédio do Decreto nº 144, de 25 de abril de 2007 (D.O.E. nº 30.913, de 26/04/2007 e republicado no D.O.E. nº. 30926 de 16/05/2007) com o objetivo de racionalizar e melhorar os serviços relacionados aos Sistemas de Informação e Telecomunicações; desenvolver novas tecnologias.

3 Notícia sobre DDS Day está disponível nos links http://www.prodepa.pa.gov.br/index.php?q=node/355 e http://fenix.prodepa.gov.br/newsletter/?q=node/1505



Figura 2 Palestra do DDS Day na PRODEPA

Outro resultado relevante foi a experiência prática do projeto proporcionada aos alunos de doutorado, mestrado e graduação envolvidos. A equipe permanente do projeto é composta por sete membros, mas no total já integraram a equipe 11 alunos (1 de doutorado, 8 de mestrado e 2 de graduação).

5. Aplicabilidade dos Resultados

Os resultados deste projeto foram aplicados ao contexto de desenvolvimento de software da PRODEPA e de aquisição de software no Governo do Pará e submetidos a COSIT para homologação oficial para todos os órgãos públicos da Administração estadual. Acredita-se que os padrões arquiteturais catalogados durante o projeto possam ser utilizados em outras empresas do setor público como referência à problemas recorrentes de arquitetura de software e que estejam principalmente relacionados à interoperabilidade de aplicações. As diretrizes de aquisição de software desenvolvidas para serem utilizadas pela COSIT podem ser utilizadas por outras iniciativas governamentais semelhantes. Atualmente, os resultados do projeto têm circulação restrita a PRODEPA e Governo do Estado do Pará, embora a divulgação do conhecimento para o público externo esteja planejada para ocorrer em 2010.

6. Características Inovadoras

Ressalta-se que, na opinião dos autores, a maior inovação do projeto consiste na elaboração de catálogos e modelos de processo que foram adaptados para a realidade local e alinhados para os paradigmas de desenvolvimento de software da atualidade.

Deste modo, os exemplos apresentados nos catálogos propostos são de sistemas existentes ou de soluções para integração entre sistemas existentes ou em desenvolvimento. De forma mais específica: por exemplo, enquanto a descrição do padrão *Pipes and Filters* [Gamma et AL 2000] é tradicionalmente ilustrada com exemplos relacionados com o cenário do desenvolvimento de compiladores, no catálogo desenvolvido o exemplo foi escolhido a partir do cenário do processamento de dados trocados entre sistemas das várias bases de cálculo de tributos do Governo do Estado.

Este cenário é ilustrado pela figura 3 que mostra a relação de dependência entre a implementação dos serviços e os filtros correspondentes no contexto deste padrão de projeto.

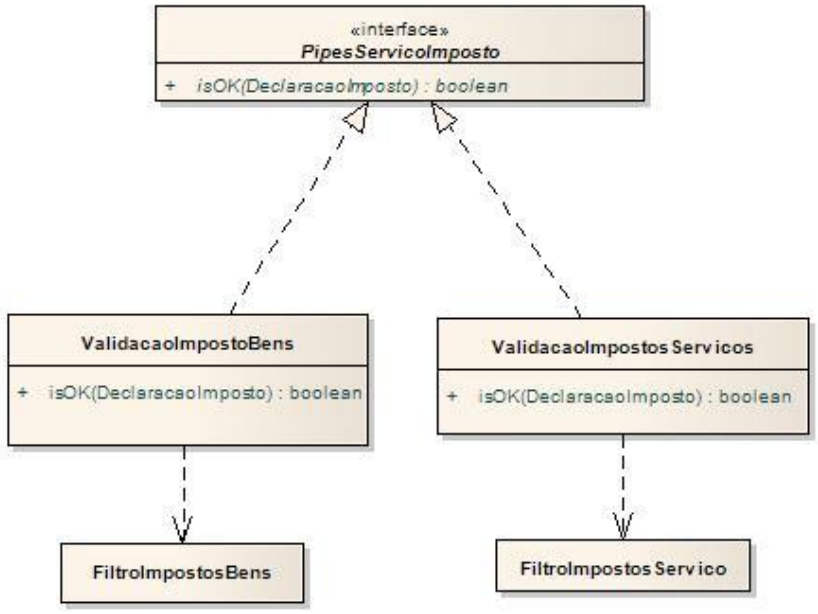


Figura 3 *Snapshot* do exemplo construído para ilustrar o padrão *Pipes and Filters*

7. Conclusão e Perspectivas Futuras

De acordo com os resultados discutidos neste relatório considera-se que os objetivos principais deste projeto de cooperação técnica foram alcançados. Embora uma avaliação quantitativa dos benefícios deste projeto ainda não tenham sido mensurada, a percepção qualitativa acerca das ações é significativa. Em entrevistas de colaboradores da PRODEPA pode-se perceber o ponto de vista dos beneficiados pelos resultados alcançados. Como exemplo, a citação a seguir menciona a visão da analista sênior Adriana Teles sobre os benefícios da documentação sobre arquitetura de software:

"Com a documentação, o conhecimento sobre a arquitetura, que era de domínio de algumas pessoas, tornou-se acessível a todos os colaboradores da empresa."

Do ponto de vista gerencial, pode-se destacar os benefícios das experiências práticas obtidas do projeto com a citação a seguir do Diretor de Desenvolvimento de Sistemas da PRODEPA, Sr. Claudio Martins: "A parceria entre a academia, representada pelo LABES/UFPA, e a PRODEPA, representa a aproximação entre o conhecimento da ciência, da prática, e da capacidade de trocar experiências que contribui para a maturidade organizacional da PRODEPA."

Como perspectivas futuras pode-se citar a edição e publicação de volumes impressos contendo os principais produtos desta cooperação técnica, tais como padrões arquiteturais, processos e metodologias associadas vislumbrada para o ano de 2010.

Espera-se que este material possa permitir o reuso das práticas e padrões desenvolvidos neste projeto em outros setores do que desenvolvem soluções em TI para o Governo em todo nível (Federal, Estadual e Municipal). Por fim, destaca-se esta perspectiva pela citação a seguir da analista sênior Adriana Teles da PRODEPA: "O Catálogo permite que o trabalho desenvolvido na PRODEPA seja propagado para outros órgãos do Estado, possibilitando o reuso de conceitos e práticas maduras e podendo contribuir, assim, para a padronização das soluções do Governo".

8. Referências Bibliográficas

e-PING. http://www.governoeletronico.gov.br/acoes-e-projetos/e-ping-padroes-deinteroperabilidade.

e-MAG. http://www.governoeletronico.gov.br/acoes-e-projetos/e-MAG.

Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. (GoF). Padrões de Projeto: Soluções reutilizáveis de Software Orientado a Objetos. Bookman . 2000.

Martins, Cláudio R. de L.; Reis, C. A. L.; Monteiro, R. W.; Sizo, A. M.; Vinicius, T.; Guimarães, L. F.; Sales, E. Uma Experiência de Implantação de Processo de Testes - Caso PRODEPA. Relato de Experiência submetido ao Simpósio Brasileiro de Engenharia de Software 2010.

Sizo, A. M., Xisto, A. N. T.; Fernandes, J. A.; Santos, L. R. D.; Martins, Cláudio R. de L. Arquitetura PRODEPA: Uma solução para criação de sistemas distribuídos.

Publicado no I Workshop de Computação Aplicada em Governo Eletrônico (WCGE 2009), Bento Gonçalves - RS. Softex (2010). MPS-BR. http://www.softex.br/mpsbr

Representação do Conhecimento Gerado no Processo de Desenvolvimento de Software Através de Storytelling

Leia F. Assis1,3, Carlos A M Pietrobon1,2

1 PPGEE - Pontifícia Universidade Católica de Minas Gerais (PUCMG) Av. Dom José Gaspar, 500 – Coração Eucarístico – 30.535-610 – Belo Horizonte – MG – Brasil

2 DECOM - Universidade Federal de Ouro Preto (UFOP) Campus Morro Cruzeiro – 34.400-000 – Ouro Preto – MG – Brasil

3 Centro Universitário Una (UNA)

leia.assis@gmail.com, capietro@pucminas.br, capietro@iceb.ufop.br

Abstract. *The software development process is a set of activities that makes intensive use of knowledge and information. However, a lot of them are lost during the development effort or to stay in the people's mind. If someone need them, it will be very difficult to access them. Some times, it will be impossible.*

One reason for this is because is difficult to record and recover all the process knowledge in a fast and easy way. This article shows an approach to sharing the software development knowledge telling histories. It's an old human ability applied to a new context: software process.

Resumo. *O desenvolvimento de software é um conjunto de atividades que faz uso intensivo de conhecimento e informação. Entretanto, muito destes são perdidos durante o esforço de desenvolvimento ou ficam na cabeça das pessoas. Se alguém necessitar deles, será muito difícil ou impossível obtê-los.*

Em parte isto acontece devido à dificuldade de se registrar e recuperar de forma rápida e intuitiva este conhecimento. Este artigo apresenta uma proposta para que o conhecimento relativo ao processo de desenvolvimento de software possa ser compartilhado através do mecanismo de contar histórias.

Esse mecanismo é uma velha habilidade humana aplicada a um novo contexto: processo de software.

1. Introdução

No processo utilizado no dia-a-dia das organizações para prover diferentes tipos de produtos e serviços, aplica-se e constrói-se constantemente um conhecimento que precisa ser preservado e disseminado. Na busca constante de aliar produtividade à qualidade torna-se essencial que o conhecimento existente nas organizações, proveniente da experiência de mercado e das pessoas que fazem ou fizeram parte do seu corpo, seja gerado, mantido, aumentado e propagado.

Entretanto, nem todas as informações e a experiência dos membros da organização estão registradas de maneira explícita. Existem atividades e procedimentos que envolvem modelos mentais, crenças e perspectivas dos indivíduos, que não sabem expressar de forma articulada como executam essas ações. Esse tipo de conhecimento é definido como conhecimento tácito, e por ser de difícil articulação, sua documentação em papéis, arquivos, relatórios etc. é muito complicada (Nonaka e Takeuchi, 1997).

Como aponta Leal (2004), um dos maiores desafios da gestão do conhecimento é realizar o processo de externalização, isto é, transformar o conhecimento tácito, localizado na mente dos indivíduos, em conhecimento explícito, articulável e que pode ser facilmente expressado. Uma das formas de se ajudar neste processo consiste em fornecer ferramental adequado para se registrar e difundir o conhecimento. No caso deste trabalho o recurso de contar histórias (*storytelling*) utilizando recursos multimídia.

Não diferente dos demais processos existentes nos variados ramos organizacionais, é indispensável gerir o conhecimento no processo de desenvolvimento de *software* (PDS). Martins (2004) argumenta ainda, que o desenvolvimento de *software* é uma atividade que possui um fluxo de trabalho relativamente variável, onde é difícil planejar antes de saber mais precisamente sobre o que vai ser produzido. O que pode ser facilitado se o conhecimento gerado durante o processo for preservado e disseminado.

O objetivo deste trabalho foi prover a representação do conhecimento com recursos que permitam construir, contar histórias (*storytelling*) sobre os fatos acontecidos durante todo o processo de desenvolvimento. O mecanismo de *storytelling* foi escolhido por ser simples, de fácil utilização e ter um alto potencial de representação do conhecimento. Basta lembrarmos que o ser humano sempre passou conhecimento sobre suas mais diversas atividades, contando histórias sobre as mesmas.

Adicionalmente, esta história será contada utilizando recursos multimídia, que enriquecerão a história contada, facilitando o seu entendimento. Para tanto, o artigo está organizado nas seguintes seções: (1) Introdução. (2) Processo de Desenvolvimento de *Software*. (3) Gestão do Conhecimento. (4) *Storytelling*. (5) História sobre Processo de *Software*. (6) Modelos. (7) ProcStory – Uma ferramenta que permite a construção de histórias. (8) Trabalhos relacionados. (9) Considerações Finais. (10) Referências.

2. Processo de Desenvolvimento de Software

Para Fuggetta (2000) um processo de *software* é definido como um conjunto coerente de políticas, estruturas organizacionais, tecnologias, procedimentos e artefatos que são necessários para conceber, desenvolver, entregar e manter um produto de *software*.

Em [Falbo, 1998] são explicitados os itens que compõem o processo de desenvolvimento de *software*. Para o contexto deste trabalho, estabelecemos como itens que compõem o processo, os recursos, as atividades e os artefatos. Estes podem ser entendidos como a realização de atividades (ações) com o objetivo de construir produtos (artefatos), a partir de recursos (pessoas, ferramentas, equipamentos, etc.) necessários à construção do *software*. Independentemente da forma (ferramentas, documentação, etc.) com que este processo é instanciado ou executado, ou do seu grau de qualidade (maturidade), temos uma realidade problemática: as pessoas envolvidas no desenvolvimento precisam compreender o processo e ter conhecimentos específicos para realizarem suas atividades.

Infelizmente, como aponta Souza e Nunes (2008) processos de *software* são complexos e envolvem um grande número de pessoas com tarefas especializadas, equipes multidisciplinares e cronogramas de longo prazo. Esta complexidade dificulta a percepção do processo como um todo por parte dos profissionais envolvidos, os quais necessitam possuir um mecanismo que permita a visualização dos diversos elementos e fases que constituem o processo.

Tanto a compreensão do processo quanto os conhecimentos necessários à execução das atividades, que são pré-requisitos para o andamento do trabalho, geralmente não são facilmente obtidos pelos envolvidos na atividade. Ferramentas de gestão do conhecimento podem apoiar o processo de desenvolvimento de *software* neste aspecto.

3. Gestão do Conhecimento

Para entender o tema gestão do conhecimento bem como sua devida importância é necessário tratar, no contexto deste trabalho, dos conceitos relacionados às diferenças entre dados, informação e conhecimento. Ao se buscar o significado destas palavras, percebemos que não há distinção clara entre as mesmas, principalmente decorrente do contexto e da área. Na Ciência da Computação a discussão sobre estes conceitos se relaciona principalmente com as formas de representação e armazenamento.

Setzer (1999) define bem as diferenças entre estes conceitos sob a perspectiva da Computação. Dado é como uma seqüência de símbolos quantificados ou quantificáveis.

Portanto, um texto é um dado. Mesmo se incompreensível para o leitor, qualquer texto constitui um dado ou uma seqüência de dados. Informação é uma abstração informal que está na mente de alguém, representando algo significativo para essa pessoa. Uma distinção fundamental entre dado e informação é que o primeiro é puramente sintático e a segunda contém necessariamente. Conhecimento é caracterizado como uma abstração interior, pessoal, de algo que foi experimentado, vivenciado, por alguém. Um nenê de alguns meses tem muito conhecimento (por exemplo, reconhece a mãe, sabe que chorando ganha comida, etc.). Mas não se pode dizer que ele tem informações, pois não associa conceitos. Conhecimento está associado com pragmática, isto é, relaciona-se com alguma coisa existente no "mundo real" do qual se tem uma experiência direta.

Neste sentido, representa-se o conhecimento através de informações, ou seja, utilizando conceitos relacionados a símbolos e palavras. Por sua vez as informações são representadas através de dados. Desta maneira, tanto o conhecimento quanto à informação são representados usando dados que fazem algum sentido quando são vistos e interpretados.

Os recursos visuais podem ser utilizados para se buscar informações, mas também para se exibir resultados pesquisados, como por exemplo, o Kartoo [Kartoo, 2007]. Neste sentido, o mecanismo de contar histórias foi escolhido por se tratar de uma maneira natural de representação do conhecimento. Além disso, recursos visuais diversos bem como de navegação serão utilizados no auxílio de construção das histórias, ajudando a absorção e compreensão da mesma.

O objeto de estudo no qual se apóia a gestão do conhecimento aborda os conhecimentos tácito e explícito. Segundo Archibugi e Lundvall (2003) o conhecimento tácito não pode ser explicitado formalmente ou ser facilmente disseminado, refere-se aos conhecimentos implícitos, como as habilidades acumuladas por um indivíduo, organização ou o conjunto, que compartilham de atividades e linguagem comum. O conhecimento explícito é armazenado, memorizado, pode ser transacionado e transferido, além de poder ser reutilizado, reproduzido e comercializado indefinidamente. (Cowan e Foray, 1997) De acordo com Nonaka e Takeuchi (1997) a criação de conhecimento nas organizações envolve as conversões dos diferentes tipos de conhecimento. Um dos maiores desafios da gestão do conhecimento é realizar o processo de externalização, transformando o conhecimento tácito que está na mente das pessoas em conhecimento explícito (Leal, 2004).

O objetivo deste trabalho foi desenvolver uma ferramenta que possibilitasse a conversão de conhecimento tácito em conhecimento explícito através de construção de histórias.

4. Storytelling

O ato de contar histórias faz parte da vida dos homens. E este recurso é uma forma simples e natural de transmitir conhecimento. Os seres humanos contam histórias para compartilhar experiências e casos. De forma similar, as organizações têm utilizado este recurso para possibilitar a transferência do conhecimento do seu corpo organizacional.

Toda organização, à medida que evolui, acumula uma série de experiências, casos e aprendizados associados à experiência adquirida pelas equipes e líderes. Porém, os artefatos gerados durante um projeto de desenvolvimento não trazem consigo os contextos e histórias pessoais que ajudaram na evolução da organização. Histórias são ricas em todos estes elementos e têm permeado a vida das organizações. [Terra, 2006] Eventos são partes de uma história. Por exemplo, uma reunião, uma viagem, a assinatura de um contrato. Para Valle, Prinz e Borges (2002) uma narração de uma cadeia de eventos contada ou escrita em prosa e verso é uma história. Já no dicionário Michaelis a palavra história é definida como uma narração de acontecimentos, de ações, em geral cronologicamente dispostos [História, 2008]. Dentro da organização, contar histórias pode se tornar uma ferramenta eficaz de transmissão de conhecimento. Trata-se de uma antiga habilidade humana aplicada a um novo contexto: o contexto da gestão do conhecimento (Valle et al., 2003).

As histórias são capazes de simplificar e ajudar a transmitir um conhecimento complexo. A *Harvard Business School* tradicionalmente ensina assuntos complicados através do uso de narrativas, sob a forma de estudos de caso (cases) (Ruggles, 1999).

Neste sentido, percebe-se em *storytelling* um grande potencial para gestão do conhecimento gerado durante o processo de desenvolvimento de *software*.

5. História sobre Processo de Software

A solução apresentada para manipular conhecimento, se baseia na construção de histórias. Entretanto, esta história precisa ter uma estrutura narrativa, e precisa ter recursos para descrever fatos sobre os processos de *software*. Com relação à estrutura, Gancho (1991) define os cinco principais elementos que são componentes de uma narração: (1) **Enredo:** Conjunto de fatos que de uma história. (2) **Personagem:** Ser fictício ou não, que participa do enredo. (3) **Tempo:** Pode se referir tanto à época em que se passa a história quanto à duração da mesma. (4) **Espaço:** Local onde se passa. (5) **Narrador:** Autor da história, ou quem conta.

Quanto à descrição dos fatos ocorridos durante o desenvolvimento, o *stakehoder* deverá utilizar termos definidos na ontologia do processo e ter recursos visuais que representem os elementos desta ontologia para narrar a história. Por exemplo, uma imagem que mostre um grupo em torno de uma mesa significaria uma reunião.

Vale lembrar que, segundo Chandrasekaran et al. (1999), uma ontologia pode ser entendida como uma representação do vocabulário de algum domínio ou problema. Mas não é o vocabulário que qualifica uma ontologia, e sim as conceituações que os termos do vocabulário pretendem capturar.

Lage (1993) afirma que a espinha dorsal da narrativa é a organização dos eventos em seqüências. Uma história pode ser dividida em eventos e o propósito da narrativa definirá a forma pela qual serão apresentados esses eventos. A reunião descrita anteriormente pode ser um evento de um enredo maior.

Considerando, que uma história é composta por eventos, e também têm os componentes descritos anteriormente: enredo, personagens, tempo, espaço e autor, os mecanismos para construção das mesmas deverão alocar esta estrutura. Na verdade, a estrutura é criada conforme o narrador conta a história. Ele é quem determina os eventos e a relação entre eles. Por exemplo, antes da reunião com o cliente, poderia ter sido narrado outro evento que seria uma reunião de fechamento da proposta comercial com personagens pertencentes à organização desenvolvedora.

Como o conhecimento está muito relacionado com a experiência, vivência, contar histórias provê um grande potencial para representação de conhecimento e conversão do conhecimento tácito para explícito. Porém, aproveitando-se do potencial da narrativa, ou seja, do mecanismo de contar história para retratar bem uma situação ocorrida ou uma experiência vivida, é importante ainda, elaborar uma história que não seja chata, que não seja difícil de ser visualizada. Desta maneira, a ferramenta proposta permite usar recursos de cores, textos, *links*, vídeos, sons, figuras e áudio para compor os diversos eventos na ordem mais apropriada para contar uma história.

Mas para gerenciar o conhecimento proveniente do processo de desenvolvimento de *software* é necessário vincular as histórias contadas a este processo permitindo a realização de pesquisas em torno de diferentes itens de conhecimento relacionados às atividades de desenvolvimento de acordo com a necessidade daqueles que estão as realizando. Assim, no momento em que história é criada as informações relativas ao processo na

qual a mesma se refere devem ser fornecidas. Estas informações são usadas para a indexação da mesma e posterior localização. Assim uma história poderá ser localizada e visualizada, por projeto, por item de processo (atividade, recurso, artefato), por autor e por personagens e por palavras-chave diversas, que também são cadastradas para uma história registrada.

A ferramenta apresentada utiliza a junção de *storytelling* com o processo de desenvolvimento, provendo a gestão do conhecimento no processo de desenvolvimento de *software*.

6. Modelos

Conforme pode ser observado no diagrama de classes ilustrado na figura 1, um item de conhecimento pode ser do tipo **Processo de *Software*** ou pode não ter relação com o processo de desenvolvimento de software (**Independente**). Para o caso da história ser sobre o processo de *software*, poderá ser tanto sobre a definição de um processo quanto também sobre a execução de um processo (projetos usando processos definidos). Os demais itens do processo de *software* estão ligados às atividades que fazem parte do processo. Uma atividade possui sub-atividades, que usam artefatos (entrada no diagrama) e geram artefatos (saída no diagrama).

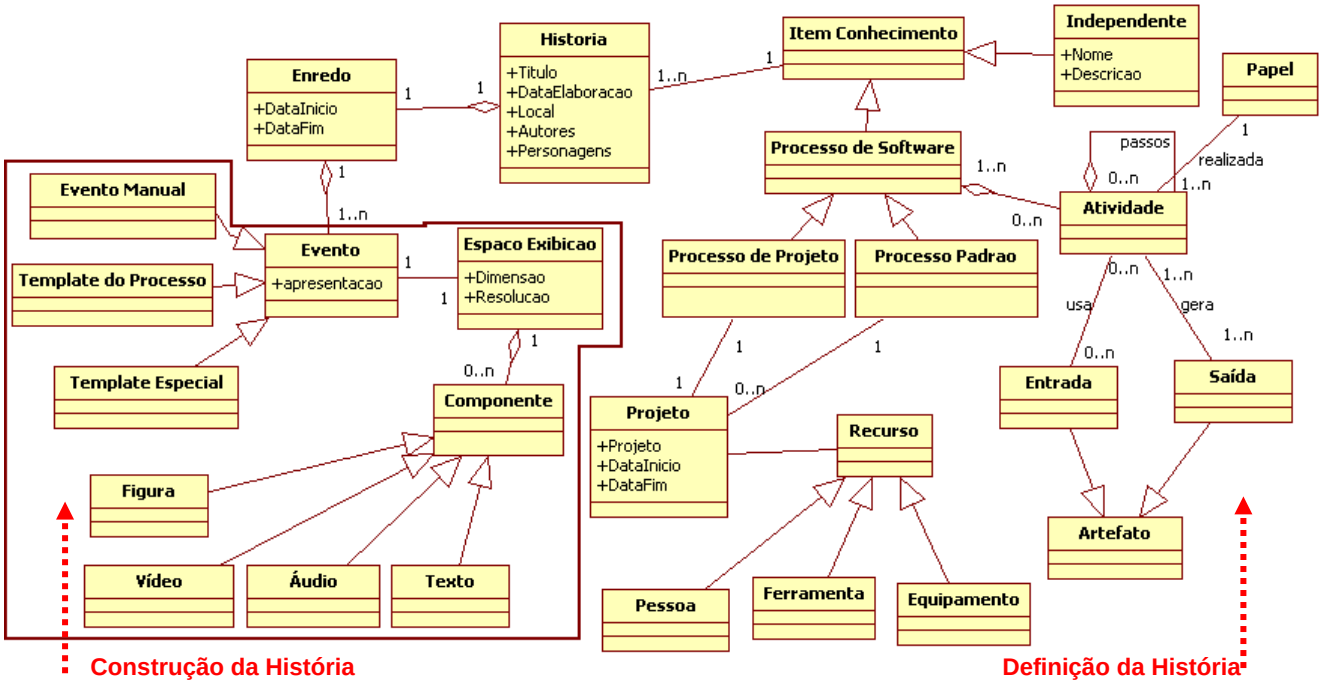


Figura 1. Diagrama de Classes da história relacionada ao desenvolvimento.

Qualquer atividade ou acontecimento relacionado ao processo de *software* envolvem um ou mais interessados ("*stakeholders*"). Estes envolvidos podem manifestar este conhecimento através de uma ou mais histórias relacionados a itens de conhecimento diversos sobre o desenvolvimento de *software*.

Por outro lado, cada item de conhecimento relacionado ao processo de desenvolvimento pode apresentar enredos variados, construídos por autores distintos.

Cada enredo é um conjunto de fatos ou eventos que faz parte de uma história contada pelo autor e arrumada de uma forma conveniente para ser apresentada.

O diagrama de classes apresentado mostra a estrutura de uma história que poderá ser contada sobre qualquer item de conhecimento relacionado ao desenvolvimento de *software*. Esta terá uma autoria, possuirá personagens, data, local e ainda um enredo. O enredo é composto pelos eventos que a compõe. Cada evento é uma parte da história que é apresentada por tela (**Espaço Exibição**) e que poderá possuir diversos elementos.

Cada elemento utilizado para construir o evento, utiliza recursos multimídia variados, como áudio, vídeo, imagens e texto.

Vale lembrar que apesar do registro das histórias serem realizados após a construção das mesmas, e após estas serem salvas, a publicação destas, bem como a disponibilização das mesmas para os demais envolvidos neste processo, dependerá da aprovação da mesma por um gerente. A figura 2 mostra este processo.

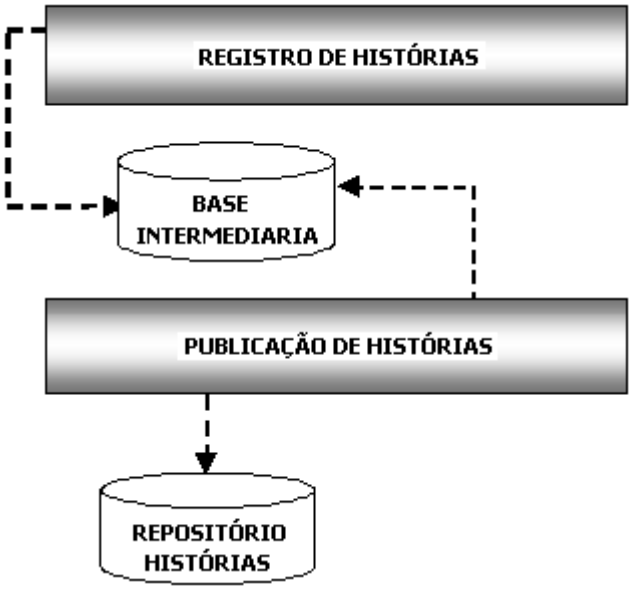


Figura 2. Publicação das Histórias.

7. ProcStory – Uma ferramenta que permite a construç ão de histórias

A ferramenta *ProcStory* foi construída para possibilitar a gestão do conhecimento no processo de desenvolvimento de *software* através da construção de histórias. Esta ferramenta está integrada ao Ambiente *Discovery* [Guedes, 2007] que é um ambiente *web* de visualização de conhecimento de processo de *software*.

A figura 3 ilustra a tela da ferramenta para quem conta (monta) uma história. Ao lado esquerdo da tela, são vistas as diferentes ações que o usuário pode realizar, referentes tanto a construção quanto a consulta das histórias.

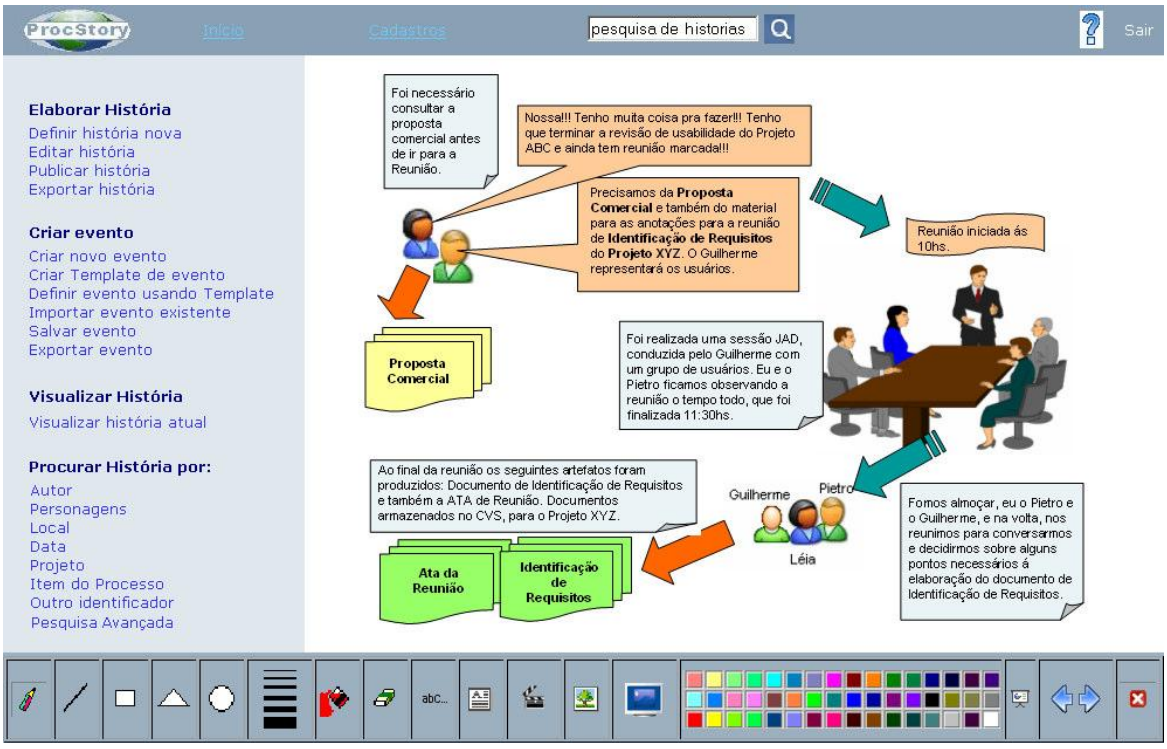


Figura 3. Tela principal da Ferramenta ProcStory.

A barra de ferramentas, localizada na parte inferior da tela, contém os recursos elementares de visualização necessários para se construir a história, tais como filmes, fotos, sons, figuras, etc. Todos estes elementos são

selecionados, posicionados na área de edição e instanciados apropriadamente, de forma a montar a história. No momento em que se inicia a criação da história, essa área é apresentada em branco.

Para entender melhor como os elementos construtores são adicionados a história, a seguir exemplificamos como ocorre a inclusão do elemento rótulo (um texto simples e não um texto grande) na história sendo construída. Uma vez que um rótulo foi selecionado na barra de ferramentas da figura 3, é apresentada a tela vista na figura 4.

Para associar uma imagem, um som, um vídeo ou um *hiperlink* ao rótulo, basta clicar em um dos botões **imagem..., som..., vídeo...** ou **hiperlink...**, respectivamente. Ao se clicar em um destes botões, a tela é automaticamente expandida para que se possa selecionar a fonte multimídia que será associada ao rótulo inserido. Os demais elementos que podem ser utilizados para construção da história também são inseridos de forma similar.



Figura 4. Tela Editar Rótulo.

Para que a história possa ser contada, ela precisa ser criada. Isto é feito utilizando a tela apresentada na figura 5. Nela são inseridas informações relativas à definição da história e também à associação da história com o processo e projeto de desenvolvimento de *software*.

A estruturação (organização entre os elementos) da história é construída conforme o Contador escreve a mesma. Alguns trechos podem ser mais detalhados, se necessário. Neste caso, outras telas poderão ser criadas para navegação. Neste caso faz-se uso de *links*. Por exemplo, se a figura “ata de reunião” for clicado, tem-se acesso a ata da reunião em questão. Caso uma história tenha sido contada sobre esta ata, sua história pode ser acessada também.

Após as devidas informações para a definição de uma história serem inseridas, uma tela em branco é exibida e a história poderá ser construída utilizando os elementos e recursos contidos na barra de ferramentas. Outras telas poderão ser adicionadas na construção da história através da definição de novos eventos para aquela história. É criada uma tela para cada evento narrado que compõe a história. A navegação pela história pode acontecer de duas maneiras: indo para o próximo evento sequencialmente que foi encadeado quando a história foi criada ou seguindo um *link* que foi adicionado a um evento. A navegação na história por *links* permite que a história possa ser visualizada numa seqüência alternativa ou com diferentes níveis de profundidade. Caso não sejam utilizados os *links* para navegação na história, a mesma será exibida, de acordo com a ordem em que os eventos foram criados. Neste caso, a primeira a ser exibida é a tela principal, a que contém o primeiro evento. Depois é exibida a tela correspondente ao próximo evento criado e posteriormente os demais eventos criados na respectiva ordem.

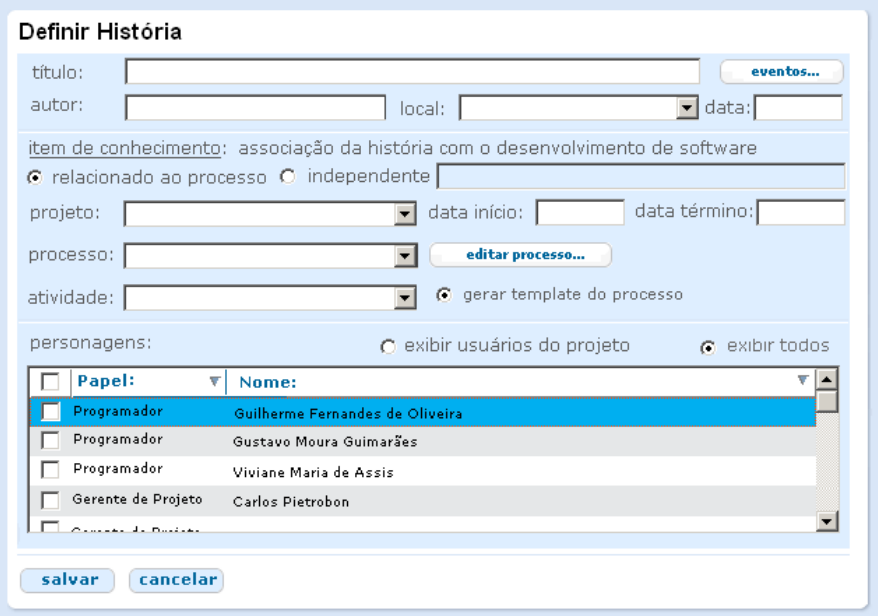


Figura 5. Tela Definir Nova História.

Desta maneira a história é construída, através de telas. E recursos variados, disponibilizados na barra de ferramenta da figura 3 poderão ser utilizados em cada tela, para construção da história e navegação nas mesmas.

Quando todas as histórias estiverem prontas, podem ser acessadas pelo Gerente de Conhecimento que realiza uma ou mais das atividades seguintes: (a) Monta a história final oficial. (b) Cria histórias apropriadas e adaptadas a cada categoria de usuário (por exemplo, histórias que são importantes para o gerente e as que são relevantes para o programador). (c) Publica uma história recebida (d) Constrói histórias apropriadas a necessidades.

8. Trabalhos relacionados

Pereira (2003) desenvolveu um trabalho com o objetivo de iniciar a discussão sobre um novo modelo de Engenharia de *Software* que fosse orientado á produção de conhecimento, distinto dos modelos atuais baseado nos diferentes tipos de conversões do conhecimento inseridos no processo de desenvolvimento de *software*. Assim, para prover a gestão do conhecimento neste processo é necessário prover mecanismos que permitam as conversões dos diferentes tipos de conhecimento. Como por exemplo, do conhecimento tácito para o explícito e do explícito para o tácito.

A Estação TABA é um ambiente de desenvolvimento de *software* que apóia a execução das atividades de desenvolvimento, utilizando ferramentas e repositórios que contém informações sobre a execução do processo de software. A partir de um processo padrão, ocorre o instanciamento para o processo específico de uma organização. Neste contexto, Montoni (2003), desenvolveu uma ferramenta de aquisição de conhecimento no processo de desenvolvimento de *software* denominada *Acknowledge*. Seu principal objetivo é apoiar a execução do processo de aquisição de conhecimento em atividades de engenharia de *software*, sendo possível registrar diferentes tipos de conhecimento que podem ser independentes ou não da execução de um processo.

Falbo (1998) desenvolveu um trabalho (ODE-*Ontology based software development environment*) com a meta de integrar conhecimento no processo de desenvolvimento de *software*, utilizando uma abordagem ontológica, onde se tem conceitos e relações, definições, propriedades e restrições em forma de axiomas sobre processos de *software*.

Tanto o TABA quanto o ODE trazem a preocupação em prover um ambiente de desenvolvimento de *software*, que facilite e auxilie as atividades de desenvolvimento considerando a gestão do conhecimento. Porém não usam a técnica de *storytelling*, e nem diferentes maneiras de representação do conhecimento.

Leal (2004) desenvolveu um trabalho usando *storytelling* que é a narrativa ou ato de contar histórias como mecanismo de disseminação do conhecimento, porém utilizando apenas recursos textuais.

Outro trabalho usando o mecanismo de contar histórias foi o **Storytelling Alice**, desenvolvido por Caitlin Kelleher como parte de seu trabalho de doutorado em Ciência da Computação. Alice é uma ferramenta que cria animações para contar histórias que pode ser utilizada para o ensino de introdução à computação. Usa gráficos 3D e uma interface mais envolvente, menos frustrante para a primeira experiência de programação.

Utiliza um mecanismo de criar animações para contar histórias. Permite a criação de filmes adicionando objetos às cenas, que fazem parte do filme, e cada objeto é manipulado, conforme uma programação orientada

a objetos. A cada objeto inserido na cena, é possível manipular as propriedades, métodos, funções, e, além disso, novas variáveis e parâmetros também podem ser criados. (Alice, 2008).

Por fim tem-se o *Impulse* [Burkhard 2005], um *framework* genérico para visualização de conhecimento em processos de negócio, baseado em conhecimento, formas de visualização, receptor do conhecimento e objetivo da visualização. O *ProcStory* implementa os três primeiros conceitos uma vez que está voltado para a descrição visual de conhecimentos apropriados as diversas categorias de usuário em processos de *software*.

Já o presente trabalho, apresentado neste artigo, permite a construção de histórias vinculadas ao processo de desenvolvimento de *software*, fazendo uso de recursos multimídia para ilustrar a mesma, que desta forma será preservada, mantida, disseminada, recuperada e utilizada.

9. Considerações Finais

A principal contribuição desta pesquisa foi desenvolver uma abordagem que permita registrar através de representações o conhecimento relacionado ao processo de desenvolvimento de software por meio da técnica de *storytelling*, conhecida há milênios, mas não aplicada à área de desenvolvimento de software e nem na qualidade de software. A construção de histórias é baseada na estrutura do texto narrativo e permite a utilização de recursos textuais, sonoros e visuais, que são associados ao processo de desenvolvimento de *software*, aos processos de qualidade ou a partes deles.

Além disso, o desenvolvimento do presente trabalho incentiva novas pesquisas sobre *storytelling* apoiando a representação do conhecimento gerado em qualquer tipo de organização e processo, como também nas organizações de desenvolvimento de *software*, e contribuindo para o processo de aprendizado, tanto na área acadêmica quanto em qualquer outra área.

Este trabalho apresenta uma ferramenta para contar histórias sobre processos de *software*. Seu objetivo é facilitar a troca de conhecimento entre os diversos interessados no desenvolvimento de *software* em um projeto e, num escopo mais abrangente, na organização como um todo. Esta história faz uso de recursos multimídia que permitem as pessoas escolherem as formas mais apropriadas de expressar os fatos que vivenciou e desta forma facilitar a transferência do conhecimento explícito, bem como o tácito.

10. Referências

Alice. 2008. www.alice.org. Carnegie Mellon University. [Online] *Alice is made freely available as a public service*, 2008. [Citado em: 10 de Agosto de 2008.] *An Educational Software that teaches students computer programming in a 3D environment*.

Archibugi, D. e Lundvall, B. 2003. *The Globalizing Learning Economy*. s.l. : Oxford University Press, 2003.

Burkhard, R. A. (2005), "*Impulse: Using Knowledge Visualization in Business Process Oriented Knowledge Infrastructures*", *Journal of Universal Knowledge Management*, Vol. 0, No. 2, pp. 170-188.

Chandrasekaran, B., Josephson, J. R. e R., Benjamins V. 1999. What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems. 1999.

Cowan R., Foray D. (1997), "*Tacit knowledge and information: that the dynamics of codification are endogeneous*", *Working Paper*, IRIS-IMRI, *University of Paris Dauphine*, 25p.

Falbo, R. A. (1998), "Integração de Conhecimento em um Ambiente de Desenvolvimento de Software", Tese de Doutorado. Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ.

Fuggetta, A. "*Software Process: A Roadmap*", *in Proc. of The Future of Software Engineering*, ICSE'2000, *Limerick, Ireland*.

Gancho, C.V. (1991), "Como analisar narrativas", São Paulo: Ática.

Guedes, F. C. (2007), "Glifo para Verificação da Utilização de Processos de Requisitos por meio da Rastreabilidade como apoio à melhoria de Qualidade de *Software"*, submetido ao SBQS 2007.

História (2008), "Dicionário Eletrônico Michaelis", http://michaelis.uol.com.br/moderno/portugues. Último acesso em março de 2008.

Judelman, G. B. (2204), "*Knowledge Visualization: Problems and Principles for Mapping the Knowledge Space*", MSC *Thesis Dissertation*, *Univerity of Lübeck, Germany, June.*

Kartoo S. A. (2007), "Metamotor de pesquisa de Informação Web", http://www.kartoo.com, Último acesso em março de 2008.

Leal, R. P. (2004) "Técnica de *Groupstorytelling* aplicada à gestão do conhecimento", Dissertação de Mestrado. Rio de Janeiro. UFRJ.

Martins, J. C. C. (2005) "Gerenciando projetos de desenvolvimento de *software* com PMI, RUP e UML", Rio de Janeiro, Brasport.

Montoni, M.A. (2003), “Aquisição de Conhecimento: Uma Aplicação no Processo de Desenvolvimento de *Software*”, Dissertação de Mestrado, UFRJ.

Nonaka, I. e Takeuchi, H. (1997) ”Criação de Conhecimento na Empresa”, Rio de Janeiro. Campus.

Pereira, M. A. M. (2003), “Produção e Disseminação de Conhecimento na Engenharia de *Software*”, Dissertação de Mestrado, Pontifícia Universidade Católica de Campinas, Programa de Pós-Graduação em Ciência da Informação.

Ruggles, R. (1999), “*The role of stories in knowledge management*”. *In: Journal of Storytelling and Business Excellence*. *Storytelling Foundation International*.

Setzer, V. W. (1999) “Dado, Informação e Competência”, Revista Eletrônica sobre Ciência da Informação Datagrama Zero,USP, Novembro.

Souza, A. L. R. e Nunes, D. J. (2008) “Uma ferramenta para Apoio a Visualização e Representação de Modelos de Processo de Software”, http://www.inf.ufrgs.br/prosoft/rsrc/Prosoft/PublicatedWork20050908205225/LincolnArtigoSemAcad.PDF, UFRGS, Março.

Terra, J. C. C. (2006), “*Storytelling* como ferramenta de gestão”, http://www.terraforum.com.br/sites/terraforum/Biblioteca/Storytelling%20como%20ferramenta%20de%20gest%C3%A3o.pdf, Terra Fórum Consultores, São Paulo.

Valle, C. et al. (2003), “*Group storytelling to support tacit knowledge externalization*”. *In: Proceedings of the 10th International Conference on Human – Computer Interaction. Crete, Grécia. Mahwah, New Jersey, USA: Lawrence Erlbaum Associates, Publishers*, 2003, v. 4, p. 1218-1222.

Valle, C., Prinz, W. e Borges, M.R.S.(2002), “*Generation of group storytelling in postdecision implementation process*.” *In: Proceedings of the* 7 the *International Conference on* CSCWD. Rio de Janeiro, 2002. p. 361-367.

Sensedia Dashboard

Rafael Navarro1-2, José Vahl1, Marcílio Oliveira1

1 Sensedia – Polis de Tecnologia Rodovia Campinas-Mogi Mirim (SP 340) Km 118,5 - Prédio 9C – 13086-902 - Campinas – SP – Brasil

2 Instituto de Computação – Universidade Estadual de Campinas (UNICAMP) Caixa Postal 6176 – 13.084-971 – Campinas – SP – Brasil

rafael.navarro@sensedia.com, jose.vahl@sensedia.com, marcilio.oliveira@sensedia.com

Resumo. *Esse projeto tem o objetivo de criar uma solução de software para auxiliar as atividades de acompanhamento da saúde de ativos de software (projetos de desenvolvimento, componentes, serviços, etc.) baseado em diversas métricas de qualidade configuráveis e extensíveis. O estabelecimento de um conjunto inicial e representativo de métricas foi fruto de uma análise empírica das necessidades da Ci&T. A Ci&T é uma integradora de software de médio porte e o levantamento levou em consideração seu centro de desenvolvimento em Campinas, que conta com mais de 400 desenvolvedores e mais de 30 projetos em execução. Os benefícios do projeto Dashboard incluem o aumento da qualidade no software produzido, a avaliação precoce de ativos não-qualificados, o monitoramento de componentes e serviços legados, entre outros.*

Abstract. *The goal of this project is to create a software solution to help monitor the health of software development assets (projects, components, services, etc). The process is based on configurable and extensible quality metrics. The use of a representative group of metrics resulted from an empirical analysis of Ci&T´s needs, a medium size software development company. The analysis took into consideration the main development center of the company, based in Campinas with 400 developers and over 30 projects. The benefits of the Dashboard project includes: quality increase, early evaluation of non-qualified assets, monitoring of services and components available, etc.*

1. Introdução

A busca incessante por maior produção, com melhor qualidade e menor custo vêm norteando os esforços de muitas organizações. A realidade de empresas de desenvolvimento de software não é diferente, em todo mundo, as áreas de TI buscam cada vez mais agilidade, economia e inovação tecnológica, que constituem o tripé da competitividade da indústria de software. E, dentre inúmeras tecnologias e soluções que aparecem todos os dias, grande parte dos especialistas aposta no reuso como forma de ganhar agilidade e otimizar investimentos.

O reuso pode ser considerado uma denominação “genérica”, envolvendo uma variedade de técnicas utilizadas desde as etapas de design até a implementação. As técnicas de reuso ainda estão em processo de amadurecimento, mas todas assumem como premissa a re-utilização de artefatos cuja qualidade já foi assegurada.

Precisar o termo qualidade de software é uma tarefa difícil, uma vez que esse é um conceito relativo e cercado de variáveis. Para muitas pessoas, qualidade de um software representa apenas um sistema que satisfaz a necessidade de um cliente, outras acreditam que seja apenas um código de qualidade. Segundo Pressman, qualidade de software é a conformidade entre: requisitos funcionais e não funcionais que têm sido explicitamente declarados, padrões de desenvolvimento que tenham sido claramente documentados e características implicitamente esperadas de todo software a ser desenvolvido [Pressman01]. A conformidade destas atividades abordadas por Pressman são essenciais para que todo o processo de desenvolvimento de software contribua para qualidade do software e para isso temos que adicionar algumas atividades de acompanhamento e disciplina sobre tal processo, algumas delas são:

- **Acompanhamento de ativos de software em desenvolvimento:** para que o setor responsável tenha um acompanhamento do progresso do desenvolvimento, assim sendo possível tomar decisões eficazes em caso de desvio do planejamento.
- **Acompanhamento da qualidade de desenvolvimento:** acompanhar o desenvolvimento do projeto e verificar métricas de eficácia do processo e dos artefatos gerados.

Os itens citados acima são justamente o foco central do projeto apresentado neste artigo e indicam a **saúde de um ativo de software**. Esse conceito será utilizado no decorrer do texto e engloba a avaliação da qualidade arquitetural de ativos e representação dos resultados consolidados de forma facilitada para servir de insumo à tomada de decisões.

O Sensedia Dashboard originou-se a partir da identificação das necessidades de uma fábrica de software, que reconheceu a necessidade de avaliação contínua da saúde de seus projetos e componentes.

2. Objetivo e justificativa

O **Sensedia Dashboard** é uma solução para analisar a qualidade e a saúde de ativos de software. A solução se baseia em um conjunto extensível de indicadores de qualidade que incluem regras de estilo, violação de dependências e testes unitários para gerar um painel de semáforos indicando a saúde arquitetural dos projetos e componentes.

O painel de semáforos é baseado em métricas configuráveis ajustadas para atender as especificidades de cada projeto e permitem que gerentes possam acompanhar a qualidade técnica e o desempenho dos times de desenvolvimento, sejam internos ou externos.

A implementação é baseada na **integração de um conjunto de ferramentas que, juntas, sejam capazes de representar a saúde de ativos mediante a execução de uma série de indicadores**. Os ativos podem ser analisados em diversos cenários, como: ativos em parques de aplicações legadas; projetos em desenvolvimento; e componentes de software armazenados em repositórios digitais. Os resultados gerados são centralizados, viabilizando o acompanhamento gerencial da qualidade do código gerado e antecipando as tomadas de decisão concernentes a qualidade arquitetural.

A avaliação da qualidade de ativos durante o processo de desenvolvimento, de maneira sistemática e automatizada, na interfere na produtividade das equipes, uma vez que roda em servidores separados, e oferece uma maturidade de construção de software diferenciada. As avaliações automáticas baseiam-se nos níveis de qualidade previamente estabelecidos por meio da configuração dos indicadores técnicos.

Além disso, avaliação de ativos já construídos é importante para atingir um nicho completo das empresas de desenvolvimento que buscam o reuso para evitar o re-trabalho no desenvolvimento de novos projetos, capitalizando trabalhos anteriores, fazendo com que soluções já desenvolvidas sejam imediatamente implementadas em novos contextos. Desta forma têm-se melhores produtos em um menor intervalo de tempo, com redução de custos de manutenção, pois as partes do sistema são independentes, e facilitando a inclusão de novas funcionalidades. Além disso, a qualidade aumenta devido ao reuso de componentes previamente analisados e qualificados [D´Souza 99][Jacobson97]. Para este fim, a solução proposta neste artigo integra-se com ferramentas de descoberta de ativos e serviços como o Sensedia Discoverer, produto originado do projeto IACS (Identificador Automático de Componentes de Software, segundo lugar no Prêmio Dorgival Brandão, edição 2007) [Oliveira07], mecanismo capaz de reconhecer em parques de aplicações já desenvolvidas, grupos de artefatos que componham ativos reutilizáveis de software, serviços web e componentes e com biblioteca de ativos como é o caso do Sensedia Repository, principal ferramenta nacional de gerenciamento de ativos digitais (prêmio B2B de Qualidade em 2005 e 2006 e prêmio IBM Best Choice em 2004 e 2005) [Bacili06]. Ainda com a mesma finalidade o Sensedia Dashboard potencializa a identificação de ativos bem qualificados para compartilhamento na RCCS (Rede de Compartilhamento de Componentes de Software, terceiro lugar no Prêmio Dorgival Brandão, edição 2006) [Oliveira05].

Para potencializar o uso da ferramenta e possibilitar que as análises dos resultados providos pelo Sensedia Dashboard possam trazer o valor esperado, foi desenvolvida uma arquitetura independente de plataforma a todos os módulos do software.

3. Metodologia de execução

O projeto foi desenvolvido visando uma arquitetura modular, com módulos integrados, fornecendo uma solução completa de análise e acompanhamento contínuo da saúde de ativos. Além da integração com o processo de desenvolvimento, o projeto prevê um escopo maior, que envolve o cenário de descoberta de ativos e SOA, mediante integração com ferramentas de descoberta de serviço e componentes e repositório de ativos digitais, criando assim o Sensedia Suite representado pela Figura 1.

Algumas premissas foram consideradas para desenvolver uma solução que, além de avaliar a saúde de ativos pontualmente, seja capaz de realizar esta tarefa em diferentes cenários de uma empresa. Os principais cenários são:

- **Acompanhamento sistemático de projetos** em desenvolvimento baseado em indicadores de qualidade customizados para cada projeto. A freqüência de geração dos relatórios de qualidade é configurável, assim como os vários níveis de detalhamento das informações. Desta forma, os relatórios são úteis a diversos públicos, desde as equipes de desenvolvimento até a gerência e os executivos, possibilitando que as ações de melhoria da qualidade sejam tomadas durante o desenvolvimento, antecipando problemas que poderiam ser encontrados apenas nas etapas de teste ou entrega do projeto.

- **Avaliação de serviços e componentes legados** descobertos automaticamente pelo Sensedia Discoverer. Após a identificação de um potencial componente ou serviço feita durante uma varredura, o Sensedia Dashboard assume a responsabilidade de mostrar graficamente a saúde do ativo encontrado. Essa avaliação servirá como insumo para os arquitetos avaliarem precocemente de possíveis problemas

com o legado, eliminando assim candidatos não qualificados que poderiam vir a integrar a biblioteca de ativos para reuso.

- **Visão arquitetural crítica de ativos** que já fazem parte da biblioteca digital mantida pela ferramenta Sensedia Repository. A integração com esta ferramenta fornece uma avaliação mais detalhada dos ativos cadastrados no repositório, permitindo a identificação de ativos com potencial para serem distribuídos por redes de compartilhamento de componentes de software (RCCS).

- **Integração com ambientes de desenvolvimento** para garantir que todos os conteúdos produzidos durante o desenvolvimento possam ser analisados e validados pelo próprio desenvolvedor antes de serem versionados, antecipando a identificação de problemas e aumentando a legibilidade do código, garantindo assim melhor facilidade de evolução e conseqüentemente redução no custo de manutenção do projeto desenvolvido.



Figura 1. Integração do Sensedia Dashboard

3.1. Funcionalidades principais do Sensedia Dashboard

Foi vislumbrado um ambiente de avaliação da saúde de software que pudesse, de fato, envolver todos que participam de alguma forma na sua qualidade e que incentive cada vez mais a avaliação contínua de ativos utilizados ou produzidos.

- **Gerenciamento de ativos:** As configurações dos ativos são feitas de acordo com a localização do ativo (sistema de arquivos local, controladores de versão, bibliotecas de ativos, etc.), e sua relevância é determinada segundo sua importância para a métrica consolidada que permite o acompanhamento geral do conjunto de ativos analisados. Para cada ativo é definido o conjunto de indicadores de qualidade e a relevância de cada um deles segundo o ativo analisado. Esta abordagem será discutida mais adiante neste mesmo tópico.

- **Configurações genéricas para acompanhamento em tempo de desenvolvimento:** Para que possa ser possível o acompanhamento interno da saúde do ativo desenvolvido, é necessário que existam configurações padronizadas da instituição, que seriam únicas para todo um grupo de desenvolvedores. Para isso o Sensedia Dashboard fornece um ambiente que permite o gerenciamento de configurações padronizadas que podem ser utilizadas por times desenvolvedores.

- **Configuração dos indicadores de qualidade:** Cada indicador requer uma configuração própria para sua execução, dependendo de sua especialidade as configurações podem exigir avaliações individualizadas de seus resultados, para suprir esta necessidade adotamos o conceito de "*Traffic lights*" (semáforos) que estipula um intervalo de satisfação para as avaliações necessárias.

- **Traffic lights:** Cada traffic light, como mostra a Figura 2, representa um intervalo de valores que define qual é a faixa ideal de cada resultado, apresentando assim os valores de resultado ideais em verde, os intermediários (alerta) em amarelo e os resultados ruins em vermelho. Desta forma o usuário pode ter uma visão geral consolidada do resultado para acompanhamento visual.



Figura 2. Exemplo de um Traffic light (semáforo).

Execução: O processo de execução tem como objetivo automatizar as análises e pode ser invocado de três formas:

i. **Execução em IDE's:** Para garantir a saúde de um ativo precisamos tomar algumas medidas de prevenção para que problemas não sejam gerados, pensando neste problema foi desenvolvido um módulo integrado a diversos ambientes de desenvolvimentos que faz a validação em tempo de desenvolvimento do que está sendo desenvolvido, assim, antes de versionar um código o desenvolvedor irá validar o conteúdo no próprio ambiente de desenvolvimento.

ii. **Execução imediata:** O Painel de configurações e resultados centraliza as informações e opções de executar um ativo específico para visualização imediata da qualidade ou agendamento. Maiores detalhes serão apresentados no tópico 3.2.

iii. **Execução agendada:** Como dito anteriormente, uma atividade essencial para o sucesso da qualidade de software é o acompanhamento contínuo da saúde dos ativos, para isso, a solução mais comum é a adoção de um agendamento periódico de analises que irá fornecer insumos através de resultados comparativos. Assim a solução proposta fornece uma funcionalidade de agendamento para execução de ativos selecionados.

- **Análise**: Conceito representativo de um conjunto de avaliações executadas sobre um ativo seja ela manual ou agendamento automático, cada execução é representada por uma *análise* que, atingindo o nível de qualidade desejado, deve ser promovida à baseline, que servirá de base de comparação para as futuras análises representadas em um gráfico de evolução da saúde do ativo.

- **Baseline:** Uma baseline é uma análise que foi definida como um padrão de comparação, ou seja, uma análise que se mostra ideal para aquele ativo. Outras análises que apresentarem seus resultados abaixo desta baseline são consideradas como uma baixa na saúde do ativo, gerando assim uma notificação para os seus responsáveis.

- **Acompanhamento geral dos resultados:** Um acompanhamento mais abrangente pode ser feito através do medidor geral da instituição, onde é consolidado o resultado de todos os ativos que estão sob acompanhamento. Na mesma interface são representados também os resultados individuais de cada ativo através de um medidor visual de satisfação. Para detalhar a visualização do resultado foi criado relatório individualizado por ativo que descreve o resultado de cada indicador e seu enquadramento no semáforo (verde, amarelo ou vermelho). Nesse mesmo relatório é possível fazer o download de uma análise ainda mais aprofundada (resultado avançado), onde podemos encontrar o detalhamento da violação no código do indicador correspondente.

- **Identificação de disponibilidade de serviços:** Alguns ativos podem fornecer ou requisitar serviços de outros sistemas. Para uma avaliação completa de um ativo é necessário verificar a disponibilidade destes serviços, analisando assim o tempo de resposta destes serviços e exibindo um conjunto de semáforos indicando se o projeto está em risco ou não. Para isso contamos com um módulo de integração com alguns dos ESBs (Enterprise Service Buss, ou Barramentos de Serviços Corporativos) mais conhecidos no mercado, este módulo será detalhado no tópico 3.2 deste artigo.

3.2. Modelagem da arquitetura

Foi necessário desenvolver uma arquitetura complexa baseada em módulos que fosse capaz de prover todas as integrações desejadas de forma independente de plataforma e que tornasse fácil a tarefa de adicionar um novo indicador, característica muito importante para empresas que já tem algum tipo de indicador de qualidade de software, porém, não consegue centralizar as informações desejadas. A figura 3 exibe a arquitetura do Sensedia Dashboard, os módulos existentes e suas integrações, a especialidade de cada módulo será descrito a seguir:

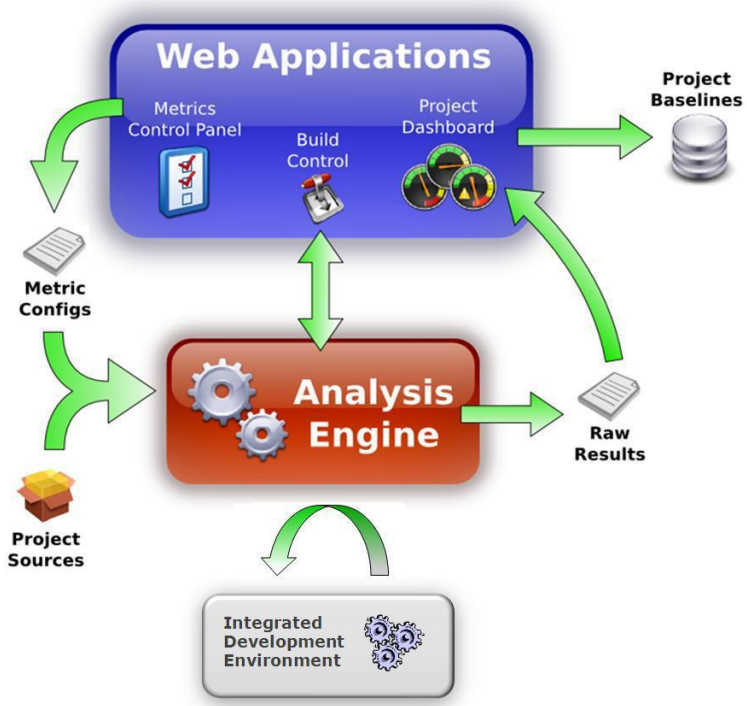


Figura 3. Arquitetura dos principais módulos do Senesedia Dashboard.

Motor de execução de indicadores (*Analysis Engine*)

O motor de execução de indicadores é o módulo onde são executados todos os indicadores ue será detalhado neste mesmo tópico. Podem ser definidos diversos módulos para execuções em ambientes variados, sendo necessário apenas o cadastramento e associação de qual motor de execução irá analisar qual ativo.

Para garantir a centralização das informações, este módulo faz requisições de configurações a serem executadas, eliminando assim a necessidade de configuração para cada motor de execução e com isso garantindo a facilidade em se acoplar novos indicadores customizados deixando a cargo do painel as configurações necessárias (extensibilidade). Embora exista a facilidade de configuração e execução através do painel, esta não é a única maneira. É também possível executar o motor a partir de uma configuração isolada, o que facilita sua integração com outros sistemas. Como é o caso da IDE, onde pode ser escolhido uma configuração do painel ou criar sua própria configuração para ativos que ainda não estão cadastrados. Painel de configurações e resultados. (Web Application)

O Painel de configurações e resultados é uma ambiente web de centralização e gerenciamento das informações dos ativos, indicadores e seus resultados. Todas as funcionalidades descritas no tópico 3.1.1 são geradas por este módulo. A comunicação com outros módulo é feita utilizando *Web Services.*

O processo de configuração dos indicadores para análise de um ativo é uma tarefa que exige conhecimentos sobre as métricas dos indicadores e sobre os ativos sob análise. Em função disso, a configuração dos indicadores de qualidade é tipicamente papel do arquiteto de software, que saberá ponderar de forma apropriada as configurações.

Plugins de execução para IDEs

Como dito anteriormente, existe um módulo de integração com os principais ambientes de desenvolvimentos que tem como responsabilidade envolver o principal setor na saúde de um ativo, o desenvolvimento. Os chamados plugins ou addins, são pequenas aplicações incorporadas as IDEs que permitem um acompanhamento mais pontual do que está sendo desenvolvido e ocasionalmente disparar uma nova análise de um ativo no próprio ambiente. Esses plugins apresentam um ambiente de visualização diferenciado dentro da própria IDE, que informam diretamente ao desenvolvedor o resultado do conteúdo analisado, assim permitimos que o controle de qualidade do desenvolvimento seja mantido desde sua construção.

Extensibilidade

A inclusão de novos indicadores é uma atividade muito praticada em um ambiente de desenvolvimento de software, que agrega cada vez mais a saúde de um ativo. Pensando nisso, a arquitetura foi projetada para ser extensível fornecendo para isso uma biblioteca própria.

4. Resultados

Desde sua primeira versão, o produto gerado vem sendo utilizado em diferentes cenários corporativos reais. Os principais exemplos de utilização são em fábrica de software sob demanda e fábrica de produtos de software. Em ambos os benefícios podem ser facilmente encontrados.

A Ci&T é um exemplo de fábrica de desenvolvimento de software sob demanda (Projetos). Na Ci&T o dashboard é utilizado para acompanhamento dos projetos desde o início do desenvolvimento, possibilitando que os desenvolvedores e arquitetos determinem os níveis de qualidade para cada indicador, e o Dashboard realiza a validação periódica, encontrando desvios de qualidade desde o primeiro dia de desenvolvimento. Além disso, é possível para os líderes da fábrica de software acompanhar o andamento de diversos projetos ao mesmo tempo, concentrando esforços em projetos críticos e destacando projetos com alto nível de qualidade.

Como exemplo de empresa de desenvolvimento de produtos, podemos citar a própria Sensedia. A Sensedia possui uma suíte de produtos que estão em constante evolução. O Dashboard é utilizado para gerar métricas periódicas das frentes de evolução e patches de cada produto. Este acompanhamento permite que os desenvolvedores, arquitetos e gestores dos produtos acompanhem a evolução do desenvolvimento, garantindo que as novas versões não se atrasem nem apresentem problemas de qualidade, aspecto fundamental para manter o processo de liberação de releases para os diversos clientes de um mesmo produto.

4.1. Produtos de software gerados

O Painel de configurações e resultados é um módulo que visa à centralização das informações e integrações de outros módulos para tornar toda a solução expansível. A Figura 4 apresenta a interfaces de resultados consolidados, onde podemos ver através dos medidores a saúde de cada ativo e do conjunto de todos os ativos da companhia de uma forma visual facilitada. Na figura 5 podemos ver exemplos de interfaces de configurações de ativos e indicadores.

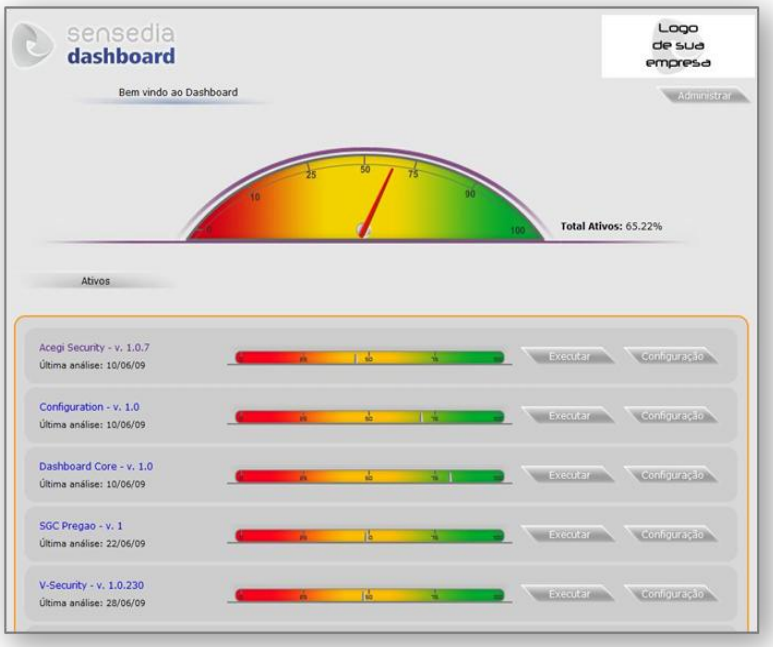


Figura 4. Visualização dos resultados consolidados, geral e por ativo.



Figura 5. Ambiente de configurações de ativos e indicadores.

Para análise de um ativo separadamente, é fornecido resultados individualizado dos indicadores selecionados para este ativo, representado pela Figura 6, e para cada indicador, um arquivo com o detalhamento da violação encontrada. Assim garantimos a identificação pontual do problema. A Figura 7 nos mostra um acompanhamento de um ativo em desenvolvimento em um período de seis meses, e a garantia da qualidade durante todo o seu processo de desenvolvimento.

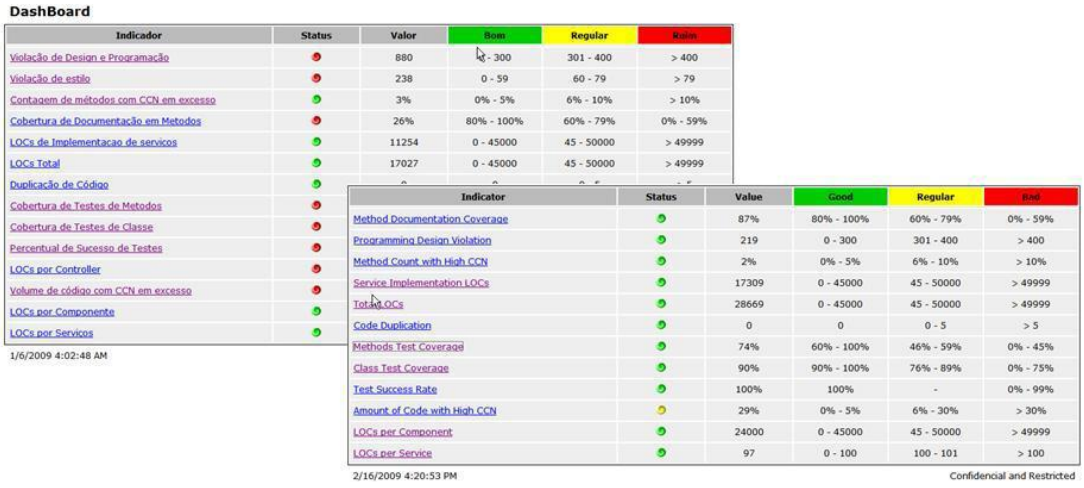


Figura 6. Visualização dos resultados detalhados indicando a melhora na saúde de um ativo.

Grupo	Indicador	27-Mar	27-Feb	27-Jan	27-Dec	27-Nov	27-Oct
Linhas de Código	Numero de Linhas de Código						
Teste Unitário	Percentual de Sucesso de Testes	(0%)	(0%)	(42%)	(35%)	(35%)	(0%)
Duplicação de Código	Duplicação de Código	(0)	(0)	(0)	(0)	(1)	(0)
Tamanho de Código	LOCs por Controller (código fonte)	(0)	(0)	(0)	(0)	(0)	(0)
	LOCs por Serviços (código fonte)	(52)	(43)	(218)	(218)	(218)	(61)
	LOCs por Componente (código fonte)	(5.277)	(2.139)	(4.979)	(4.962)	(4.165)	(2.320)
CCN	Volume de código com CCN em excesso	(0%)	(0%)	(0%)	(0%)	(0%)	(0%)
	Contagem de métodos com CCN em excesso	(0%)	(0%)	(0%)	(0%)	(0%)	(0%)
Documentação	Cobertura de Documentação em Métodos	(78%)	(83%)	(87%)	(87%)	(75%)	(84%)
CheckStyle	Violação de estilo	(1.190)	(1.483)	(1.733)	(1.930)	(2.102)	(2.898)
PMD	Violação de Programação e Design	(175)	(163)	(256)	(251)	(257)	(235)
Alertas	Alertas Novos		(1)			(1)	
	Alertas Abertos		(64)			(23)	
	Alertas Aceitos		(0)			(0)	
Dependência	Violação de Dependência entre Componentes	(0)	(0)	(0)	(0)	(0)	(0)
	Violação de Dependência entre Camadas	(0)	(0)	(0)	(0)	(0)	(0)
Linhas de Código	Total de Linhas de Código	(16.870)	(8.596)	(18.274)	(15.025)	(13.034)	(8.051)
	Total de Linhas de Código (Implementadas)	(15.836)	(8.125)	(16.851)	(13.643)	(12.124)	(7.611)

Figura 7. Visualização do relatório de acompanhamento geral do projeto em um período de seis meses.

4.2. Recursos humanos capacitados

O projeto contou com pesquisadores do Instituto de Computação Unicamp, consultores, arquitetos e gerentes de projetos da Sensedia e com alunos de instituições de ensino superior. Um dos principais resultados do projeto foi a capacitação de pessoal, gerando especialistas nas tecnologias relacionadas ao projeto, sobretudo em qualidade de software. Como o projeto envolveu a definição de “perfis de qualidade” de acordo com cada projeto, foi fundamental desenvolver pesquisas específicas sobre qualidade de software (sob o ponto de vista arquitetural, funcional e não funcional). Destas pesquisas surgiram não só o produto, como especialistas em qualidade de software, atuando atualmente dentro das fábricas de software nas empresas.

4.3. Outros produtos gerados (que foram disponibilizados para o mercado)

Os subprodutos gerados pelo projeto e que agregam grande valor são os módulos de integração com sistema de descoberta de ativos e serviços e com bibliotecas de metadados via Web Services. Estes módulos foram utilizados para integrar o Sensedia Dashboard ao Sensedia Discoverer – ferramenta de identificação de componentes de software em aplicações legadas, premiada com o Prêmio Dorgival Brandão na edição 2007 – e ao Sensedia Repository [Bacili06] - principal ferramenta de repositório de metadados do mercado nacional.

A integração com os produtos comerciais deu grande visibilidade ao projeto, por abranger assim uma maior necessidade das empresas, além de agregar valor ao negócio dos clientes Sensedia Repository e Discoverer, conforme exemplificado na próxima seção.

5. Aplicabilidade dos resultados

O projeto desenvolveu uma solução completa para acompanhamento da saúde de ativos e conseqüentemente uma avaliação do conjunto de ativos que representam uma companhia. Esta solução foi adotada em processos de desenvolvimentos de software baseados em RUP e processos ágeis, em ambos os processos o acompanhamento da qualidade de projetos novos e projetos de manutenção se mostrou satisfatório para todos os objetivos propostos.

Um exemplo prático da aplicabilidade desse projeto em situações práticas é a sua recente adoção pela empresa Synchro (provedora de soluções tecnológicas para a área fiscal), que buscava uma solução capaz de oferecer à empresa dados gerenciais mais concretos para o acompanhamento diário do processo de desenvolvimento de software. O objetivo da companhia, ao trabalhar com o Sensedia Dashboard, é o acompanhamento sistemático da saúde arquitetural de projetos em desenvolvimento para antecipar potenciais problemas e evitar o retrabalho na fase de testes [DecisionReport10].

O conjunto de aplicativos legado no mercado geral de TI é extremamente significativo e de suma importância para qualquer projeto que envolva um conjunto de ações. A integração com sistemas de descoberta de ativo

em legados, aplicado a esse contexto, permite a avaliação e eliminação de legados não qualificados. Podemos fazer um paralelo aos ativos armazenados em bibliotecas de ativos e serviços, como ditas anteriormente neste artigo, a integração com o Sensedia Repository nos permite a avaliação de ativos não qualificados.

6. Características inovadoras

Relacionado às características inovadoras, podemos citar:

- Desenvolvimento de uma técnica para avaliação de saúde de ativos, agregando aspectos arquiteturais, funcionais e não funcionais.
- Implementação real desta técnica, utilizando uma série de mecanismos já existentes, atuando de maneira integrada.
- Integração com sistema de descoberta de ativos em legados.
- Integração com biblioteca de ativos digitais.
- Análise de disponibilidade de serviços fornecidos e providos de ativos, sendo exibido juntamente com o resultado.
- Integração com sistema mobile para acompanhamento contínuo de agentes externos.
- Arquitetura modularizada que possibilita a inclusão de novos indicadores.

7. Conclusão e Perspectivas futuras

As pesquisas realizadas para o desenvolvimento desse projeto nos mostraram que ainda hoje a avaliação da saúde de ativos e componentes que estão sendo desenvolvidos ou evoluídos é considerada uma dificuldade para a maioria das empresas. A proposta deste projeto é de, por meio de um conjunto de soluções mostradas neste artigo, solucionar esse problema de forma automatizada, com uma arquitetura flexível e expansível, viabilizando a incorporação de novos indicadores customizados aos já existentes na ferramenta.

Além do desenvolvimento da solução, o projeto apresentou uma série de efeitos colaterais positivos, que precisam ser amadurecidos, como:

Avanço na integração com SOA, validando padrões como WS-I (padrão de interoperabilidade) e boas práticas de implementação de Web Services estabelecidas pela W3C [W3C] e Oasis [Oasis];

Identificação de novos indicadores para evolução do produto, apontados pelos próprios usuários da solução;

Reconhecimento da equipe na área de qualidade de software, proporcionando convites para participação em eventos como a *2ª Conferência de Qualidade de Software*, promovida pela ASR e realizada nos dias 11 e 12 de novembro de 2009 em São Paulo. Nessa oportunidade ministramos a palestra "*Qualidade Arquitetural em Aplicações*" com base no produto Sensedia Dashboard.

Referências

[Pressman01] PRESSMAN, R.S., 2001, *Software Engineering: A Practitioner´s Approach*, Fifth Edition, McGraw Hill.

[Bacili06] Bacili, K., Oliveira, M., *DigitalAssets Manager: sharing and managing software development assets*, OOPSLA'06 Demo Session, ACM, NY, 2006, pp. 700-701.

[D'Souza99] D. F. D'Souza and A. C. Wills. *Objects, Components, and Frameworks With UML: The Catalysis Approach*. Addison Wesley. USA., 1999.

[Jacobson97] L. Jacobson, M. Griss, and P. Jonsson. *Software Reuse: Architecture, Process and Organization for Business Success*. Addison Wesley. USA., 1997.

[Oliveira05] M. S. Oliveira, I. Garcia, e A. Nunes. *RCCS, Rede de Compartilhamento de Componentes de Software*. X Salão de Ferramentas - XXIII Simpósio Brasileiro de Redes de Computadores - Fortaleza, CE - maio de 2005, 2005.

[Oliveira07] Oliveira, M. Goncalves, E.M. Bacili, K.R. *Automatic Identification of Reusable Software Development Assets: Methodology and Tool - Information Reuse and Integration*, 2007. IRI 2007. IEEE International Conference on

[DecisionReport10] Decision Report. *Synchro melhora eficiência nos processos com SOA*. http://www.decisionreport.com.br/publique/cgi/cgilua.exe/sys/start.htm?infoid=5938&sid=4 (Último acesso:25/02/2010)

[Oasis] OASIS. *Organization for the Advancement of Structured Information Standards*. http://www.oasis-open.org/

[W3C] (2004) *W3C: Web Services Internationalization Usage Scenarios,* W3C Working Group Note 30 July 2004. http://www.w3.org/TR/2004/NOTE-ws-i18n-scenarios-20040730/

Uma Ferramenta para Ter Conhecimento sobre Riscos

Felipe Muller Diehl1, Marcelo Hideki Yamaguti 2

1, 2Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul Av. Ipiranga, 6681 - Prédio 32, CEP: 90619-900 - Porto Alegre – RS

1Tlantic Sistemas de Informação Av. Ipiranga, 6681 - Prédio 96E – TECNOPUC – CEP 90619-900 Porto Alegre - RS

1 fdiehl@tlantic.com.br, 2 yamaguti@ pucrs.br

Abstract. This paper presents the development of a tool that focuses on linking lessons learned with projects identified risks. This tool allow creating, updating, deleting and retrieving information, such as risks, lessons learned, projects, risks categories and learned lessons categories, which are related to linking learned lessons with risks. In this way, the tool allows a more effective identification and monitoring of risks related to projects and their related lessons learned.

Resumo. Este trabalho apresenta o desenvolvimento de uma ferramenta que tem por objetivo realizar o vínculo de lições aprendidas com riscos identificados em projetos. Esta ferramenta permite a criação, atualização, remoção e consulta a informações, tais como riscos, lições aprendidas, projetos, categorias de riscos e categorias de lições aprendidas, que estão relacionadas ao vínculo de lições aprendidas e riscos de projetos. Desta forma, a ferramenta permite a identificação e acompanhamento mais eficaz de riscos associados a projetos e suas respectivas lições aprendidas.

1 Introdução

Nos dias de hoje o termo "qualidade de software" está cada vez mais presente como uma das preocupações das organizações. Isso se deve ao fato de que os usuários e os clientes estão exigindo mais qualidade do software (Pressman, 2001).

Ao gerenciar projetos de desenvolvimento de sistemas, busca-se constantemente garantir o cumprimento de compromissos quanto ao prazo, custos, escopo e qualidade do produto. Porém, sempre se está à mercê de eventos que podem afetar o andamento de nossos projetos, estes eventos são os riscos.

No decorrer do planejamento de projetos, uma boa prática recomendada por modelos referência de qualidade, tais como o Capability Maturity Model Integration (CMMI), por processos de desenvolvimento de software, tais como o Rational Unified Process (RUP), e por modelos de gerenciamento de projetos, tais como o Project Management Body of Knowledge, é o planejamento de riscos, e ao longo do andamento dos projetos, o monitoramento de riscos.

O planejamento de riscos de um projeto consiste em documentar ações de como tratálos caso ocorram, planejando ações para evitá-los ou ações para reduzir seus impactos nos projetos. Já o monitoramento de riscos consiste em periodicamente verificar o estado dos riscos planejados e, caso seja necessário, executar as ações planejadas para tratá-los (PMI, 2008).

Considerando que riscos são pertinentes a todo e qualquer projeto, torna-se de grande importância o gerenciamento de riscos nos projetos. Porém, nem sempre se consegue ter a visibilidade de todos os riscos de um projeto, ou ainda, com o decorrer do projeto, as ações planejadas para o tratamento de riscos acabam tornando-se obsoletas, e é justamente neste ponto onde lições são aprendidas.

Lições são aprendidas nas mais diversas disciplinas envolvidas na execução de um projeto de desenvolvimento de software, sejam elas disciplinas de engenharia de software, de gerenciamento de projetos (onde está envolvido o tratamento de riscos), de treinamentos, ou até de âmbito comercial (PMI, 2008).

A grande importância de lições aprendidas é na busca em se evitar a repetição de erros previamente cometidos, e, considerando-se isso, manter uma base de lições aprendidas tornasse fundamental para toda e qualquer organização que deseja aumentar sua produtividade, garantindo uma maior agilidade e menores custos despendidos na solução de situações adversas previamente já enfrentadas (MPS, 2007).

Em organizações multinacionais focadas no desenvolvimento de projetos de software, existem muitos riscos envolvidos em suas atividades, como por exemplo, riscos de comunicação e tecnológicos, os quais acabam se tornando habituais no planejamento de riscos dos projetos, e, por conseqüência, habituais em bases e documentos de lições aprendidas elaborados por projetos.

2. Objetivos e Justificativa

Visando o aumento da produtividade das organizações e o auxílio a gerentes no gerenciamento de riscos, este trabalho propõe uma ferramenta para o planejamento e monitoramento de riscos, e vinculação de lições aprendidas a estes riscos em um ambiente colaborativo, onde os gerentes possam ter uma base única de riscos vinculados a lições aprendidas, assim podendo-se utilizar do conhecimento obtido em outros projetos para o tratamento de riscos.

Como objetivos específicos buscou-se:

- fornecer um sistema de informação com o intuito de auxiliar o gerenciamento de riscos de projetos através do uso de lições aprendidas vinculadas aos riscos em questão.
- O sistema de informação implementado visa manter uma base única de riscos de projetos, que possa ser utilizada como repositório de riscos de uma organização.
- Através da vinculação de riscos e lições aprendidas, o sistema deve facilitar o gerenciamento de riscos de projetos.
- O sistema implementado visa auxiliar o registro e uso de lições aprendidas de projetos utilizando uma base única.

Lições aprendidas podem ocorrer a qualquer momento de um projeto (PMI, 2008), e devem ser registradas para que os mesmos problemas já enfrentados por um projeto, não sejam novamente enfrentados por outro.

Estes problemas podem ser decorrentes de riscos de projetos, sendo assim, as lições aprendidas durante o gerenciamento destes problemas podem ser vinculadas aos riscos que deram origem aos mesmos.

Por outro lado, ao realizar o planejamento de um projeto, o gerente de projetos deve consultar as lições aprendidas da organização para que possa estar melhor preparado na condução de seu projeto, evitando que tenha problemas já enfrentados por outros projetos, assim aumentando sua eficácia e eficiência (PMI, 2008).

Uma das etapas do planejamento de um projeto, é a etapa de planejamento de riscos (Chrissis, 2006) (PMI, 2008) (Rosenau, 2005), e neste planejamento é importante saber que ações podem e devem ser feitas para que os riscos possam ser mitigados e/ou contingenciados, assim tornando importante o conhecimento de como outros projetos gerenciaram seus riscos, para que as lições aprendidas destes outros projetos possam ser aproveitadas.

Assim sendo, torna-se de grande valia o vínculo de lições aprendidas a riscos para dois momentos do ciclo de vida de um projeto:

- No planejamento do projeto, onde devem ser verificadas as lições aprendidas de projetos anteriores, e os riscos do projeto atual versus os riscos de projetos anteriores, com o objetivo de identificar riscos semelhantes e descobrir o conhecimento vinculado aos mesmos que outros projetos já tiveram para gerenciá-los, assim otimizando o gerenciamento do projeto.
- No fechamento de um problema decorrente, e/ou de um risco propriamente dito. Neste momento torna-se fundamental o vínculo das lições aprendidas oriundas deste problema e/ou risco, alimentando a base de lições aprendidas da organização, e evidenciando as soluções encontradas no projeto atual, que servirão para o planejamento de projetos futuros.

Com base nestas considerações, a criação de uma ferramenta que vincule riscos e lições aprendidas é importante para a obtenção de melhor desempenho em projetos, otimizando o gerenciamento de riscos com o uso de lições aprendidas.

3. Metodologia de Execução

O projeto foi desenvolvido em seis etapas:

1. Fundamentação teórica: contextualização do trabalho, aprofundamento de conceitos relativos à gestão de riscos e lições aprendidas em projetos.

2. Pesquisa de trabalhos similares: busca de trabalhos anteriores que abordassem o tema de vinculação de riscos a lições aprendidas em projetos.

3. Levantamento e especificação de requisitos: identificação das principais características e funcionalidades esperadas para uma ferramenta de suporte a gestão de riscos e lições aprendidas.

4. Análise e projeto da solução: foram definidas as soluções conceituais e lógicas para a implementação da ferramenta em termos de arquitetura, tecnologias de informação (linguagem de programação, banco de dados, servidor de aplicação, etc.) 5. Implementação: realizou-se a geração do código da ferramenta, a instanciação do banco de dados e a configuração do servidor de aplicação.

5. Testes: foram realizados testes funcionais a partir dos casos de uso definidos anteriormente.

4. Resultados Obtidos

Produtos de software gerados (módulos ou programas de computador resultantes do projeto, disponibilizados para o mercado): como resultado concreto do projeto foi gerado um protótipo da ferramenta que possibilita o gerenciamento de riscos, registro de lições aprendidas e realização do vínculo entre estes, assim compartilhando o conhecimento adquirido em projetos, para que os próximos projetos tenham maior produtividade ao tratar seus riscos.

A ferramenta foi desenvolvida durante um trabalho de conclusão de Curso de Graduação de Bacharelado em Sistemas de Informação na Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul.

5. Aplicabilidade dos Resultados

O uso da ferramenta proporciona a gerentes de projetos a vinculação de lições aprendidas a riscos de projetos, com o objetivo de compartilhar o conhecimento adquirido em projetos, para que os próximos projetos tenham maior produtividade ao mitigar ou contingenciar os seus riscos.

6. Características Inovadoras

Já existem ferramentas para o gerenciamento de riscos em projetos, e usualmente empresas com processos mais maduros disponibilizam meios aos seus colaboradores para o compartilhamento de lições aprendidas.

Entretanto, não foi encontrada nenhuma ferramenta que vincule os riscos para a gestão de lições aprendidas, que é o aspecto inovador proposto neste trabalho.

7. Conclusão e Perspectivas Futuras

O sistema gerado disponibiliza as seguintes funcionalidades:

- Inserção, exclusão, edição e listagem de riscos no sistema;
- Inserção, exclusão, edição e listagem de estratégias de mitigação de riscos;
- Inserção, exclusão, edição e listagem de estratégias de contingência de riscos;
- Inserção, exclusão, edição e listagem de categorias de riscos;
- Inserção, exclusão, edição e listagem de lições aprendidas no sistema;
- Inserção, exclusão, edição e listagem de categorias de lições aprendidas;
- Inserção, exclusão, edição e listagem de projetos no sistema;
- Inserção, exclusão, edição e listagem de usuários no sistema;
- Vinculação de lições aprendidas a riscos;
- Geração de relatórios;
- Monitoramento de riscos;
- Inserção de riscos da base única em projeto;
- Geração de histórico de alterações em riscos.
- Inserção, exclusão, edição e listagem de áreas da empresa no sistema;
- Inserção, exclusão, edição e listagem de categorias de mitigação de riscos no sistema;
- Inserção, exclusão, edição e listagem de categorias de contingência de riscos no sistema.

Estas funcionalidades permitem que gerentes de projetos distintos possam gerenciar os riscos de seus projetos, consultar riscos e lições aprendidas de outros projetos, e vincular novos riscos de seus projetos a lições aprendidas, e desta forma melhorar o conhecimento organizacional sobre como gerenciar os riscos em seus projetos.

Os trabalhos futuros identificados são baseados na evolução da ferramenta atual com a inserção de outros itens de controle de gerenciamento de projetos, objetivando a realização de um gerenciamento integrado em uma única ferramenta.

- Expandir a ferramenta para que ela possa ser utilizada como um diário de projeto, vinculando riscos e lições aprendidas a acontecimentos do projeto, com o objetivo de dar visibilidade ao gerente de projetos do impacto destes acontecimentos em riscos, bem como criar a possibilidade de vincular lições aprendidas a estes acontecimentos.

- Expandir a ferramenta para vincular indicadores de projetos com os riscos, assim verificando o impacto direto de um risco em um indicador do projeto.

Aproveitando a primeira idéia citada, seria possível verificar quais são as ações que estão sendo tomadas para que os indicadores fiquem de acordo com o esperado.

- Dispor uma interface em forma de site para que envolvidos interessados nos riscos do projeto possam acompanhá-los conforme necessidade, assim monitorando os riscos do projeto sem a necessidade do gerente de projetos ficar enviando relatórios periodicamente.

8. Referências Bibliográficas

CHRISSIS, Mary Beth; KONRAD, Mike; SHRUM, Sandy. CMMI: Guidelines for Process Integration and Product Improvement. Editora Addison-Wesley Professional. 2 ª Edição, 2006.

MELHORIA DE PROCESSO DE SOFTWARE BRASILEIRO. Mini-curso de Gerência do Conhecimento. Belo Horizonte, 2007.

PRESSMAN, Roger S. Software Engineering: a practitioner's approach. Editora McGraw-Hill Companies, Inc. 5ª edição 2001.

PROJECT MANAGEMENT INSTITUTE, INC. A guide to the project management body of knowledge: PMBOK guide. 4ª edição 2008.

ROSENAU, Milton D.; GITHENS, Gregory D. Successful project management: a step-by-step approach with practical examples. Editora John Wiley & Sons, Inc. 4ª edição 2005.

Verificação dos Processos de um Modelo de Desenvolvimento com o MPS.BR

Priscila Souza 1

1Curso de Tecnologia em Análise e Desenvolvimento de Sistemas de Informação – Faculdade Jesus Maria José (FAJESU) - Taguatinga – DF – Brasil

priscila.se@gmail.com

Abstratct. *This article will present the study in check the processes of the Model of Software Development Environment, prepared by the Management of Information Technology from a company, with regard to the processes of the level of G MPS.BR.*

Resumo. *Este artigo apresentará o estudo realizado na verificação dos processos do Modelo de Ambiente de Desenvolvimento de Software elaborado pela Gerência de Tecnologia da Informação de uma empresa, com relação aos processos do nível G do MPS.BR.*

1. Introdução

Nos últimos anos surgiram várias empresas produtoras de software que vêm concentrando esforços na tentativa de elaborar produtos cada vez mais sofisticados, e que atendam às necessidades tecnológicas dos diversos ambientes empresariais existentes.

Nessa busca constante, as fábricas de softwares têm passado por várias experiências, nem sempre satisfatórias, pois a cada novo produto desenvolvido são realizados procedimentos diferentes dependendo do tamanho, da complexidade e das pessoas que estão envolvidas no trabalho. [1]

Com o passar dos anos, muitas delas descobrem aspectos que podem ser igualmente utilizados no desenvolvimento de novos produtos, ou mesmo aqueles que devem ser analisados isoladamente. Com isso, foram desenvolvendo formas de trabalhar que tornaram a entrega do produto final mais rápida, com qualidade, dentro daquilo que realmente foi pedido pelo cliente e, principalmente, com menor probabilidade de erros futuros. [2]

Dentro dessa perspectiva, surgiu o Modelo de Processo de Software Brasileiro (MPS.BR) elaborado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) para que as micros, pequenas e médias empresas brasileiras produtoras de software possam ter uma referência e/ou certificação de qualidade.

Com o desejo de normatizar seu processo de software, a Gerência de Tecnologia da Informação de uma empresa do setor previdenciário elaborou um modelo próprio de desenvolvimento de software que será objeto de estudo deste trabalho, tendo em vista a necessidade de verificar se existem neste modelo os processos apresentados no MPS.BR.

Nas próximas seções deste artigo serão apresentados conceitos gerais do MPS.BR enfatizando o nível G na seção 2, os aspectos gerais do Modelo de Ambiente de Desenvolvimento de Software da entidade na seção 3, apresentação do estudo de verificação do modelo com o MPS.BR na seção 4, e a análise dos resultados obtidos na seção 5.

2. O MPS.BR

Muitas empresas Brasileiras desejam alcançar os padrões reconhecidos internacionalmente. Outras não possuem este interesse, mas pretendem de alguma forma melhorar seus processos com o objetivo de aumentar a qualidade de seus produtos. Assim o MPS.BR é o modelo brasileiro que surge como uma alternativa para atender as expectativas destas empresas, pelo baixo custo da sua implementação, por apresentar características peculiares da produção de software no Brasil e por permitir que sejam alcançados os mesmos padrões de qualidade do CMMI[3].

Cabe ressaltar que o MPS.BR tem como fundamentação principal as normas NBR ISO/IEC 12207- Processo de Ciclo de Vida de Software, NBR ISO/IEC 15504 – Avaliação de Processo, e também utiliza como referência o modelo CMMI-DEV que trata das disciplinas Engenharia de Sistemas, Engenharia de Software, Aquisição de Software, Gerência e Desenvolvimento da Força de Trabalho, Desenvolvimento Integrado do Processo e do Produto.

O MPS.BR está subdivido em sete níveis de maturidade, numa escala de evolução que começa no nível mais baixo, o G, até se atingir o nível mais alto que é A. Estes níveis indicam o desempenho da organização: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado).

Um nível de maturidade é alcançado quando os propósitos e todos os resultados esperados dos respectivos processos e dos atributos dos processos daquele nível são evidenciados.

Os processos atingidos em nível anterior são cumulativos com o próximo nível. Assim, uma organização que pretende se adequar ao nível de maturidade E, deve estar com os propósitos e todos os resultados esperados do processo e dos atributos de processo dos níveis G, F e E totalmente implantados.

2.1. O nível de maturidade G

No nível de maturidade G os processos executados pela organização são parcialmente gerenciados. É o primeiro nível de maturidade que deve ser atingido. Nele estão os processos: Gerência de Projetos e Gerência de Requisitos. [3]

Gerência de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas às atividades de projeto a fim de atender seus requisitos. As pessoas envolvidas no projeto devem estar preocupadas com cinco processos importantes: iniciação, planejamento, execução, monitoramento, controle e encerramento. Também é necessário observar as seguintes áreas de conhecimento para a entrega do produto final: qualidade, escopo, tempo e custo. [2]

Para efetivação do Processo Gerência de Projetos, a organização deve apresentar os resultados esperados conforme tabela 1.

Tabela 1. Resultados esperados do processo Gerência de Projetos[4].

GPR 1	O escopo do trabalho para o projeto é definido.
GPR 2	As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados.
GPR 3.	O modelo e as fases do ciclo de vida do projeto são definidas.
GPR 4.	O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas.
GPR 5.	O orçamento e o cronograma do projeto, incluindo marcos e/ou pontos de controle, são estabelecidos e mantidos.
GPR 6.	Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados.
GPR 7.	Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo.
GPR 8.	As tarefas, os recursos e o ambiente de trabalho necessários para executar o projeto são planejados.
GPR 9.	Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança.
GPR 10.	Planos para execução do projeto são estabelecidos e reunidos no Plano do Projeto.
GPR 11.	A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados.
GPR 12.	O plano de projetos é revisado com todos os interessados e o compromisso com ele é obtido.
GPR 13.	O progresso do projeto é monitorado com relação ao estabelecido no Plano do Projetos e os resultados são documentos.
GPR 14.	O envolvimento das partes interessadas no projeto é gerenciado.
GPR 15.	Revisões são realizadas em marcos do projeto conforme estabelecido no planejamento.
GPR 16.	Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas.
GPR 17.	Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até sua conclusão.

O processo Gerência de Requisitos visa controlar a evolução dos requisitos dos projetos identificando inconsistências entre estes, os planos do projeto e os produtos de trabalho do projeto. As mudanças identificadas devem ser documentadas e a rastreabilidade dos requisitos deve ser mantida para garantir que os objetivos do projeto sejam alcançados. [4]

Os resultados esperados deste processo constam na tabela 2.

Tabela 2. Resultados esperados do processo Gerência de Requisitos[4].

GRE 1	O entendimento dos requisitos é obtido juntos aos fornecedores de requisitos.
GRE 2	Os requisitos de software são aprovados utilizando critérios objetivos.
GRE 3	A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida.
GRE 4	Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos.
GRE 5	Mudanças nos requisitos são gerenciadas ao longo do projeto.

Para indicar o grau de refinamento e institucionalização da execução dos processos referentes ao nível G, seus atributos de processos devem ser avaliados conforme tabela 3.

Tabela 3. Resultados esperados do dos atributos de processo do Nível G[4].

AP 1.1 – O processo é executado	
RAP 1	O processo atinge seus objetivos
AP 2.1 - **O processo é gerenciado**	
RAP 2	Existe política organizacional estabelecida e mantida para o processo.
RAP 3	A execução do processo é planejada
RAP 4	Medidas são planejadas e coletadas para monitoração do processo.
RAP 5	Os recursos necessários para a execução do processo são identificados e disponibilizados.
RAP 6	As pessoas que executam o processo são competentes em termos de formação, treinamento e experiência.
RAP 7	A comunicação entre as partes interessadas no processo é gerenciada de forma a garantir o seu envolvimento no projeto.
RAP 8	Métodos adequados para monitorar a eficácia e a adequação do processo são determinados.
RAP 9	A aderência dos processos executados às descrições de processo, padrões e procedimentos é avaliada objetivamente e são tratadas as não conformidades.

3. O Modelo de Ambiente de Desenvolvimento de Software

O Modelo de Ambiente de Desenvolvimento de Software foi elaborado pela Gerência de Tecnologia com o objetivo de padronizar todos os processos que envolvem a produção de software da empresa. [5] Esta organização é uma das três maiores Entidades Fechada de Previdência Complementar (EFPC) do país, teve como motivação para elaborar este trabalho, a necessidade de documentar e acompanhar com maior critério os produtos e serviços apoiados em software, para atender com dinamismo às suas necessidades diárias e aos novos projetos, de forma tempestiva, garantindo a qualidade e transparência de seus serviços da área de Tecnologia da Informação.

Para elaborar o modelo a Gerência se valeu de conceitos retirados de regulamentações e *framework*s existentes como as normas ISO/IEC 12207, 15504 e 9126, o MPS.BR, a UML, o COBIT e o ITIL [5]. Utilizando-se das melhores práticas, adaptou no modelo os processos de forma que atendam às necessidades da organização, ou seja, de acordo com a forma de trabalho institucionalizada e com as características dos projetos sob sua gestão.

O modelo está estruturado em Método de Desenvolvimento de Software Orientado a Objetos, Método de Gestão de Configuração e Mudanças de Software, Arquitetura do Ambiente de Desenvolvimento de Software e Repositório de Componentes.

Ainda no modelo, são apresentados todos os procedimentos pertinentes a cada um dos processos elaborados, constituindo assim, os trâmites que devem ocorrer entre usuários, Gestor de Sistemas e gestores de cada um

dos processos dentro da equipe de TI, até que o projeto seja concluído. Esses trâmites incluem a forma de comunicação entre as partes e a documentação de todos os artefatos produzidos durante o projeto. Os *templates* a serem utilizados pela equipe também são apresentados no modelo da entidade em questão.

4. A Verificação

4.1 A avaliação segundo o MPS.BR

A avaliação formal dos processos de uma organização é realizada por Instituições Avaliadoras autorizadas que possuem profissionais avaliadores certificados pela SOFTEX. O avaliador deve seguir as descrições e subprocessos determinados pelo Método de Avaliação do MPS.BR (MA-MPS)[6].

Em linhas gerais, a avaliação segundo o MA-MPS ocorre da seguinte forma: indica-se o nível de maturidade a ser avaliado e para isso são escolhidos alguns projetos executados pela organização. É realizada a coleta de documentos relativos aos projetos selecionados e entrevistas com as pessoas envolvidas. As evidências encontradas são identificadas através de artefatos diretos e/ou indiretos dentro de cada resultado esperado do processo e dos atributos de processo.

Após levantamento das evidências, é realizada a caracterização do grau de implementação de todos os resultados esperados do processo e dos atributos do processo, conforme tabela 4. Com esse levantamento é possível indicar o grau de implementação de cada processo da organização ou da unidade em avaliação como Satisfeito ou Não Satisfeito. Será formalizado o nível de maturidade pretendido quando todos os processos deste nível forem caracterizados como Satisfeitos.

Tabela 4. Escala para caracterização do grau de implementação de um resultado esperado do processo e de resultado esperado de atributo do processo[6].

Grau de Implementação	Caracterização
Totalmente Implementado(T)	□ O indicador está presente a é julgado adequado □ Existe pelo menos um indicador indireto e/ou afirmação contemplando a implementação □ Não foi notado nenhum ponto fraco substancial.
Largamente Implementado (L)	□ O indicador direto está presente ou é julgado adequado. □ Existe pelo menos um indicador indireto e/ou afirmação confirmando a implementação. □ Foi notado um ou mais pontos fracos substanciais.
Parcialmente Implementado (P)	□ O indicador direto não está presente ou é julgado inadequado. □ Artefatos / afirmações sugerem que alguns aspectos do resultado esperado estão implementados. □ Pontos fracos forma documentados.
Não Implementado (N)	□ Qualquer situação deferente das acima.
Não Avaliado (NA)	□ O projeto não está na fase de desenvolvimento que permite atender ao resultado ou não faz parte do escopo do projeto atender ao resultado.
Fora do Escopo (F)	□ O resultado esperado está fora do escopo da avaliação documentado no plano de avaliação.

4.2 Verificação do Modelo

Neste trabalho, o Modelo de Ambiente de Desenvolvimento de Software foi verificado com o objetivo de conferir se apresenta os processos do MPS.BR. Para esta análise foi escolhido o nível G pois o modelo está em fase inicial de implantação na organização e em fase de aplicação em projeto piloto.

Diante do exposto e por não se tratar de avaliação formal, neste trabalho não será contemplado a caracterização em grau de implementação dos processos, nem o grau de maturidade da unidade organizacional que criou o modelo.

4.2.1 Método de Verificação.

A verificação foi realizada com base nas definições de cada resultado esperado dos processos e dos atributos de processos que constam no Guia de Implementação do Nível G.

A planilha de verificação (figura 1) foi elaborada para identificar as evidências do modelo em relação aos resultados esperados dos processos e dos atributos de processo deste nível: Gerência de Projetos e Gerência de Requisitos.

Gerência do Projeto

Resultado esperado segundo MPS.BR	Modelo do Ambiente de Desenvolvimento de Software		Referência		O que está ausente
	ARTEFATO/DOCUMENTO	ITEM	NORMA	ITEM/ARTEFATO	
O propósito do processo de Gerência de Projetos é identificar, estabelecer, coordenar e monitorar as atividades , tarefas e recursos que um projeto necessita para produzir um produto e/ou serviço, no contexto dos requisitos e restrições do projeto.					
GPR 1. O escopo do trabalho para o projeto está definido Verificar se o escopo do projeto foi definido. O escopo pode definido por meio da EAP ou do documento visão	DECLARAÇÃO DO ESCOPO DO PROJETO - DEP	5.4	PmBok	4.2	
	TERMO DE ABERTURA DO PROJETO - TAP	5.3	PmBok	4.1	
	Estrututa Analítica do Projeto - EAP	5.12	PmBok	5.3.2	A EAP não está representada em forma de gráfico.
	Modelo de Negócio	5.13	RUP	Conjunto de Artefato da Modelagem de Negócio	
	Plano de Gerenciamento do Projeto	3.4.14.5.3(apenas referencia)	PmBok	4.3	Está previsto mas ainda não foi colocado no anexo.

Figura 1. Planilha de Verificação.

A planilha apresenta detalhadamente cada um dos resultados esperados destes processos. Assim, cada evidência foi registrada ao lado do resultado esperado correspondente. As evidências correspondem aos artefatos encontrados no modelo. Também foi registrada ao lado de cada artefato, a norma que serviu como base no modelo e em que item a referida norma trata da elaboração do artefato.

Para validar a planilha, foram realizadas entrevistas com as pessoas envolvidas na criação do modelo, na intenção de confirmar as informações registradas e identificar os artefatos que possivelmente são utilizados na gerência, mas ainda não foram incluídos no mesmo.

Todo material referente ao modelo da organização e às normas utilizadas como referência foram disponibilizados pelos responsáveis.

A fase final constituiu em identificar a evidências que não foram encontradas no modelo da organização, segundo as descrições dos resultados esperados no Guia de Implementação do nível G. Assim, nas entrevistas foi solicitada a confirmação dos artefatos que realmente estão ausentes ou não.

5. Análise dos resultados obtidos

Após o levantamento de todas as evidências para cada um dos resultados esperados dos processos e dos atributos do processo, foi possível classificá-las em quatro grupos:

- Artefatos previstos, mas que ainda não foram inseridos no modelo;
- Artefatos totalmente ausentes;
- Artefatos encontrados conforme a descrição dos resultados esperados dos processos e dos atributos dos processos;
- Evidências parcialmente encontradas.

Na figura 2 podemos visualizar percentualmente os grupos em que as evidências se enquadraram: 58% dos artefatos foram encontrados conforme os resultados esperados, 22% dos artefatos estão previstos e ainda serão incluídos no modelo, 14% dos artefatos estão ausentes e 6% dos artefatos foram parcialmente encontrados.

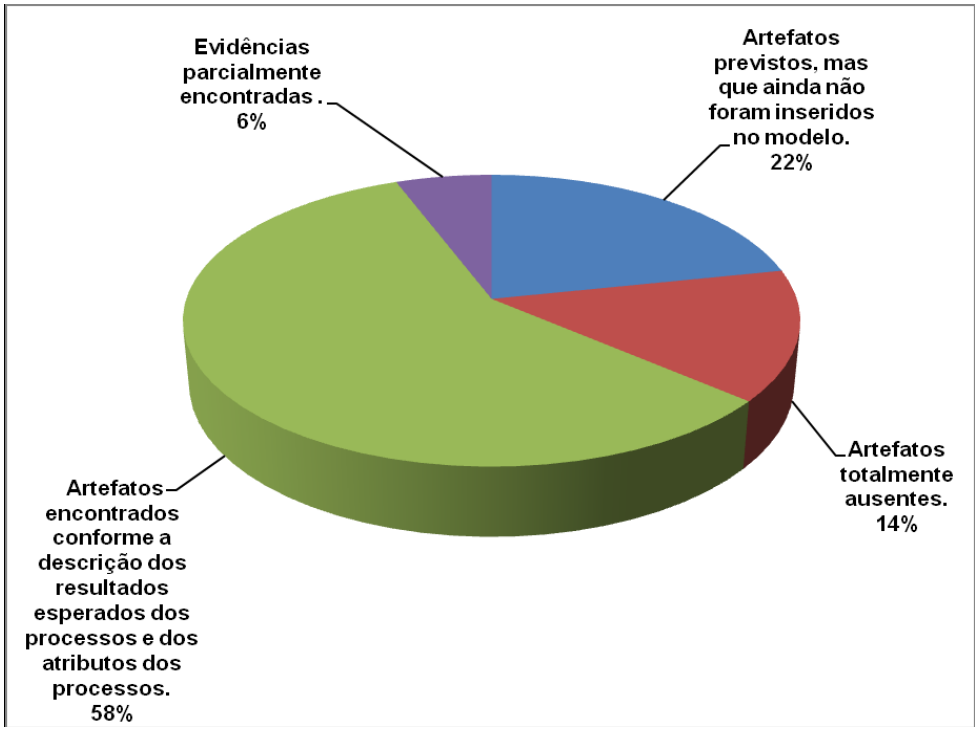


Figura 2. Percentual de artefatos encontrados no modelo.

Os artefatos que estão totalmente ausentes (14%) são aqueles que neste momento não são considerados relevantes sua utilização no Modelo do Ambiente de Desenvolvimento de Software.

Para exemplificar algumas das evidencias encontradas podemos citar o Plano de Gerenciamento de Projeto que está previsto mais ainda não foi colocado no modelo, e o documento de Revisões e Inspeções que foi parcialmente encontrado do framework por estar em fase de elaboração.

Considerando os artefatos parcialmente encontrados e os totalmente encontrados, na conjuntura atual, o modelo apresenta 64% dos processos relativos ao nível G do MPS.BR. Desconsiderando apenas os artefatos totalmente ausentes, podemos inferir que após o término da inclusão dos artefatos previstos e o ajuste dos itens parcialmente encontrados, o modelo da entidade estará com 86% dos seus processos compatíveis com o nível em questão.

Com este trabalho foi identificado que o modelo da empresa apresenta maior compatibilidade com processo Gerência de Requisitos do que com o processo de Gerência de Projetos.

6.Considerações Finais

O MPS.BR permite que empresas brasileiras possam utilizar suas orientações para padronizar seus processos. E o nível G contempla os processos Gerência de Projetos e Gerencia de Requisitos que são igualmente importantes, pois é na execução deles que ficam as marcas indicadoras do sucesso ou não de um projeto. Analisar o quanto os processos de uma organização estão em consonância com os processos apresentados em um modelo de referência como o MPS.BR, é um trabalho sistemático e que envolve conhecimento avançado do que representa cada um desses processos na execução de um projeto.

A verificação realizada permitiu conhecer quanto os processos do Modelo de Desenvolvimento de Ambiente de Software possuem referência no nível G do MPS.BR. Isso porque, pelo estudo realizado, observou-se que o modelo possui grande parte das evidências relativas aos resultados esperados deste nível. As evidências consideradas ausentes foram assim classificadas por não existirem artefatos relativos aos resultados esperados correspondentes. Mas é importante salientar que o modelo faz referência a muitos dos resultados por meio das suas diretrizes.

O modelo analisado foi desenvolvido com a preocupação de utilizar artefatos aderentes à cultura da organização e que propiciam alcance de resultados conforme as expectativas dos clientes.

A importância de realizar este trabalho como este está na oportunidade conhecer e verificar como ocorrem os processos de desenvolvimento de software de uma organização, mesmo que este não seja sua atividade fim. Ainda mais quando os seus processos são correlatos a um modelo apoiado em normas internacionalmente reconhecidas.

Para trabalhos futuros sugere-se que sejam verificados os processos do modelo quanto aos demais níveis do MPS.BR. Pode-se também optar por realizar uma análise pontual de cada uma das evidências ausentes a fim de realizar a implementação destes no modelo.

7. Referências

[1] Filho, Wilson de Pádua Paula. **Engenharia de Software.** Fundamentos, Métodos e Padrões. Ed. LTC. 2ª Edição. Rio de Janeiro. 2003.

[2] PMI - Project Management Institute. **Um Guia do Conjunto de Conhecimentos do Gerenciamento de Projetos - PMBOK Guide** – 3ªed. Four Campus Boulevard, Newton Square, 2004.

[3] ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. **MPS.BR - Guia Geral**, v 1.2, 2007. Disponível em www.softex.br

[4] ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. **MPS.BR - Guia de implementação – Parte 1**, versão 1.1, junho 2007. Disponível em www.softex.br

[5] Modelos de Ambiente de Desenvolvimento de Software. Elaborado pela entidade objeto deste estudo.

[6] ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. **MPS.BR - Guia de avaliação**, versão 1.1, julho 2007. Disponível em www.softex.br.

WDP-RT: Uma Técnica de Leitura para Inspeção de Usabilidade de Aplicações Web

Marcos Gomes1, Tayana Conte1

1Departamento de Ciência da Computação Universidade Federal do Amazonas (UFAM) – Manaus, AM - Brasil

marcos.sgomes@gmail.com , tayana@dcc.ufam.edu.br

Abstract. *This paper presents the development of the usability inspection technique WDP-RT from its conception to its use on industry. This technique aims guarantee quality of Web applications and was evaluated through experimental studies. The goal of this paper is to describe how usability inspections had been executed using WDP-RT, in order to encourage the industries in using usability inspections to improve its systems quality.*

Resumo. *Este artigo apresenta o desenvolvimento da técnica de inspeção de usabilidade WDP-RT desde sua concepção até sua utilização na indústria.*

Esta técnica visa garantir a qualidade das aplicações Web e foi avaliada através de estudos experimentais. O objetivo deste artigo é descrever como foram executadas inspeções de usabilidade utilizando a técnica WDP-RT, de forma a incentivar empresas de desenvolvimento a realizar avaliações de usabilidade para melhorar a qualidade de seus produtos.

1. Introdução

Assim como testes funcionais são importantes para avaliar a implementação frente aos requisitos do software, a avaliação de interface é importante para analisar a qualidade de uso de um software. A qualidade de uso está estritamente relacionada à capacidade e facilidade dos usuários atingirem suas metas com eficiência e satisfação (PRATES *ET al.*, 2003). O conceito de qualidade de uso mais amplamente utilizado é a usabilidade.

A usabilidade é um dos aspectos relacionados à qualidade em uso de interfaces de sistemas, sendo um dos mais importantes critérios de aceitação para aplicações interativas em geral, e em particular para aplicações Web (HITZ *et al.* 2006). Devido sua relevância para as aplicações Web, a indústria de desenvolvimento de software está investindo em técnicas e ferramentas para projetos e avaliações que ajudem a melhorar esse atributo de qualidade em suas aplicações Web (MATERA *et al.* 2006). Técnicas de avaliação de usabilidade específicas para aplicações Web, tais como CWW (BLACKMON *et al.* 2002), MiLE+ (BOLCHINI e GARZOTTO, 2007) e WDP (CONTE *et al.* 2009a), estão sendo desenvolvidas. Entretanto, o resultado da avaliação utilizando-se algumas das principais técnicas conhecidas pode ser influenciado pelo grau de experiência dos avaliadores, como registrado no método de Avaliação Heurística (NIELSEN, 1993).

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 discorre sobre a técnica desenvolvida e os resultados dos estudos realizados. A seção 5 apresenta a aplicabilidade dos resultados. A seção exibe as características inovadoras desta pesquisa. Por fim, a seção 7 conclui o artigo.

2. Objetivos e Justificativa

A técnica WDP (*Web Design Perspectives-based Inspection Technique*) (CONTE *et al.* 2009a) é uma técnica de inspeção de usabilidade baseada em *checklist,* desenvolvida a partir de uma metodologia baseada em evidências (SHULL *et al.* 2001; MAFRA *et al.* 2006). Os resultados do estudo de observação (CONTE *et al.* 2009a) indicaram que a evolução da WDP para uma técnica de leitura ajudaria os inspetores novatos em inspeções de usabilidade. De acordo com TRAVASSOS *et al.* (2002), uma técnica de leitura é um tipo específico de técnica de inspeção que contém uma série de passos para a análise individual de um produto de software de forma a alcançar a compreensão necessária para uma tarefa específica.

Este trabalho se propõe a definir uma tecnologia para inspeção de usabilidade que apóie a obtenção de maior qualidade nas aplicações Web, através da extensão da técnica WDP para uma técnica de leitura, a WDP-RT (*Web Design Perspectives-based Inspection – Reading Technique*). Esta meta será alcançada através de três objetivos:

- Analisar conjuntos de características de usabilidade, que possam servir de base teórica para a técnica WDP-RT.
- Desenvolver uma técnica de leitura, através da extensão da técnica WDP;
- Evoluir a técnica proposta através de estudos experimentais;

O propósito final é que a técnica definida possa ser utilizada pelos próprios stakeholders (como desenvolvedores e avaliadores de requisitos) dos projetos de desenvolvimento de aplicações Web, apresentando boa relação custo-benefício mesmo quando o inspetor tiver pouca expertise em avaliações de usabilidade.

3. Metodologia de execução

Estudos experimentais são os "blocos de construção do conhecimento" necessários para construir evidências e determinar quais as melhores situações para empregar determinada tecnologia. TRAVASSOS E BARROS (2003) apresentam uma classificação para os estudos experimentais, considerando as diferenças de acordo com o nível de controle existente sobre o ambiente onde são executados, sendo classificados em: *in vivo*, *in vitro*, *in virtuo*, e *in silico*.

Estudos *in vitro* são realizados em um ambiente controlado, com a participação de pessoas que atuam como representantes da população de interesse. Estudos *in vivo* envolvem pessoas em seu próprio ambiente de trabalho em condições realistas. Estudos *in virtuo* e *in silico* possuem como característica a exploração de simulações do mundo real, conduzidas em ambientes virtuais compostos por modelos computacionais.

Para a elaboração da WDP-RT, foram realizados dois estudos *in vitro* e um estudo *in vivo*. Estes estudos possibilitaram o acompanhamento do desenvolvimento e a evolução da técnica WDP-RT.

4. Resultados obtidos

4.1. Processo para melhoria da qualidade de software: Versão inicial da WDP-RT

A WDP-RT se baseia em um conjunto de instruções que devem ser executadas para a verificação da usabilidade da aplicação, agrupadas de acordo com as três perspectivas de projeto Web. Com o propósito de aumentar a cobertura de avaliação da WDP, as instruções da WDP-RT foram definidas realizando-se uma análise de equivalência entre três conjuntos de características de usabilidade: conjunto de itens propostos pela WDP, com base nas heurísticas de (NIELSEN, 1994), os "requisitos não funcionais de usabilidade" (FERREIRA E LEITE, 2003), e as "características funcionais de usabilidade" (JURISTO *et al.* 2007). A Figura 1 apresenta uma visão da primeira versão da WDP-RT (WDP-RT v1). O texto completo da WDP-RT v1 está disponível em (GOMES e CONTE, 2009a).

1 - www.sbc.org.br

3. Para cada página que possuir entrada de dados (preenchimento de caixas de texto, itens de seleção, etc.):
    a. Observe se os dados obrigatórios na entrada de dados estão claramente definidos (por exemplo, marcados com um *).
    b. Observe os campos a serem preenchidos. Existe informação sobre o formato correto para entrada de dados específica? Por exemplo, dd/mm/aaaa para campos do tipo data. O sistema facilita a entrada de dados simples ou estruturalmente complexos?

Figura 1 - Extrato da WDP-RT v1

4.1.1. Execução de estudos experimentais: Primeiro estudo *in-vitro*

Segundo SHULL *et al.* (2001), o primeiro estudo a ser realizado para avaliação de uma nova tecnologia deve ser um estudo de viabilidade, que visa verificar se a nova tecnologia é viável e se o tempo empregado é bem utilizado. Para isto, foi realizada uma inspeção de usabilidade (Junho/2009) comparando a viabilidade da WDP-RT com os resultados apresentados pela WDP. Este primeiro estudo foi discutido detalhadamente em (GOMES *et al.* 2009) e será apresentado um resumo dos resultados do mesmo.

Participantes: Dezoito alunos da UFAM e duas desenvolvedoras de software de uma instituição de pesquisa. Os participantes foram divididos em dois grupos de dez inspetores (equipes WDP e WDP-RT) com níveis de experiência balanceados de acordo com as respostas dos formulários de caracterização preenchidos por eles.

Objeto de estudo: A aplicação inspecionada foi o portal da Sociedade Brasileira de Computação (SBC)[1], para o qual foi definido um roteiro de oito atividades.

Procedimento: Cada grupo recebeu um treinamento de uma hora que incluía conceitos sobre usabilidade e a técnica específica que o grupo deveria utilizar para realizar a inspeção. Cada inspetor pode realizar a detecção individual no horário que lhe fosse mais conveniente, respeitando-se apenas o prazo de dois dias.

Coleção de dados: Dezesseis inspetores enviaram as suas planilhas de anotação de discrepâncias. Quatro planilhas foram descartadas, pois os inspetores não informaram o tempo de inspeção ou não seguiram o roteiro. Foram analisadas seis planilhas do grupo WDP e seis do grupo WDP-RT. As listas de discrepâncias individuais foram integradas a uma única lista e uma equipe de três pesquisadores classificou as discrepâncias em únicas e duplicatas (discrepâncias equivalentes apontadas por mais de um inspetor). A reunião de discriminação contou com a participação destes três pesquisadores e de um diretor da SBC. Cada discrepância relatada foi avaliada e após a discussão da equipe, o diretor da SBC classificava a discrepância em defeito ou falso-positivo.[10]

1 - www.sbc.org.br

Resultados obtidos: Foram analisadas a eficácia e a eficiência da WDP-RT em comparação com a WDP. A eficácia é definida como a razão entre o número de defeitos detectados pela técnica e o número total de defeitos que nós conhecemos (73). A eficiência é definida como a razão entre o número total de defeitos (incluindo as duplicatas) e o tempo gasto na inspeção. A análise estatística apontou que a WDP-RT foi mais eficaz e tão eficiente quanto a WDP. A Tabela 1 sumariza os resultados.

Tabela 1: Resultados do 1º estudo de viabilidade

Técnica	Total de Defeitos	Eficácia	Defeitos por Inspetor	Média de tempo (h)	Defeitos / hora	DP2 (Defeitos/ Inspetor)	%DP2 (Defeitos/ Inspetor)
WDP	55	63,01	9,16	1,23	7,44	2,31	25,27%
WDP-RT	127	90,41	21,16	2,86	7,39	7,83	37%

Os inspetores também responderam um questionário para verificar sua opinião em relação à técnica. De acordo com os questionários, os inspetores consideram a WDP-RT útil para inspeções de usabilidade e de fácil utilização, e suas instruções são fáceis de compreender. Porém, dois inspetores relataram que algumas instruções eram semelhantes e metade dos inspetores ressaltou o alto tempo de inspeção.

4.1.2. Evolução da WDP-RT

Após o primeiro estudo de viabilidade, foi realizada a primeira revisão da técnica, que teve como principais objetivos procurar por instruções ambíguas e estudar uma forma de tornar a inspeção com WDP-RT mais rápida. Após análise detalhada da técnica, verificou-se que as instruções das perspectivas de Apresentação e Conceituação eram bem semelhantes, com o foco da instrução era direcionado para a Apresentação ou para a Conceituação. Mesmo que as instruções apresentem focos diferentes, o inspetor teria que executar uma mesma tarefa duas vezes para realizar a inspeção, sendo este um dos possíveis fatores que tornariam a inspeção com WDP-RT mais demorada.

Decidiu-se então pela junção das perspectivas de Apresentação e Conceituação, sendo as instruções semelhantes das duas perspectivas agrupadas, de forma que o inspetor possa verificar em paralelo a usabilidade em relação à Apresentação e à Conceituação. Desta forma, a inspeção com WDP-RT passa a ter duas etapas: na primeira parte, verifica-se a usabilidade em relação às perspectivas de Apresentação e Conceituação, e na segunda parte verifica a usabilidade em relação à Navegação. Foram revisadas as definições sobre o objetivo de cada uma das etapas, para deixar claro para o inspetor o objetivo de inspeção em paralelo das duas perspectivas. Estas modificações resultaram na segunda versão da WDP-RT (WDP-RT v2). O texto completo da WDPRT v2 está disponível em (GOMES e CONTE, 2009b).

4.1.3. Execução de estudos experimentais: Segundo estudo *in vitro*

No primeiro estudo de viabilidade, muitos participantes apresentavam bons conhecimentos em usabilidade, inspeções e desenvolvimento de aplicações Web. Para investigar se a WDP-RT ajuda inspetores com pouco conhecimento em inspeções de usabilidade, foi realizado um segundo estudo de viabilidade (Outubro/2009). O estudo constituiu-se de uma inspeção de usabilidade utilizando-se as técnicas WDP e WDP-RT v2, na qual foram avaliadas a eficácia e eficiência de ambas as técnicas.

Participantes: Doze alunos do curso de Ciência da Computação da UFAM, que foram divididos em dois grupos de seis inspetores (equipes WDP e WDP-RT) com níveis de experiência balanceados de acordo com as respostas de formulários de caracterização.

Objeto de estudo: A aplicação inspecionada foi o portal da Sociedade Brasileira de Computação (SBC)1, para o qual foi definido um roteiro de três atividades.

Procedimento: Cada grupo recebeu um treinamento de uma hora que incluía conceitos sobre usabilidade e a técnica específica que o grupo deveria utilizar para realizar a inspeção. Cada inspetor pode realizar a detecção individual no horário que lhe fosse mais conveniente, respeitando-se apenas o prazo de uma semana.

Coleção de dados: Os doze inspetores enviaram as suas planilhas de anotação de discrepâncias. As listas de discrepâncias individuais foram integradas a uma única lista por dois pesquisadores que classificaram as discrepâncias em únicas e duplicatas e tentavam associar as novas discrepâncias às discrepâncias da lista de discrepâncias do primeiro estudo de viabilidade. As discrepâncias que não puderam ser associadas foram analisadas e, após discussão, classificadas em defeito ou falso-positivo.

Resultados obtidos: Novamente, foram analisadas a eficácia e a eficiência da WDP-RT em comparação com a WDP. Os inspetores que utilizaram a WDP-RT encontraram mais defeitos que os inspetores que usaram a WDP (107 x 80), contudo tanto a eficácia quanto a eficiência foram similares às da WDP. A Tabela 2 sumariza os resultados.

Tabela 2: Resultados do 2º estudo de viabilidade

Técnica	Total de Defeitos	Eficácia	Defeitos por Inspetor	Média de tempo (h)	Defeitos / hora	DP2 (Defeitos/ Inspetor)	%DP2 (Defeitos/ Inspetor)
WDP	80	52,08	13,33	1,75	7,60	7,36	55,24%
WDP-RT	107	55,20	17,83	2,31	7,71	11,28	63,28%

Nos questionários de avaliação, os inspetores consideraram a WDP-RT útil para inspeções de usabilidade, mas dois deles relataram dificuldades em aplicar a técnica.

4.1.4. Execução de estudos experimentais: Estudo de caso *in vivo*

Depois dos estudos realizados na academia, decidiu-se realizar um estudo na Indústria (Dezembro/2009 e Janeiro/2010), para avaliar a adequação da WDP-RT ao ambiente industrial. Este estudo contou com a parceria da empresa FabriQ Informática Ltda[2].[11]

Participantes: Oito profissionais de informática (5 analistas de sistemas e 3 analistas de suporte) divididos em dois grupos de quatro inspetores (grupos A e B) com níveis de experiência balanceados de acordo com as respostas de formulários de caracterização.

Objeto de estudo: A aplicação inspecionada foi um módulo de uma aplicação desenvolvida pela empresa, para o qual foram definidos dois roteiros de inspeção.

Procedimento: Foi realizado um treinamento de uma hora e quinze minutos que incluía conceitos sobre usabilidade e a técnica WDP-RT. A detecção de defeitos foi realizada individualmente pelos inspetores, que tiveram um prazo de uma semana.

Coleção de dados: Sete inspetores enviaram as suas planilhas de discrepâncias (um dos inspetores não pode realizar a inspeção por motivos pessoais). Para cada roteiro de atividades foi gerada uma lista única contendo todas as discrepâncias relatadas nas planilhas de discrepâncias dos inspetores do roteiro. Foi realizada uma reunião de discriminação para cada roteiro, que contou com a participação dos inspetores que executaram a inspeção do roteiro e de dois responsáveis pelo sistema. Nesta reunião, as interações avaliadas foram re-executadas, e após a discussão da discrepância entre os inspetores e responsáveis, esta era classificada em defeito ou falso-positivo.

Resultados obtidos: Para verificar a adequação da WDP-RT ao ambiente industrial, primeiramente observamos o número de defeitos apontados por cada um dos inspetores.

Em relação a este objetivo, o resultado foi positivo, tendo os inspetores apontado um total de 84 problemas de usabilidade.

O indicador eficácia não pode ser calculado, visto que não conhecíamos o número total de discrepâncias do módulo inspecionado. Quanto ao indicador eficiência, este foi calculado como a razão entre o número de defeitos por tempo de inspeção. Em média, os inspetores gastaram 1 hora e 32 minutos na detecção. Assim, a *Eficiência na etapa de Detecção* foi de 7,81defeitos / hora por inspetor.

Para calcular o indicador *Esforço nas etapas de Detecção e Discriminação*, é necessário considerar o tempo gasto na atividade de discriminação (foram realizadas duas reuniões de discrimação, uma com duração de 1 hora e a outra com duração de 1 hora e 40 minutos). Neste estudo de caso, o custo da inspeção se mostrou baixo, pois o esforço médio de um inspetor, somando o esforço na atividade de detecção (1 hora e 32 minutos) e discriminação (1 hora e 20 minutos) foi de 2 horas e 52 minutos.

O terceiro indicador calculado foi o *Grau de Apreensibilidade*. Este indicador denota o quão fácil é aprender um método (BOLCHINI e GARZOTTO, 2007). De forma análoga à (BOLCHINI e GARZOTTO, 2007), este indicador foi verificado através dos fatores tempo gasto no treinamento da técnica e percepção sobre a dificuldade em aplicar a técnica. Em relação ao primeiro fator, o tempo de treinamento da WDP-RT foi de apenas 1 hora e 15 minutos. Sobre o segundo fator, foram realizadas entrevistas semi-estruturadas com os inspetores, e todos consideram que a WDP-RT facilita a inspeção. Ela ajuda a melhorar os conhecimentos sobre usabilidade, e eles presumem que isto os ajudará no desenvolvimento de sistemas com melhor usabilidade.

Eles também consideram que a WDP-RT poderia ser utilizada para a realização de inspeções nos produtos desenvolvidos pela empresa.

2 - www.fabriq.com.br

Sobre os benefícios obtidos para o software e a organização (FabriQ), além da melhora da usabilidade do módulo inspecionado, a inspeção proporcionou uma melhoria do padrão de interação adotado no sistema, o que acarreta automaticamente benefícios em relação à usabilidade de todos os módulos da aplicação.

4.2.Artigo publicado

O artigo "WDP-RT: Uma técnica de leitura para inspeção de usabilidade de aplicações Web" (GOMES *et al.* 2009) foi apresentado no VI Experimental Software Engineering Latin American Workshop (ESELAW 2009), realizado em São Carlos(SP).

4.3.Outros resultados: Recurso Humano Capacitado

Dissertação de mestrado de Marcos Gomes: Mestrado em Informática. Universidade Federal do Amazonas – UFAM.

5. Aplicabilidade dos resultados: Benefícios para a Indústria

No cenário de desenvolvimento de aplicações de software convencionais, é possível constatar aumentos substanciais de qualidade e redução de defeitos devido à adoção de inspeções e outras técnicas de revisão (TRAVASSOS *et al.* 1999).

As inspeções melhoram a produtividade uma vez que os defeitos são encontrados quando são mais fáceis e mais baratos para corrigir. No estudo de caso na Indústria relatado, o custo da inspeção se mostrou baixo, não sendo necessária a contratação de um especialista em usabilidade e nem utilização de ambiente especial.

Os profissionais da empresa precisaram apenas de um treinamento de 1 hora e 15 minutos para realizar a inspeção, e gastaram em média 2 horas e 52 minutos na detecção e discriminação. Desta forma, a empresa foi capaz de realizar a inspeção investindo somente 4 horas e 07 minutos dos profissionais disponibilizados para esta atividade.

Além dos benefícios já citados, a utilização da WDP-RT proporciona direcionamento para a execução da inspeção de usabilidade, tornando a inspeção mais fácil para os inspetores com pouco conhecimento em usabilidade. Segundo os inspetores que participaram do estudo de caso na indústria, a técnica ajudou-os a melhorar os seus conhecimentos sobre usabilidade, o que se espera contribuir para que os próximos aplicativos sejam desenvolvidos com melhor usabilidade.

6. Características inovadoras

Como característica inovadora deste projeto, temos o desenvolvimento de uma técnica de leitura baseada em perspectivas específica para inspeções de usabilidade de aplicações de software Web, a WDP-RT. Uma técnica de leitura proporciona foco, que é usado para guiar um inspetor durante a atividade de detecção de problemas de usabilidade, visando reduzir as dificuldades de inspeção dos inspetores com menor conhecimento sobre avaliação de usabilidade (CONTE, 2009).

A construção e avaliação da técnica WDP-RT se fundamentou em uma abordagem baseada em experimentação, através de uma série de estudos experimentais na academia e na indústria. Além da avaliação experimental, esta abordagem apóia a transferência da nova tecnologia para a indústria de software Web de forma segura, maximizando o benefício de adoção da tecnologia de software em questão.

7. Conclusões e trabalhos futuros

Este artigo apresentou a WDP-RT, uma técnica de leitura para inspeção de usabilidade de aplicações Web desenvolvida neste ciclo do EQPS, e os estudos experimentais realizados para a transferência segura da tecnologia da academia para a Indústria. Os estudos realizados apontam que a WDP-RT facilita a inspeção de usabilidade e, embora os resultados de um único estudo de caso não possam ser generalizados para outros contextos, o resultado qualitativo e quantitativo deste estudo é um indicador da viabilidade de utilização pelos próprios membros da equipe de um projeto como inspetores em uma avaliação de usabilidade. Espera-se com estes resultados incentivar a indústria de software Web a realizar avaliações de usabilidade com maior freqüência.

Como trabalhos futuros, temos a evolução da WDP-RT para uma nova versão, considerando os resultados quantitativos e qualitativos obtidos nos estudos realizados, a fim de tornar a inspeção mais ágil. Para avaliar a utilização da técnica pelos inspetores, será realizado um estudo de observação, já com a nova versão da WDP-RT. Este estudo visa coletar dados de como os inspetores aplicam a técnica. Também serão realizados novos estudos de casos na Indústria, para avaliar a adequação da WDP-RT ao ambiente industrial, e em especial sua utilização por inspetores novatos.

Agradecimentos

Os autores agradecem a todos que participaram dos estudos experimentais e ao projeto SAUIM, ao CNPq, à FAPERJ, à FAPEAM e à Trópico Telecomunicações Avançadas pelo apoio financeiro. Também agradecemos especialmente à empresa FabriQ Informática Ltda. pelo apoio no estudo de caso na indústria.

Referências Bibliográficas

BLACKMON, M. H., POLSON, P. G., KITAJIMA, M., LEWIS, C., 2002. "Cognitive walkthrough for the web". In: Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves, v. 5 (1), pp. 463 - 470, Minneapolis, Minnesota, USA.

BOLCHINI, D., GARZOTTO, F., 2007. "Quality of Web Usability Evaluation Methods: An Empirical Study on MiLE+". In: International Workshop on Web Usability and Accessibility (IWWUA) WISE 2007 Workshops, v. LNCS 4832, pp. 481 - 492, Nancy, France.

CONTE, T., 2009. “Técnica de Inspeção de Usabilidade Baseada em Perspectivas de Projeto Web”. Rio de Janeiro: UFRJ/COPPE 194 p. Tese (Doutorado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação.

CONTE, T., MASSOLAR, J., MENDES, E., TRAVASSOS, P. G. H., 2009a. “Web Usability Inspection Technique Based on Design Perspectives.” IET Software Journal, v. 3, n. 2, pp. 106-123.

CONTE, T., VAZ, V., MASSOLAR, J., MENDES, E., TRAVASSOS, G. H., 2009b. "Improving a Web Usability Inspection Technique using Qualitative and Quantitative Data from an Observational Study". In: XXIII Simpósio Brasileiro de Engenharia de Software - SBES 2009, v. 1, pp. 227 - 235, Fortaleza, CE FERREIRA, S. B. L., LEITE, J. C. S. P., 2003. “Avaliação da Usabilidade em Sistemas de Informação: O Caso do Sistema Submarino”. Revista de Administração Contemporânea - RAC, Curitiba, PR, v. 7, n. 3, p. 115-136.

GOMES, M., CONTE, T., 2009. “WDP-RT: Uma técnica de leitura para inspeções de usabilidade de aplicações Web”. Relatório técnico RT-DCC-ES001/2009. DCC/ UFAM.

GOMES, M., CONTE, T., 2009b. “WDP-RT v2: Uma técnica de leitura para inspeções de usabilidade de aplicações Web”. Relatório técnico RT-DCC-ES002/2009. DCC/ UFAM.

GOMES, M., SANTOS, D. V., CHAVES, L., CASTRO, A., VAZ, V. T., SOARES, A., TRAVASSOS, G. H., CONTE, T., 2009. "WDP-RT: Uma técnica de leitura para inspeção de usabilidade de aplicações Web". In: VI Experimental Software Engineering Latin American Workshop (ESELAW 2009), v. 1, pp. 124-133, São Carlos, São Paulo.

HITZ, M., LEITNER, G., MELCHER, R., 2006. "Usability of Web Applications".In: Kappel, G., Pröll, B., Reich, S., Retschitzegger, W. (eds), Web Engineering: The Discipline of Systematic Development of Web Applications, Chapter 11, John Wiley \& Sons.

JURISTO, N., MORENO, A., SANCHEZ-SEGURA, M.-I., 2007. “Guidelines for Eliciting Usability Functionalities”. IEEE Transactions on Software Engineering, v. 33, n. 11, pp. 744-758.

MAFRA, S., BARCELOS, R., TRAVASSOS, G. H., 2006. “Aplicando uma metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software”. In: Proceedings of the 20th Brazilian Symposium on Software Engineering (SBES 2006), v. 1, pp. 239 – 254.

NIELSEN, J., 1993. “Usability Engineering”. Academic Press, Cambridge, MA.

NIELSEN, J., 1994. “Heuristic evaluation”. In: Jacob Nielsen, Mack, R. L. (eds), Usability inspection methods, Heuristic Evaluation, New York, NY, John Wiley & Sons, Inc.

PRATES, R. O., BARBOSA, S. D. J., (2003). “Avaliação de Interfaces de Usuário – Conceitos e Métodos”. In: Coello, J. M. A., Fabbri, S. C. P. F. (eds), Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação, Capítulo 6, Campinas.

SHULL, F., CARVER, J., TRAVASSOS, G. H., 2001. “An empirical methodology for introducing software processes”. ACM SIGSOFT Software Engineering Notes, v. 26, n. 5, pp. 288-296.

TRAVASSOS, G. H., SHULL, F., FREDERICKS, M., BASILI, V., 1999. "Detecting defects in object-oriented designs: using reading techniques to increase software quality." ACM SIGPLAN Notices, v. 34, n. 10, pp. 47-56.

TRAVASSOS, G. H., SHULL, F., CARVER, J., BASILI, V., 2002. Reading Techniques for OO Design Inspections. University of Maryland.

TRAVASSOS, G. H., BARROS, M., 2003. "Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering". In: Proceedings of the 2nd Workshop in Workshop Series on Empirical Software Engineering (WSESE 2003), pp. 117-130, Rome.

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 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 Morais 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 Quites 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 Morais 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: Qualiti 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 Morais 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 SOFTEX 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, Káthia 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: Káthia 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