

Tecnologia Software

[6.01] Adaptação de um Software de Gerência de Projetos de Código Aberto para Atendimento dos Resultados Esperados do Nível G do MPS.BR

Entidade: 1Universidade Federal de Santa Catarina - Florianópolis - SC - Brasil. e
2Universidade Do Vale do Itajaí - São José - SC - Brasil.

Autores: Jean Carlo Rossa Hauck¹ Christiane Gresse von Wangenheim,¹ 2 Marcello Thiry²

Resumo. *Este artigo apresenta os resultados do projeto voltado a adaptação do software dotProject para suportar o atendimento dos resultados esperados do processo de Gerência de Projetos do nível G do MPS.BR. São descritas as principais melhorias no software e a sua compatibilidade com o modelo de referência MR-MPS, bem como a aplicação da nova versão do sistema em organizações de desenvolvimento de software.*

1. Introdução

A gerência de projetos de software é um dos primeiros passos normalmente tomados para iniciar a melhoria dos processos de software de uma organização [SEI 06] [SOF 07]. O PMI define gerência de projetos como “a aplicação de conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos” [PMI 04]. Existem atualmente diversos modelos, normas e abordagens para a gerência de projetos, tais como: [HUG 02] [MAR 06] [GAL 05] [MUR 00] [PMI 04] [ISO 97] [ANS 98]. Ainda assim, a gerência de projetos de software é um processo complexo dificultado pelas características inerentes ao desenvolvimento de software, tais como a invisibilidade e intangibilidade do produto desenvolvido [JAL 00]. Para suportar a implantação de processos de gerência de projetos na prática, modelos de melhoria como, por exemplo, o MPS.BR [SOF 07], identificam os requisitos mínimos que processos como a gerência de projetos devem atender.

Estes modelos, normas e abordagens não exigem que sejam utilizadas ferramentas para a gerência de projetos, entretanto o uso de ferramentas pode facilitar o estabelecimento do processo pela automatização de tarefas manuais, suportando atividades e melhorando sua eficiência e eficácia. Atualmente existem inúmeras ferramentas para gerência de projetos, desde soluções completas até simples sistemas para registro de tarefas. As ferramentas *open-source* representam uma alternativa para que sejam completadas as funcionalidades necessárias que não estejam disponíveis em uma versão original.

O software *dotProject* é uma ferramenta web de gerência de projetos desenvolvido em PHP, gratuita e de código aberto, sob a licença GPL (*GNU General Public License*).

Dentre as ferramentas de código aberto, com estas características, é uma das mais largamente utilizadas no mundo [SOU 07] para gerência de projetos. Conta com uma comunidade ativa de desenvolvedores [DOT 08] [SOU 07] que suporta o seu desenvolvimento e compartilham os incrementos desenvolvidos, o que gera uma grande quantidade de extensões que possibilitam ampliar as funcionalidades existentes na ferramenta [DOT 08].

Os módulos básicos do *dotProject* compreendem funcionalidades específicas para a gerência de projetos, incluindo: registro de projetos, cadastro de empresas, tarefas, usuários, recursos, arquivos do projeto, WBS, registro do progresso das atividades, esforço, etc. Entretanto, apesar de ser bastante utilizado e completo, as funcionalidades oferecidas atualmente no sistema, incluindo os *add-ons* disponíveis, não atendem completamente os resultados esperados do MR-MPS [SOF 07] do MPS.BR para o processo de Gerência de Projetos.

A partir da necessidade de uma ferramenta gratuita que atenda os resultados esperados do MR-MPS, o grupo CYCLOPS [CYC 07], em parceria com o LQPS/UNIVALI [LQPS 07], realizou o presente projeto. O CYCLOPS é um grupo de pesquisas da Universidade Federal de Santa Catarina (UFSC), especializado na pesquisa e desenvolvimento de tecnologias inovadoras na área de processamento de imagens médicas.

2. Objetivos e Justificativa

Este projeto tem por principal objetivo a adaptação de um software de gerência de projetos livre e de código aberto para suportar o atendimento dos resultados esperados para o processo de Gerência de Projetos do nível G do MPS.BR.

Espera-se que a adaptação desta ferramenta, disponível publicamente, resulte numa alternativa de baixo custo para suportar a implementação do processo de gerência de projetos suportado por uma ferramenta de livre e código aberto alinhada aos requisitos do modelo MPS.BR. Entretanto a adoção de uma ferramenta em si não garante a satisfação destes requisitos, especialmente no caso do processo de gerência de projetos, ela pode substancialmente suportar e facilitar a implantação deste processo na prática.

Outra vantagem da adaptação de uma ferramenta especialmente voltada para gerência de projetos (incluindo interfaces para outras ferramentas comuns) ao invés de utilização de uma ferramenta integrada propicia que organização não necessite abandonar as ferramentas que porventura já estejam sendo utilizadas com sucesso para o suporte a outros processos. Neste sentido, ferramentas gratuitas e de código aberto oferecem, além de custo inicial menor, a possibilidade de customização ao processo da organização e de alinhamento da ferramenta às melhores práticas dos modelos de referência.

Com isto espera-se contribuir positivamente no número de organizações que estabelecem um processo de gerência de projetos e, conseqüentemente, na melhoria da gerência de projetos. Por fim, espera-se auxiliar no aumento da qualidade dos projetos principalmente em termos de custos e prazos e assim também na competitividade das empresas.

Além disso, como a ferramenta já está pré-adaptada ao MPS.BR, também será facilitada uma avaliação oficial do processo de gerência de projetos suportado pela ferramenta.

3. Metodologia de Execução

Inicialmente foi realizada uma análise de ferramentas para gerência de projetos e foi realizado um estudo comparativo entre os softwares livres mais utilizados atualmente para gerência de projetos [SOU 07]. Para este fim foi constituído um grupo de estudos incluindo membros do Laboratório de Qualidade e Produtividade de Software da UNIVALI (implementadores MPS.BR) e do grupo de pesquisas CYCLOPS da UFSC.

Para este estudo foi definido um conjunto de critérios alto-nível com base na teoria da gerência de projetos [PMI 04] e nos resultados esperados do processo de gerência de projetos do modelo MPS.BR V 1.1. Utilizando estes critérios as ferramentas foram avaliadas e comparadas. Nesta avaliação o *dotProject* alcançou o maior cobertura de critérios e foi escolhido como base para realizar as adaptações.

Em seguida foi realizada a análise de requisitos para extensão do *dotProject*. Para isto foi realizada uma avaliação detalhada das funcionalidades da ferramenta em relação aos resultados esperados para o processo de Gerência de Projetos do nível G do MPS.BR já na versão 1.2 e de requisitos com base na teoria da gerência de projetos do PMBOK [PMI 04]. Os requisitos foram formulados conformes à norma IEEE Std 830 [IEEE 98].

Estes requisitos foram refinados em casos de uso. A partir disto foi realizada uma priorização dos casos de uso e eles foram agrupados em 4 iterações de implementação,

que foram então detalhadamente planejadas. Todos os resultados da análise de requisitos foram documentados na ferramenta Enterprise Architect (EA) [SPA 07].

A implementação das iterações foi realizada por meio de extensões ao código original do *dotProject* em PHP, utilizando o framework de desenvolvimento disponibilizado pelo *dotProject*, de forma a manter a compatibilidade com a versão standard.

Testes de unidade foram então realizados pelos programadores e testes de integração foram realizados pelos membros do SEPG do CYCLOPS com base nos requisitos definidos, com seus resultados também documentados no EA.

Assim que cada iteração foi sendo finalizada, esta versão foi colocada em produção para validação no grupo CYCLOPS [CYC 07]. Para facilitar o uso do sistema foi criada uma secção no WIKI da organização do grupo explicando passo a passo o uso da ferramenta dentro do processo específico da organização com hiperlinks para as respectivas secções da ferramenta. Feedback sobre a usabilidade e erros da ferramenta foram sistematicamente relatados através da WIKI da organização e incluídos nas iterações seguintes da implementação.

4. Resultados Obtidos

O principal resultado obtido deste projeto é uma adaptação de um software livre de gerência de projetos alinhado ao nível G do MPS.BR para o processo de Gerência de Projetos, publicamente disponível para estudo e download em:
<http://projetos.telemedicina.ufsc.br/demo/>.

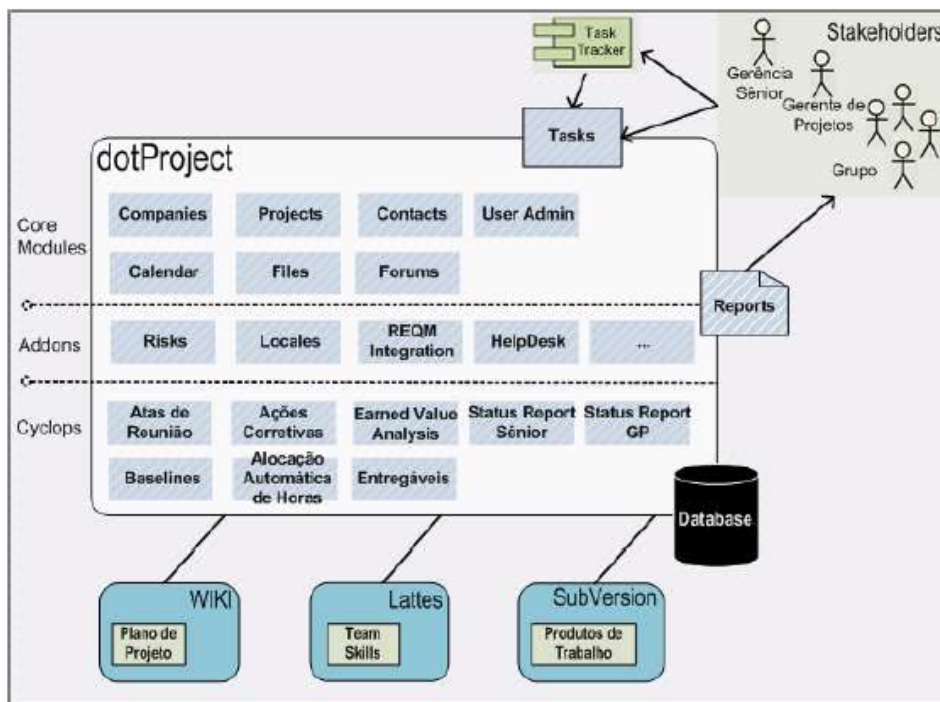


Figura 1: Arquitetura da customização do *dotProject*

Conforme demonstra a figura 1, as principais funcionalidades implementadas foram:

Novo módulo de Monitoração e Controle

Para contemplar os resultados esperados relativos à monitoração e controle dos projetos, foram implementadas diversas funcionalidades com base nos indicadores de valor agregado (EAV) [ANS 98] como: suporte a reuniões de status do projeto, incluindo registro de atas, check-list de monitoração, relatórios de status (vide figura 2), suporte à

gerência de ações corretivas e suporte a reuniões de monitoração com a gerência sênior (vide figura 3).

Entretanto, para o cálculo do EAV algumas variáveis necessárias não estavam presentes ou não eram armazenadas de forma satisfatória na versão padrão do *dotProject*. Neste sentido, algumas alterações foram necessárias na forma de alocar horas, que passou a ser semi-automatizada em homens/hora por tarefa e por recurso humano. A partir destas alterações foi possível calcular os indicadores de valor agregado: EAV, CPI, SPI e EAC, e exibi-los nos relatórios de monitoração da gerência de projeto (figura 2), Sênior (figura 3) e da Equipe do Projeto. Estes valores são armazenados no momento da geração dos citados relatórios e não são mais recalculados em visualizações posteriores destes relatórios, de forma a evitar atualizações dos indicadores após possíveis atualizações do cronograma.

Algumas dificuldades foram encontradas ao tentar aplicar a técnica EAV diretamente, como, por exemplo, quando ocorriam eliminações de tarefas no replanejamento, o que gerava uma comparação com o realizado, gerando o valor zero para o valor agregado e conseqüentemente uma divisão por zero no cálculo do CPI, SPI e EAC. Estas dificuldades foram superadas pelo analista e pelos desenvolvedores utilizando-se o registro de *baselines*.

O recurso de armazenamento de *baselines* de planejamento permite a monitoração do projeto, utilizando a técnica EAV, em comparação com qualquer das *baselines* armazenadas ou em relação ao cronograma atual.

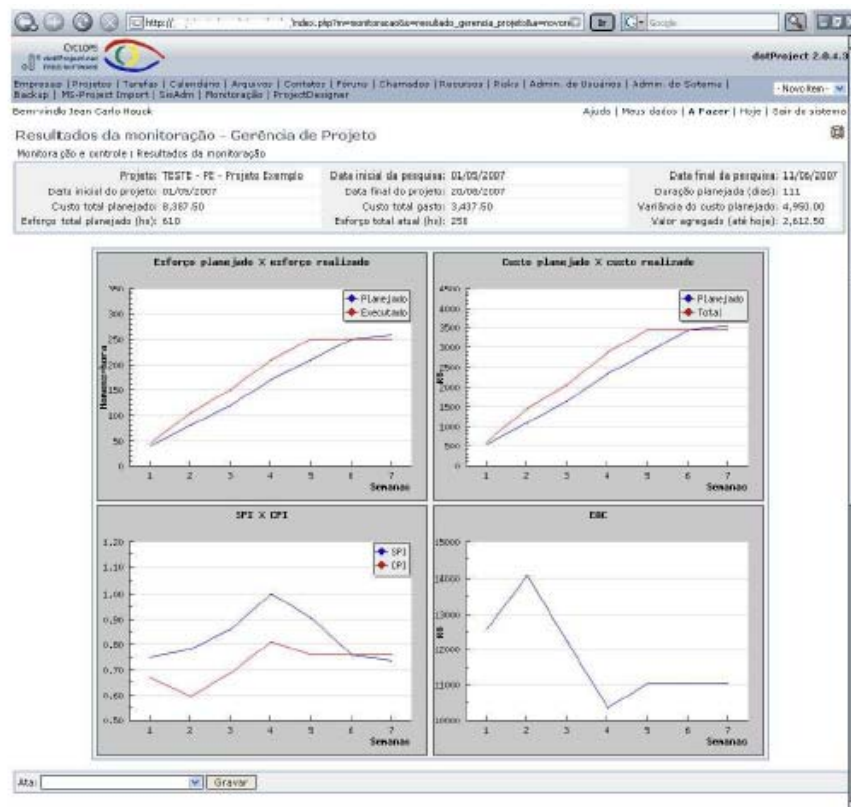


Figura 2: Relatório de Status para o Gerente de Projeto

Novo relatório

Monitoração : Gerência Sênior : Monitoração e Controle

Dados: Exibir: ☐ Somente Projetos Ativos ☐ Somente Projetos Não Finalizados

Gerente:

Usuários:

- thiago
- Thry?
- tiago
- tomazella
- tondeilo
- vieira
- vitorjunior
- wallauer
- wilmar
- zuri

Participantes da Reunião:

- jeancarlo(Responsável)
- Greise
- ma
- awagnerh
- andrade

Itens de Monitoração

- ☐ A utilização e comunicação dos Dados está segundo plano?
- ☐ O Cronograma está sendo realizado de acordo com o plano?
- ☐ O Envolvimento dos Interessados está segundo o plano?
- ☐ Ocorreram alterações nos Riscos?
- ☐ Ocorreram Riscos?
- ☐ Os Custos estão sendo realizados de acordo com o plano?
- ☐ Os Marcos estão sendo realizados conforme o plano?
- ☐ Os Recursos estão sendo utilizados de acordo com o plano?

Pesquisa

P.	%	Projeto	Gerente	SPB	CPI	Situação	Bas.	Observações	Ata
-	57.2	BuissFilms	Jean Carlo Hauck	0.22	0.47	N OK	0		Incluir
	69.6	Video ABC	Jean Carlo Hauck	0.83	2.40	OK	4		Incluir
-	38.2	PE - Projeto Exemplo	Jean Carlo Hauck	0.31	0.76	N OK	0		Incluir

Figura 3: Suporte a reuniões de monitoração com a Gerência Sênior

Gerencia de Produtos de trabalho e itens de entrega/interface para sistemas de gerencia de configuração

Foi adicionada a funcionalidade de gerência de produtos de trabalho e itens de entrega, de forma a contemplar a gerência de dados do projeto. A versão original do *dotProject* já permite o *upload* de arquivos gerados pelo projeto, mas não o planejamento dos dados do projeto. A figura 4 apresenta um dado do projeto sendo registrado.

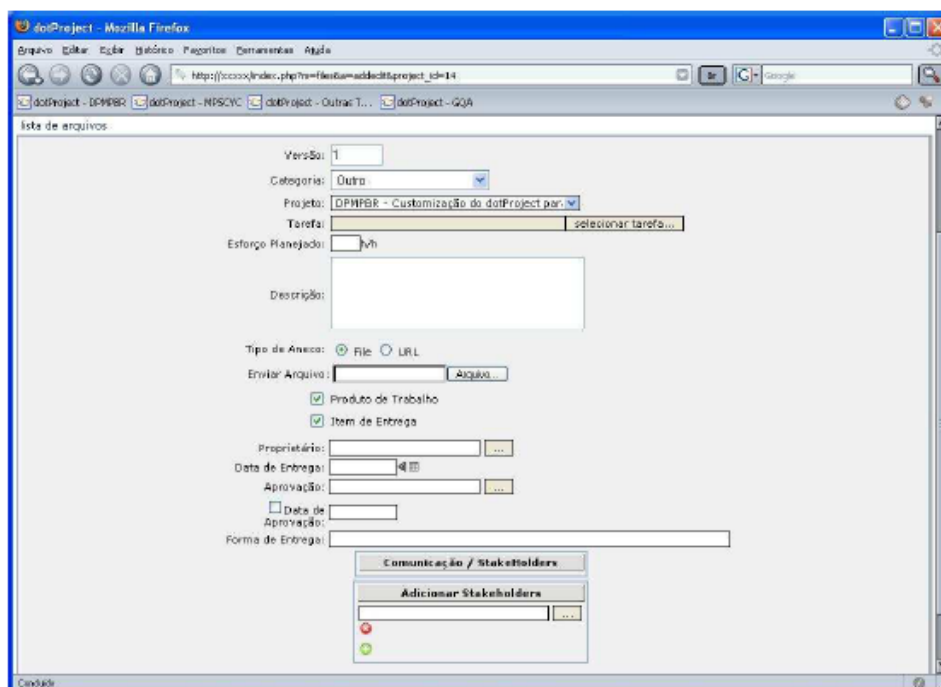


Figura 4: Produtos de Trabalho e Itens de Entrega

Outros Resultados Obtidos:

No âmbito da organização foram obtidos outros resultados interessantes a partir deste projeto, especialmente no que se refere à adesão ao processo definido. Inicialmente, com a utilização da versão padrão do *dotProject* e realizando-se o cálculo manual dos indicadores de valor agregado houve considerável resistência dos participantes dos projetos o que dificultou consideravelmente a aderência dos projetos ao processo.

Somente a partir da instalação do primeiro release do *dotProject* customizado a organização passou a seguir sistematicamente o processo auxiliado pelas novas funcionalidades implementadas na ferramenta.

Os principais benefícios percebidos pelos gerentes de projeto são principalmente referentes à maior visibilidade do status do projeto e a sua evolução, especialmente devido à automatização na geração de relatórios e o suporte às reuniões de monitoração.

Também foi percebido pela gerência sênior o suporte às decisões estratégicas, possibilitando a priorização de projetos e o alinhamento de projetos e recursos da organização.

Também pode ser observado que a automação, mesmo que parcial, de tarefas e a integração colaborativa de dados de diversos *stakeholders* do projeto, contribuiu na melhoria da eficiência do processo. A ferramenta customizada ao processo da organização e alinhada a modelos de melhoria também impõe uma maior institucionalização do processo definido.

Atualmente todos os projetos da organização estão sendo planejados, monitorados e controlados diretamente no *dotProject*, atingindo em média, 60% de alinhamento aos resultados esperados do MPS.BR, segundo auditorias internas. A resistência inicial no uso da nova ferramenta foi superada com treinamentos técnicos e motivacionais, reduzindo até o momento 20% as não-conformidades inicialmente observadas.

5. Aplicabilidade dos Resultados

No contexto nacional, onde a maioria das organizações de desenvolvimento de software é Micro ou Pequenas Empresa (MPE) [MCT 05], que tipicamente apresenta processos bastante informais e imaturos, as iniciativas de melhoria, em geral, são mais enfocadas em atingir níveis de maturidade iniciais (nível G do MPS-BR). Neste sentido, o principal enfoque deste nível de maturidade, concentra-se na gerência dos projetos e dos requisitos de software, para melhorar a qualidade do processo da organização. As organizações de desenvolvimento de software, em geral, trabalham orientadas a projetos, onde são implementados e/ou customizados os produtos de software. Desta forma, melhorar a qualidade do processo e do produto de software de MPEs passa, normalmente, pela gerência do projeto de software.

A existência de um software livre que forneça um suporte já pré-adaptado aos resultados esperados do modelo MPS.BR, facilitando e automatizando o processo de gerência de projetos, tende a auxiliar na melhoria deste processo nas organizações brasileiras.

Assim, criamos uma solução de baixo custo em alinhamento com o modelo MPS.BR.

Atualmente, a customização da ferramenta esta sendo utilizada no grupo de pesquisas CYCLOPS da Universidade Federal de Santa Catarina para gerência de todos os seus projetos de desenvolvimento de software e se mostrou adequada, suportando de forma efetiva e eficiente o processo de gerência de projetos.

6. Características Inovadoras

A principal inovação obtida com o resultado deste projeto está na disponibilização para a comunidade de desenvolvimento de software a customização de uma ferramenta de código aberto e livre já pré-adaptada os resultados esperados do nível G do MPS.BR para o processo de Gerência de Projetos.

Atualmente, não existem softwares livres que completamente suportam todos estes resultados. Também não são encontrados relatos de experiências obtidas na aplicação de software livre para gerência de projetos de software em pequenas organizações, alinhado ao MPS.BR.

7. Conclusão e Perspectivas Futuras

Neste trabalho é apresentada a evolução da ferramenta *dotProject* para o alinhamento ao processo de Gerência de Projetos do nível G do MPS.BR.

O principal benefício obtido por meio deste projeto e a disponibilização pública e livre desta customização, oferecendo uma solução de baixo custo e alinhada ao modelo de melhoria de processo de software nacionalmente reconhecido.

Atualmente a customização da ferramenta esta sendo validada por um grupo de pesquisas CYCLOPS da UFSC. Com base nestes resultados e feedback estão sendo planejadas futuras evoluções, melhorando os módulos de planejamento e de monitoração e controle, incluindo também funcionalidades para iniciação e finalização de projetos.

Também estão sendo previstos a ampla divulgação da customização e a sua aplicação e avaliação em outras organizações de software no Brasil.

7. Referências Bibliográficas

[ANS 98] ANSI/EIA 748, "A Standard for Earned Value Management Systems", ANSI, 1998.

[CYC 07] CYCLOPS, Disponível em: < [http://www.cyclops .ufsc.br](http://www.cyclops.ufsc.br)>, Acesso em: 18/02/2008

[DOT 08] DOTPROJECT, disponível em < <http://www.dotprojecct.net> >. Acesso em: 23/11/2007

[GAL 05] GALORATH Mike Ross Galorath Inc. "Integrating Three Level 2 CMMI™ Process Areas: Closing the Loop on Software Project Management" IEEEAC paper 1410, Revision B,

December 19, 2005

[HUG 02] HUGHES B., M. Cotterell, "Software Project Management", 3rd Edition, McGraw-Hill, 2002.

[IEEE 97] IEEE Recommended Practice for Software Requirements Specifications. Software Engineering Standards Committee of the IEEE Computer Society IEEE Std 830, 1998

[ISO 97] ISO/IEC - International Organization for Standardization. "ISO/IEC 10006: Quality Management – Guidelines to Quality in Project Management", ISO/IEC International Standard, 1997.

[JAL 00] JALOTE Pankaj, "CMM in Practice: Processes for Executing Software Projects at Infosys", Addison Wesley Longman, 2000.

[LQPS 07] LQPS - Laboratório de Qualidade e Produtividade de Software, disponível em <www.univali.br/lqps>. Acesso em: 23/11/2007

[MAR 06] MARTINS José Carlos Cordeiro, "Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML", Brasport, 3. Ed, Rio de Janeiro, 2006.

[MCT 05] MCT - Ministério da Ciência e Tecnologia. "Qualidade e Produtividade no Setor de Software Brasileiro, Resultados da Pesquisa 2005" (resultados antecipados recebidos por email do MCT).

[MUR 00] MURCH R., "Project Management: Best Practices for IT Professionals", Prentice-Hall, 2000.

[PMI 04] PMI - Project Management Institute, "Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos Terceira edição (Guia PMBOK®)", PMI, Pennsylvania, 2004.

[SEI 06] SEI - Software Engineering Institute, "Capability Model Integration, version 1.2 – CMMI® for Development, Version 1.2, 2006.

[SOF 07] SOFTEX , "MPS.BR – Melhoria de Processo do Software Brasileiro", Guia Geral, Versão 1.2 , 2007.

[SOU 07] SOURCEFORGE, disponível em <<http://sourceforge.net>>. Acesso em: 09/11/2007.

[SPA 07] SPARKSYSTEMS, disponível em <<http://www.sparxsystems.com.au>>. Acesso em: 02/11/2007.

[6.03] Do MPS.BR, nível G, para o CMMI, nível 2

Entidade: 1IVIA - Av. Senador Virgílio Távorá 1701, 2º andar - Fortaleza - CE

Autores: Edgy Paiva¹, Fábio de Castro Leite¹, Josyleuda Oliveira¹, Karlson Oliveira¹,
Luciana Almeida¹, Mariângela Bezerra¹ -
{edgy.paiva,fabio.leite,josy.oliveira,karlson.oliveira,luciana.almeida
mariangela.bezerra}@ivia.com.br

Abstract. *This work presents how easy to IVIA became CMMI level 2 was after its MPS.BR level G appraisal. Also are showed CMMI level 2 appraisal results and the benefits of using software quality processes in the organizational appraisal.*

Resumo. *Este trabalho mostra as facilidades que uma empresa como a IVIA, empresa de software cearense, pode encontrar, para obter o nível 2 do CMMI, após a avaliação do MPS.BR, nível G. Aqui também é exibido o resultado da avaliação do CMMI e ainda os ganhos do uso da qualidade de software para a melhoria da organização.*

1. Introdução

A IVIA é uma instituição de desenvolvimento de software que atua no mercado desde 1996. É certificada ISO 9001:2000 desde novembro de 2003.

Em dezembro de 2006, a IVIA foi avaliada e aprovada no MPS.BR nível G, que trata do processo de gerência de projetos e do processo de gerência de requisitos.

A partir de Janeiro de 2007, a organização, procurando a melhoria contínua, incrementou seus processos, passando a trabalhar com os processos do CMMI nível 2.

Inicialmente, utilizamos os resultados da avaliação do MPS.BR nível G como diagnóstico, a fim de detectar quais eram os ajustes necessários no atual processo da IVIA para se adequar ao CMMI nível 2. Estes ajustes foram implementados e instanciados para os projetos.

Em dezembro de 2007, a IVIA foi avaliada e aprovada no CMMI nível 2.

Este trabalho descreve as facilidades que uma empresa como a IVIA pode encontrar obtendo o nível 2 do CMMI, somente após a avaliação do MPS.BR, nível G, mostrando claramente a evolução dos processos da IVIA. O trabalho foi dividido nas seguintes seções: na seção 2, são descritos os objetivos e justificativa do projeto. A seção 3 apresenta a metodologia de execução do projeto, ou seja, como ocorreu a implantação do nível 2 do CMMI na IVIA. Os resultados obtidos com o projeto são apresentados na seção 4. A seção 5 apresenta a aplicabilidade dos resultados do projeto.

Na seção 6, são descritas as características inovadoras do projeto. A conclusão do trabalho e as perspectivas futuras para a empresa estão na seção 7. As referências bibliográficas são exibidas na seção 8.

2. Objetivos e Justificativa

O principal objetivo da IVIA com esse projeto, não era só atingir o nível 2 na avaliação do CMMI, mas, principalmente, melhorar continuamente seus processos de software, para em seguida expandir seus negócios no mercado internacional, como também tornar-se uma empresa mais competitiva nas licitações do governo brasileiro. Como a IVIA obteve, através do MPS.BR, uma melhoria significativa em seus processos, buscou-se, à partir do CMMI, nível 2, obter um controle mais efetivo do desenvolvimento de software da organização. A IVIA tem consciência de que, melhorando seus processos ela estará melhorando seus produtos e, com isso, aumentando a satisfação de seus usuários.

3. Metodologia de Execução

Inicialmente foi definida a equipe interna na IVIA, responsável pela definição dos processos de gerência de projeto, gerência de requisitos, gerência de configuração, garantia da qualidade, medição e análise, lembrando que a IVIA já apresentava, antes de iniciar os trabalhos do CMMI, uma metodologia baseada no PMBOK (2004), na ISO 9001 (2000) e MPS.BR nível G. Era necessário detectar primeiramente quais pontos dessa metodologia não atendiam ao nível 2, do CMMI. Em seguida, foi contratada uma empresa de consultoria em CMMI.

Assim, de posse das melhorias sugeridas na avaliação do MPS.BR e avaliação por parte da consultoria, foi elaborado um planejamento da execução dos passos para a implantação do CMMI, nível 2, na organização.

Participaram deste trabalho os membros da equipe de SEPG, um consultor sênior, um especialista em medição, a equipe de garantia da qualidade, um especialista em gerência de configuração e o patrocinador do projeto.

Esse trabalho inicial foi de extrema importância para direcionar os esforços nos pontos em que a IVIA realmente precisava melhorar e evitar desperdício de tempo da equipe interna e das horas de consultoria. E o fato de já ter passado pela avaliação do MPS.BR, nível G, facilitou bastante nesta fase inicial.

3.1. Diagnóstico da Gerência de Projetos

Foram identificadas as seguintes oportunidades de Melhoria a serem trabalhadas no processo atual de Gerência de Projetos, na IVIA:

- A representação gráfica do Ciclo de Vida é sempre a mesma, independente do modelo de ciclo de vida do projeto. Deveria ser criada uma figura para os diferentes ciclos de vida.
- No plano do projeto devem constar as responsabilidades pela aprovação dos artefatos. O que há na descrição do processo não é suficiente para todos os artefatos.
- Um único documento de ações corretivas para mais de um projeto não é adequado.
- Melhorar a descrição das ações corretivas, identificando também data prevista de início e fim.
- A alocação de colaboradores para os projetos envolve várias consultas na base de competências, além dos cronogramas dos projetos. É necessário um procedimento mais ágil, principalmente com o crescimento da empresa.
- Utilizar um método de estimativa como, por exemplo, pontos de função ou pontos de caso de uso.

3.2. Diagnóstico da Gerência de Requisitos

Da mesma forma do processo de gerência de projetos, no processo de gerência de requisitos foi detectada a seguinte melhoria:

- Gerar versões do documento de requisitos, incluindo as mudanças de requisitos aprovadas.

3.3. Diagnóstico dos Demais Processos

Com relação aos demais processos do CMMI nível 2, percebeu-se que:

- Garantia da qualidade já estava implantada.
- A organização tinha iniciado a implantação de gerência de configuração, medição, e treinamento.
- A IVIA possuía uma base de lições aprendidas e procedimentos para sugestões de melhorias implementadas.

3.4. Definição dos Processos

Após a conclusão do diagnóstico, iniciaram-se as atividades de adequação dos processos da IVIA para o CMMI, nível 2.

Assim, foram definidos os novos processos, incluídos itens na política organizacional, realizadas inclusões/atualizações nos papéis e *templates*.

Pelo fato de já termos implantado a gerência de requisitos e gerência de projetos do MPS.BR, nível G, foram realizados apenas ajustes na área de processo de

gerenciamento de requisitos do CMMI e nas seguintes áreas de processo: planejamento do projeto e monitoramento e controle do projeto.

O processo de garantia da qualidade já existente sofreu algumas melhorias, dentre elas a divisão da avaliação da qualidade em duas: avaliação da qualidade do processo e avaliação da qualidade do produto. Assim, os projetos passaram a ser avaliados se estavam seguindo o processo definido e se o produto estava sendo gerado de acordo com a solicitação do cliente e com qualidade.

O processo de medição e análise também sofreu melhorias. Foram incluídos novos indicadores que facilitaram o acompanhamento e controle dos projetos por parte dos gerentes de projetos e do diretor de software. Um exemplo de um indicador criado e que foi muito importante para a detecção de riscos e problemas e conseqüente tomada de decisão por parte da gerência é o “Desempenho de Esforço do Projeto na Fase” e o “Desempenho de Custo do Projeto na Fase”.

O processo de gerência de configuração foi o processo que teve que ser definido, pois o que se tinha ainda era muito pouco para o que o modelo exigia e o que a organização necessitava. Mas toda a definição do processo foi realizada juntamente com a consultoria.

A IVIA passou a ter seus processos monitorados e auditados, garantindo que os mesmos estão sendo melhorados e utilizados conforme descritos na metodologia/política da empresa. Os gerentes de projetos passaram a ter uma maior facilidade para planejar e acompanhar os seus projetos. Os artefatos da organização passaram a ser controlados e versionados, garantindo a organização e segurança dos mesmos.

4. Resultados Obtidos

Com a experiência adquirida na implantação do nível G do MPS.BR na IVIA, tanto a equipe de SEPG responsável pela definição dos processos, como toda a organização, tinham consciência com relação à importância de trabalhar com processos e o trabalho de implantação dos novos processos para a avaliação do CMMI contou com a colaboração de toda a área de desenvolvimento de software da organização. Notava-se claramente uma maior organização, controle e qualidade do desenvolvimento de software da empresa, liberação de produtos com uma maior qualidade e uma maior satisfação dos clientes.

4.1. Recursos Humanos Capacitados

Os novos processos foram apresentados a todos os colaboradores da empresa através de treinamentos:

- Treinamento sobre gerência de configuração (CM) com foco no CMMI, com a participação de toda a área de desenvolvimento de software da organização;
- Treinamento na gerência de requisitos, planejamento do projeto e monitoramento e controle do projeto, com a participação de toda a área de desenvolvimento de software da organização;
- Treinamento sobre gerência e garantia da qualidade, específico para a equipe de garantia da qualidade;
- Treinamento nos processos da IVIA: garantia da qualidade, gerência de configuração e medição e análise, com a participação de toda a área de desenvolvimento de software da organização;
- Treinamento sobre engenharia de requisitos, com a participação de todos os especialistas de sistemas da área de desenvolvimento de software da organização;

A equipe definida pela IVIA para participar como avaliador da avaliação oficial do CMMI participou do curso oficial de introdução ao CMMI. Esse curso foi organizado pela própria IVIA e ministrado por Viviana L. Rubinstein, avaliadora líder do CMMI.

5. Aplicabilidade dos Resultados

Como uma forma de a IVIA avaliar o uso dos processos de qualidade na melhoria do desenvolvimento de software da empresa, foi realizada uma avaliação organizacional com base no processo de medição da IVIA.

Para isso, foram considerados os dados coletados dos projetos executados ao longo da implantação dos modelos de qualidade MPS.BR e CMMI. Para compararmos a evolução da empresa com relação ao amadurecimento desses modelos dentro da organização, foram gerados dois gráficos, conforme ilustrado nas figuras 1 e 2.

Após a implantação do MPS.BR nível G e as melhorias detectadas no projeto piloto do MPS.BR, a empresa passou a ter projetos tanto acima do valor de referência, quanto abaixo dele, porém não tão distantes quanto o projeto piloto do MPS.BR (Figura 1).

Com a implantação do modelo CMMI nível 2, os projetos pilotos ficaram todos acima do valor de referência, destacando-se o projeto M, um dos projetos avaliados no CMMI nível 2, no qual foram aplicadas as lições aprendidas dos projetos pilotos, demonstrando claramente que a empresa está amadurecendo e evoluindo em seus processos.

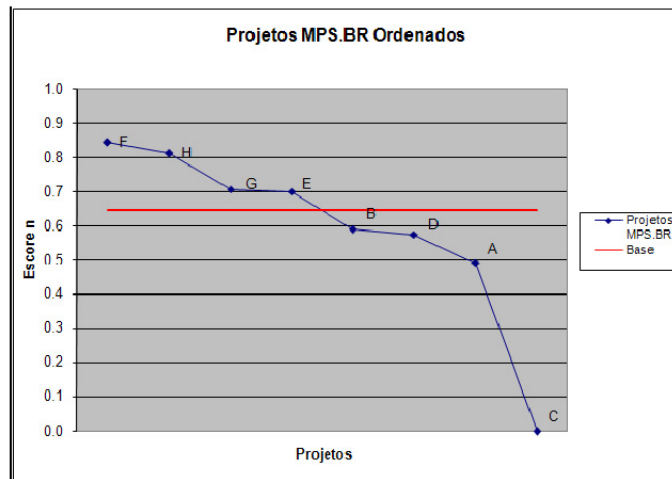


Figura 1 - Projetos do MPS.BR

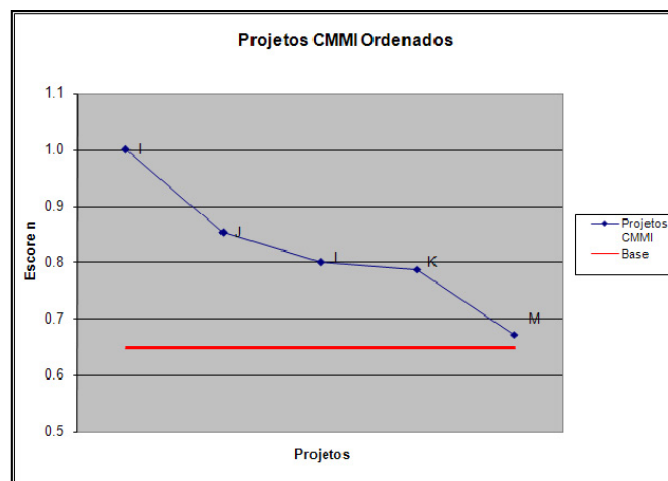


Figura 2 - Projetos do CMMI

Assim, ficou evidenciada a importância da melhoria de processos de desenvolvimento de software na organização, propiciando uma maior conscientização dos colaboradores, uma maior organização, controle e qualidade e uma maior satisfação dos clientes.

6. Características Inovadoras

Com esse processo de qualificação, após a experiência da avaliação do MPS.BR, nível G, a IVIA teve mais segurança e maturidade para alcançar o seu objetivo final, que era obter o nível 2 do CMMI. Com as melhorias implementadas para o CMMI, a IVIA passou a ter seus processos monitorados e auditados, garantindo que os mesmos estão sendo utilizados e aprimorados. Seus gerentes de projetos passaram a ter uma maior facilidade para planejar e acompanhar seus projetos.

7. Conclusão e Perspectivas Futuras

Em dezembro de 2007, a IVIA foi avaliada e obteve o nível 2 do CMMI. A equipe avaliadora divulgou o resultado da avaliação, sugerindo algumas melhorias para os processos, bem como algumas melhorias para a organização.

Como a IVIA está crescendo, é importante o uso de processos para que esse crescimento seja seguro e estável. Pelo fato do CMMI ser uma certificação reconhecida internacionalmente, a IVIA poderá participar de forma competitiva no mercado internacional e participar de forma efetiva em licitações brasileiras.

A IVIA tem como meta organizacional obter o nível E do MPS.BR e em seguida o nível 3 do CMMI e assim sucessivamente. Para a empresa, é importante alternar avaliações MPS.BR e CMMI, pois o MPS.BR possui uma divisão mais granular de área de processo que o CMMI. Desta forma, a dificuldade enfrentada pela organização é minimizada, porque são trabalhadas poucas áreas de processo em cada nível do MPS.BR, para posteriormente chegar a um nível do CMMI.

8. Referências Bibliográficas

CMMI (2006). CMMI Product Team. Capability Maturity Model Integration, version 1.1. CMMI for software engineering (CMMI-SW/IPP, v1.2) staged representation. Software Engineering Institute, 2006.

ISO 9001 (2000) - ABNT – ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS.

NBR ISO 9000:2000 – Sistemas de gestão da qualidade e garantia da qualidade – Fundamentos e Vocabulário. Rio de Janeiro: ABNT, 2000.

MPS.BR (2006), Melhoria de Processo do *Software* Brasileiro, **Guia Geral**, versão 1.1., Maio

PMBOK (2004). A guide to the project management body of knowledge. Syba: PMI Publishing Division, 2004. Disponível em: <www.pmi.org>.

OLIVEIRA, Karlson B. de. Aplicação da Estatística Multivariada para Apoiar a Avaliação Organizacional. Dissertação (Mestrado em Informática Aplicada), Universidade de Fortaleza, Fortaleza, 2008.

[6.05] Evaluate – Ferramenta de Gerenciamento de Avaliações nos Modelos MPS.BR e CMMI

Entidade: ENGSOFT - Engenharia de Software - Rua 15 de novembro, 321/503 - 98.700-000 - Ijuí - RS - Brasil

Autores: Cristiano Schwening, Fernando Scherer Fischer, Vinícius Urnau - crsch@engsoft.com.br, fernandofischer@gmail.com, vinyurnau@yahoo.com.br

Resumo. *Este projeto teve como objetivo desenvolver um sistema web com a finalidade de realizar o gerenciamento de avaliações. O objetivo do sistema é servir de ferramenta de apoio para os avaliadores, dos modelos MPS.BR e CMMI, nas suas atividades habituais de avaliação, permitindo maior agilidade na coleta de evidências, no apontamento de pontos forte e fracos detectados na avaliação e na classificação final dos processos.*

1. Introdução

Atualmente, os sistemas de informação estão abrangendo as mais diversas áreas, e sendo utilizados para as mais diversas finalidades. Com isto, surgem a cada momento, novas empresas focadas no desenvolvimento de novas soluções, e conseqüentemente, novas equipes são formadas e apoiadas em processos definidos de desenvolvimento de software.

Tanto para as novas empresas, quanto para as já estabelecidas no mercado, vislumbra-se uma busca cada vez maior pela melhoria e aprimoramento dos processos, visando-se um ganho de qualidade.

Fator este proporciona as empresas uma melhor competitividade e maior confiança de seus clientes.

Objetivando a busca da melhoria da qualidade no processo de desenvolvimento de software, o Departamento de Defesa Americano apoiou a criação do SEI (Software Engineering Institute) na Universidade de Carnegie Mellon. Este instituto, no ano de 1995, criou o SW-CMM (Capability Maturity Model for Software), que consistia em um modelo utilizado pelas organizações para identificar as melhores práticas, ajudando-as a aumentar a maturidade em seus processos. No ano de 2000, este modelo foi atualizado para CMMI (Capability Maturity Model Integration) na versão 1.1, que pode ser usado para apoiar projetos de melhoria de processo de uma divisão ou da organização inteira.

No Brasil, em novembro de 2003, foi criado pela Softex (Associação para Promoção da Excelência do Software Brasileiro), o MPS.BR (Melhoria de Processo do Software Brasileiro) que consiste de um modelo de referência (MR-MPS) e de um método de avaliação (MA-MPS), este modelo de referência oferece conformidade com as normas internacionais ISO/IEC 12207 - Processos do Ciclo de Vida do Software, e suas emendas 1 e 2, ISO/IEC 15504 - Avaliação de Processo e com o modelo CMMI.

2. Objetivos e Justificativa

No presente projeto, desenvolveu-se uma pesquisa teórica contemplando conceitos envolvendo o assunto e o desenvolvimento prático de um sistema para apoiar a avaliação da qualidade do processo de desenvolvimento de software baseado nos modelos MPS.BR e CMMI. Este sistema sofreu uma homologação inicial através de pré-avaliação do nível G do MPS.BR em uma empresa de software.

Um aspecto importante a ser observado, além da implementação de um projeto de melhoria de processos, é realizar a avaliação dos resultados obtidos com a implantação, permitindo assim estabelecer uma base sólida quanto a evolução do processo. A literatura descreve esta avaliação de processos como sendo um exame disciplinado dos processos utilizados pela organização em relação a um modelo de referência, visando determinar a capacidade dos processos ou a maturidade de uma organização. Tipicamente existem dois tipos de avaliações a interna e a externa.

As avaliações internas podem servir para a organização saber o quanto ela encontra-se preparada para uma avaliação externa que visa certificação. Durante o processo de avaliação, podem ser encontradas oportunidades de melhorias que devem ser tratadas com ações tomadas pela organização, o que contribui para o processo de melhoria contínua.

Já as avaliações externas, realizadas por empresas especializadas nesta atividade, o processo é mais detalhado e necessita o registro de informações e evidências que possam posteriormente subsidiar a definição de um plano de melhorias que aborde um perfil alvo de processos a serem implementados ou que possam auxiliar o avaliador no momento da caracterização de um nível de maturidade ou capacidade. Habitualmente esta atividade é realizada utilizando-se de planilhas eletrônicas, que são ferramentas pobres de recursos, dificultando a documentação da avaliação e armazenamento de evidências.

3. Metodologia de execução

O projeto teve duas importantes etapas de execução. Na primeira etapa foi abordada uma pesquisa do tipo teórico-experimental, onde foram realizadas pesquisas bibliográficas, na internet e através de entrevistas com profissionais conhecedores dos modelos CMMI e MPS.BR, visando um melhor entendimento dos modelos e de seus processos de avaliação. A segunda etapa foi responsável pela realização do desenvolvimento do sistema de gerenciamento de avaliações, a documentação básica e a homologação deste sistema através de simulações práticas.

Para subsidiar o desenvolvimento do sistema foram realizadas no início da segunda etapa na análise minuciosa nas informações obtidas nas entrevistas e na pesquisa. Em consequência disso foram definidos os requisitos funcionais desejáveis e que deveriam compor o sistema:

- Permitir o gerenciamento de níveis, processos, objetivos, práticas e versões dos modelos abordados.
- Permitir o gerenciamento de fases, processos e tarefas padrões dos métodos de avaliação SCAMPI e MA-MPS.
- Realizar o controle do planejamento e aprovação da avaliação, permitir a definição da equipe avaliadora e suas responsabilidades.
- Fornecer o gerenciamento da execução de avaliações das práticas e objetivos do modelo avaliado.
- Permitir o gerenciamento da caracterização de implementação das práticas e atribuição de níveis de capacidade e maturidade das mesmas.
- Fornecer um repositório dos pontos fortes e fracos detectados na avaliação e que serão repassados para a empresa avaliada.

- Gerar relatório de resultado final da avaliação, contendo dados da avaliação, caracterização dos processos e o nível atingido pela organização.
- Controlar as unidades organizacionais avaliadas e o acompanhamento de seus níveis de capacidade ou maturidade.

4. Resultados obtidos

Os resultados alcançados por este projeto abrangem desde a capacitação de recursos humanos para a área de avaliação de processos até o desenvolvimento de um sistema utilizando componentes de código livre. A seguir são apresentados os principais resultados obtidos:

- Produtos de software gerados (módulos ou programas de computador resultantes do projeto, disponibilizados para o mercado): desenvolvido e disponibilizado um sistema de gerenciamento de avaliações. Este sistema encontra-se atualmente disponível para avaliação dos usuários em <http://evaluate.engsoft.com.br>, sendo o usuário para acesso: administrador e a senha: 2007engsoft. Neste artigo, o anexo A apresenta sucintamente um tutorial com as principais funcionalidades.
- Métodos e/ou algoritmos desenvolvidos: definição de uma metodologia de avaliação. O fluxo para a utilização do sistema fornece uma metodologia para a realização de uma avaliação.
- Artigos publicados: publicação e apresentação de artigo no 9º Salão de Iniciação Científica e 6ª Jornada de Pesquisa do SAPS (Salão de Pesquisa SETREM) em Três de Maio/RS.
- Recursos humanos capacitados (especialistas, mestres, doutores, etc.): dois bacharelados em sistema de informação capacitados em melhoria e principalmente em avaliação de processos. Estes recursos humanos capacitados podem disseminar este conhecimento nas empresas de software onde trabalharem.
- Outros resultados: Primeiro lugar da área de Computação do 9º Salão de Iniciação Científica na SETREM.

5. Aplicabilidade dos resultados

Os resultados alcançados por este projeto podem ser aplicados no apoio a realização das avaliações de processos, visto que o sistema desenvolvido durante o período fornece uma excelente ferramenta de apoio direcionada aos avaliadores. Desta forma, o sistema tenta suprimir a necessidade de manuseio de planilhas eletrônicas que atualmente são muito utilizadas para direcionar uma avaliação e armazenar as evidências obtidas.

A utilização do sistema desenvolvido abrange e impacta, principalmente, os avaliadores brasileiros do MR MPS que possam visualizar esta ferramenta como um excelente suporte de apoio nas suas atividades habituais de avaliação, permitindo maior agilidade da coleta de evidências e no apontamento de pontos forte e fracos detectados na avaliação. A possibilidade de imprimir o relatório de avaliação pelo sistema agiliza o envio dos resultados finais para a Softex, podendo assim impactar também nessa entidade.

6. Características inovadoras

O sistema desenvolvido neste projeto é considerado pela equipe responsável como sendo uma ferramenta original. Durante a pesquisa teórica não foram encontrados no Brasil produtos semelhantes e com os requisitos que foram atendidos por este sistema.

No âmbito tecnológico o projeto utilizou componentes desenvolvidos pela comunidade de software livres que permitiram um rápido desenvolvimento do produto utilizando-se de

características de componentes pré-desenvolvidos e nativos na linguagem de programação utilizada.

7. Conclusão e perspectivas futuras

Acredita-se que a utilização do sistema para gerenciamento de avaliações, conforme os requisitos levantados fornecerão uma maior produtividade e suprirão as necessidades atuais no gerenciamento de avaliações, sendo viável a adoção deste por instituições avaliadoras e implementadoras e organizações que desejam realizar uma auto-avaliação.

A utilização da ferramenta Evaluate foi de grande importância para a execução mais rápida do processo de avaliação dos processos de software na empresa utilizada como piloto de homologação.

Com uso desta, a avaliação foi criada automaticamente de acordo com o modelo avaliado e o método utilizado para tal, deixando o ambiente pronto para a caracterização de práticas. A possibilidade de registro de pontos fracos, pontos fortes e da possibilidade de anexar documentos de evidências diretas e indiretas, proporcionaram a centralização dos dados e a redução do tempo nestas tarefas, em comparação com o uso de planilhas eletrônicas que atualmente são usadas para estas finalidades.

Os próximos passos deste projeto no âmbito de pesquisa buscam submetê-lo, através de artigos em eventos relacionados com Qualidade e Engenharia de Software, além de buscar recursos em organismos como a FINEP e o CNPQ para desenvolver melhorias tecnológicas na ferramenta. Com relação a utilização da ferramenta em outras avaliações, será necessário desenvolver melhorias de interface que foram propostas durante a homologação, outra necessidade será tentar homologar a ferramenta utilizando o modelo CMMI e por fim iniciar a pesquisa e o posterior desenvolvimento de uma funcionalidade que permita que partindo do processo de avaliação escolhido pelo avaliador possa obter também a atribuição do nível de maturidade para mais de um modelo de melhoria estagiado simultaneamente.

8. Referências bibliográficas

FISCHER, Fernando S.; URNAU, Vinícius; ROCKENBACH, Renato; SCHWENING, Cristiano. Desenvolvimento e Homologação de um Sistema para a Avaliação da Qualidade do Processo de Desenvolvimento de Software baseado nos Modelos MPS.BR e CMMI. Três de Maio, RS: SETREM, 2007.

SEI - Software Engineering Institute. CMMI for Development, version 1.2. Estados Unidos: Carnegie Mellon University, 2006.

SEI - Software Engineering Institute. Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.2: Method Definition Document. Estados Unidos: Carnegie Mellon University, 2006.

SEI - Software Engineering Institute. Appraisal Requirements for CMMI, Version 1.2 (ARC, V1.2). Estados Unidos: Carnegie Mellon University, 2006.

SOFTEX – Associação para Promoção da Excelência do Software Brasileiro. MPS.BR - Guia Geral, v 1.1 2006. Disponível em www.softex.br.

SOFTEX – Associação para Promoção da Excelência do Software Brasileiro. MPS.BR - Guia de Avaliação, v 1.0 2006. Disponível em www.softex.br.

ZAHRAN, S. Software Process Improvement. Estados Unidos: Addison-Wesley, 1998.

Anexo A - Principais funcionalidades da ferramenta, tomando por base uma avaliação do nível G do MR MPS.

Inicialmente a avaliação é criada no sistema através da tela de Avaliações, conforme

demonstra figura 1.

Figura 1: Tela de Avaliações.

Assim que a avaliação é criada no sistema, os processos que compõem o nível G do MPS.BR (Gerenciamento de Requisitos e Gerenciamento de Projetos), e seus atributos são automaticamente carregados. As fases, processos e atividades definidas pelo método da avaliação (MA-MPS) também são automaticamente carregados, sendo em seguida realizado o planejamento das fases e processos na tela de planejamento da avaliação, conforme demonstra a figura 2.

EVALUATE - Sistema Gerenciador de Avaliação de Processos

Logado como: cristiano - Sair

Avaliação [7 (Organização X)] - Editar

Gerar | Planejamento | Abrangências | Projetos Avaliados | Classificações | Membros da Equipe | Registros | Anexos

Avaliação: 7
Organização: Organização X
Patrocinador: Gerente da Organização X
Versão do Método: MA-MPS 1.0
Classe: Classe A
Nível Desejado: MPS.BR 1.1 - G (Parcialmente Gerenciado)
Nível Atualizado: Revisão
Representação: Estojão
Escopo: Matriz da empresa em [cidade da organização]

Percentual de Pessoas Envolvidas:
Percentual Projetos Avaliados:
Fatores Críticos:

Status: Nova

EVALUATE - Sistema Gerenciador de Avaliação de Processos

Logado como: cristiano - Sair

Avaliação [7 (Organização X)] - Editar

Gerar | Planejamento | Abrangências | Projetos Avaliados | Classificações | Membros da Equipe | Registros | Anexos

Data de Início: Segunda 2 de Julho 2007
Data de Conclusão: Quarta 4 de Julho 2007
Validado:
Fases:

Fase	Data de Início	Data de Conclusão
MA-MPS 1.0 - Fase 1 (Contratar a avaliação)	2 julho 2007	2 julho 2007
MA-MPS 1.0 - Fase 2 (Preparar para a realização da avaliação)	3 julho 2007	3 julho 2007
MA-MPS 1.0 - Fase 3 (Realizar a avaliação)	3 julho 2007	3 julho 2007
MA-MPS 1.0 - Fase 4 (Documentar os resultados da avaliação)	4 julho 2007	4 julho 2007

Fases da Avaliação: [Verificar] [Imprimir] [Excluir]

Salvar e fechar | Salvar | Cancelar

Figura 2: Planejamento da Avaliação.

Todas as atividades realizadas podem ser cadastradas no decorrer da avaliação através da tela de execução de atividades. Após são definidos e cadastrados os projetos a serem avaliados, na tela de projetos avaliados.

Assim que os projetos estejam no sistema, a caracterização dos resultados esperados em projetos é automaticamente pré-cadastrada. Através da tela de caracterização dos resultados esperados por projetos (como demonstra a figura 3), o avaliador então necessita apenas caracterizar a implementação dos resultados esperados nos projetos avaliados conforme o andamento da entrevista com integrantes das equipes. As evidências coletadas também podem ser cadastradas neste momento.

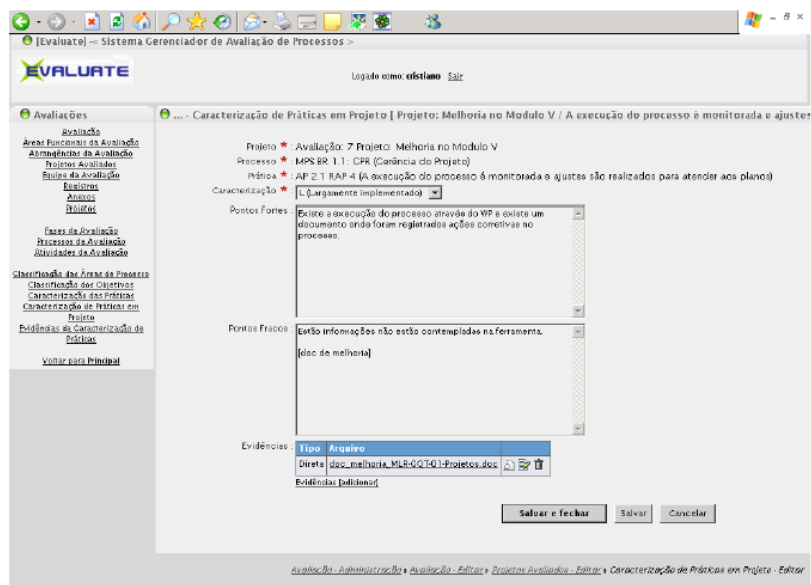
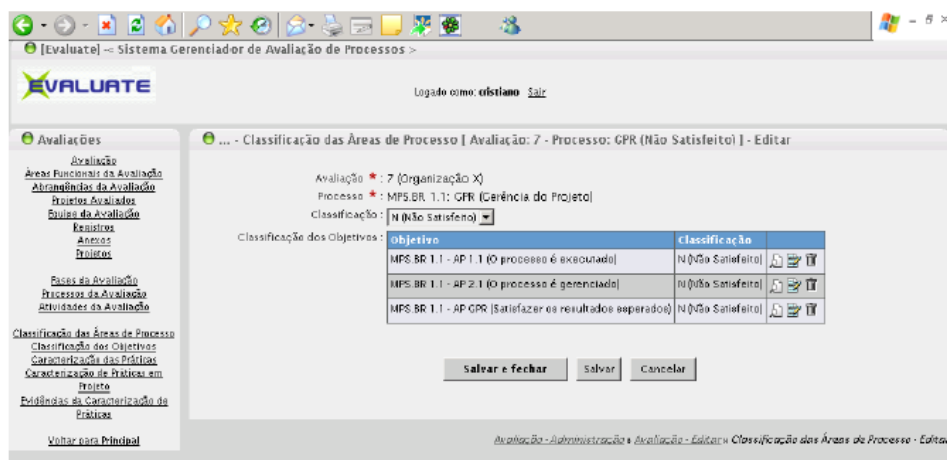


Figura 3: Caracterização dos resultados esperados por projetos.

Após os resultados esperados serem caracterizados pelo avaliador por projeto, a caracterização agregada dos resultados esperados é realizada na tela de caracterização de resultados esperados, seguindo as regras subjetivas estabelecidas pelo método. Tendo os resultados esperados caracterizados em nível organizacional, os atributos de processo devem ser classificados pelo avaliador, como satisfeitos ou não, através da tela de classificação de atributos de processo.

Para finalizar a classificação, é realizada a classificação dos processos através da tela



de classificação de processos.

Figura 4: Classificação de processos.

Ao final da avaliação é disponibilizada a opção de emissão do relatório final da avaliação.

[6.08] IACS - Identificação Automática de Componentes de Software

Entidade: 1 Laboratório de Inovação - DigitalAssets / Ci&T Invasoft - Unicamp e 2 Instituto de Computação - Universidade Estadual de Campinas (UNICAMP) - Caixa Postal 6176 - 13.084-971 - Campinas - SP - Brasil

Abstract. *Software reuse is seen as one of the main alternatives to increase productivity in the development of new applications. The reuse of legacy assets plays a vital role anticipating the ROI (Return on Investment) on SOA (Service Oriented Architecture) and reuse enterprise programs. This work presents a tool that implements an Automatic Identification of Software Components (AISC). AISC is an approach that brings to light what companies have already developed by applying reuse indicators with sophisticated mechanisms to identify artifacts that can be considered as reusable assets. Thus, they will have the potential of being reused in new applications, avoiding redevelopment of already existing features, enabling savings and increasing agility.*

Resumo. *O reuso de software é uma das principais alternativas para aumentar a produtividade no desenvolvimento de aplicações. O reuso de ativos legados tem papel fundamental na antecipação do ROI (Retorno n Investiment) dentro dos programas de SOA (Service Oriented Architecture) e reuso. Este trabalho apresenta uma ferramenta que implementa a Identificação Automática de Componentes de Software (IACS). IACS é uma abordagem que destaca o que as companhias já desenvolveram, aplicando indicadores de reuso aliados a mecanismos sofisticados para identificar artefatos que podem ser considerados ativos reutilizáveis. Com isso, esses ativos terão potencial para serem reutilizados em novas aplicações, evitando desenvolvimento duplicado de funcionalidades já existentes, aumentando a agilidade e proporcionando economia. O processo de varredura e análise de novos ativos é feito visualmente por meio de um gráfico interativo dos resultados e de um mecanismo de exportação dos ativos identificados em um Modelo de Representação de Metadados largamente utilizado.*

1. Introdução

O reuso de software em desenvolvimento de sistemas é uma estratégia que proporciona melhoria na produtividade e na qualidade dos produtos [Boehm99]. O Desenvolvimento Baseado em Componentes (DBC) é um paradigma alinhado com esta estratégia, uma vez que ela é voltada para reuso de partes previamente construídas (componentes), construção de componentes e suporte à manutenção e melhoria dos produtos mediante substituição e adaptação destes componentes.

São notáveis as pesquisas nos últimos anos em ferramentas que promovam o reuso de tais ativos de software. Considerando em termos intra-organizacionais, temos exemplos de repositórios de ativos digitais como o DigitalAssets Manager [Bacili06] e ambientes de desenvolvimento integrados a estes repositórios. Já em termos interorganizacionais, temos o padrão XML Web Services e redes Peer-to-Peer de compartilhamento de ativos como o RCCS [Oliveira05].

Em meio a essa tendência de maximizar o potencial de reuso, sistemas legados são aplicações já existentes, normalmente de missão crítica e ainda em funcionamento, representando um aspecto crucial a ser considerado. As principais características dos sistemas legados são o tamanho, em média, de milhões de linhas de código; encapsulam lógicas de negócio significativas; e provavelmente apresentam uma queda ou desvio acentuado de qualidade do projeto original diante de modificações em atividades de correção ou evolução do sistema [Zou03].

O Projeto IACS aborda tecnologias de identificação automática de componentes de software nesses sistemas já existentes, bem como mecanismos e padrões para a representação destes componentes e outros ativos digitais de forma flexível e interoperável em meta-modelos de dados. A identificação automática de componentes tem três aplicações principais: Reengenharia de sistemas legados, Convergência de aplicações para um novo paradigma de orientação a serviços (SOA) e Reuso de componentes.

O foco central do projeto consistiu em montar uma heurística de identificação através de um mecanismo capaz de reconhecer, nos parques de aplicações já desenvolvidas, grupos de artefatos que componham componentes. Visando, sobretudo, a extração destes componentes para catalogação e reutilização, gerando economia de investimento e evolução tecnológica/arquitetural.

Por meio de pesquisas desenvolvidas em laboratório, foram montados mecanismos capazes de identificar o agrupamento de artefatos relacionados, sugerindo ativos de software (componentes, serviços, etc.) candidatos à reutilização. Conforme ilustrado na figura a seguir, a ferramenta realiza diferentes etapas desde a leitura das aplicações legadas até a catalogação dos ativos reutilizáveis identificados em repositórios de componentes.

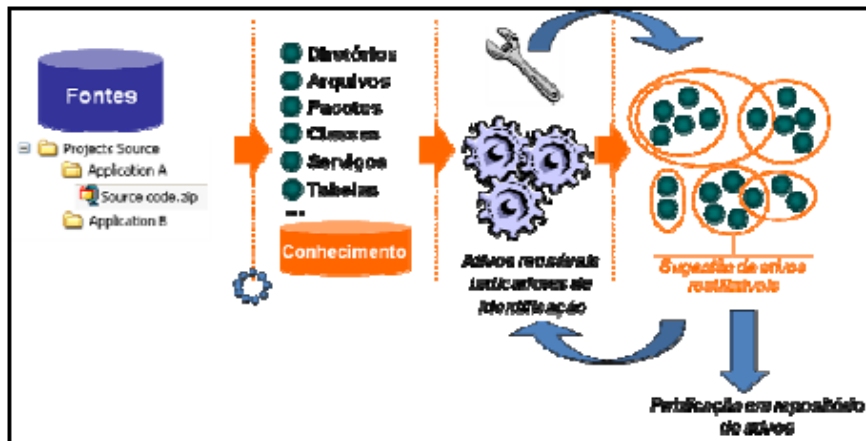


Figura 1: Mecanismo de mineração de ativos do IACS

O mecanismo de identificação de componentes de software permite realizar a reengenharia dos sistemas legados em sistemas baseados em componentes, identificando partes reutilizáveis do sistema legado e permitindo reestruturá-lo em torno destas peças.

2. Objetivo e justificativa

O foco central do projeto IACS consiste no desenvolvimento de um mecanismo capaz de reconhecer, em parques de aplicações já desenvolvidas, grupos de artefatos que componham ativos reutilizáveis de software, serviços web e componentes. Visando, sobretudo, a extração destes componentes para catalogação e reutilização.

O principal objetivo é evitar o re-trabalho no desenvolvimento de novos projeto, capitalizando trabalhos anteriores, fazendo com que as soluções já desenvolvidas sejam imediatamente implementadas em novos contextos. Desta forma, tem-se melhores produtos em um menor intervalo de tempo, com redução nos 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 bem testados [D'Souza 99] [Jacobson97].

Para que seja possível reusar algo, é necessário identificar o que pode ser reutilizado. É sob esta perspectiva que este trabalho foi desenvolvido. O principal desafio foi desenvolver uma arquitetura baseada em padrões que permitam a fácil integração com outras ferramentas como a RCCS (Rede de Compartilhamento de Componentes de Software). A RCCS é o projeto de uma rede Peer-to-Peer, baseada em padrões abertos e que pode ser utilizada por qualquer pessoa ou entidade interessada em buscar ou compartilhar componentes. A RCCS foi contemplada com o prêmio Dorgival Brandão Júnior Qualidade de Software, no ciclo 2006 durante o Encontro de Qualidade e Produtividade de Software (EPQS).

3. Metodologia de execução

A identificação de ativos reutilizáveis a partir de aplicações existentes usando o DigitalAssets Discoverer ocorre em quatro estágios, como demonstrado na Figura 1.

1. **Varredura de aplicações existentes:** Aplicações existentes são selecionadas, e o processo de varredura nas fontes especificadas seleciona os artefatos primários (arquivos-fonte, bibliotecas, etc.) que integram a aplicação. Estes artefatos serão os insumos para o estágio de criação da base de conhecimento.

2. **Criação da Base de Conhecimento:** A partir dos arquivos-fonte da aplicação encontrada na varredura, analisadores estáticos de código geram a base de conhecimento das aplicações cadastradas no estágio anterior.

3. **Configuração e execução dos indicadores:** Indicadores são as heurísticas que analisam a base de conhecimento e através de critérios apoiados em práticas de programação como por exemplo modularização; nomenclatura; padrões de projeto; arquitetura de software; coesão e acoplamento; e desenvolvimento baseado em componentes; identificam partes de código com potencial para reutilização. Em um processo iterativo, calibrações dos indicadores e execuções podem ser executadas com o objetivo de alcançar resultados otimizados.

4. **Colheita:** Os ativos sugeridos pelos indicadores como candidatos a componentes é apresentado. O analista pode navegar pelos resultados, decidir quais sugestões são relevantes, possivelmente adaptar as sugestões às suas necessidades.

4. Resultados

4.1. Pedidos de patente

O projeto IACS deu origem a um produto comercial batizado de DA Discoverer absorvido no portfólio de produtos da empresa DigitalAssets. O DigitalAssets Discoverer está em processo de encaminhamento do Pedido de Registro de Programa de Computador através do INPI (Instituto Nacional de Propriedade Intelectual) com os direitos patrimoniais atribuídos a DigitalAssets.

4.2. Produtos de software gerados

O DigitalAssets Discoverer é uma ferramenta de apoio a analistas e consultores dentro de iniciativas de modernização de aplicações, reuso e SOA. Tendo isso em mente, o fluxo de trabalho apresentado na seção *Metodologia de execução* foi implementado e está mapeado na interface gráfica do DigitalAssets Discoverer. Outras funcionalidades não relacionadas com o referido fluxo, mas importantes no aspecto qualitativo do software também estão presentes.

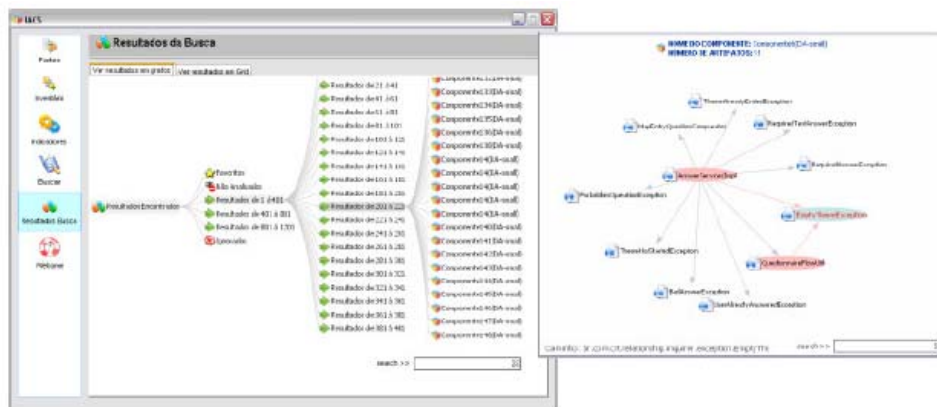


Figura 2: DigitalAssets Discoverer (interfaces de navegação nos componentes)

4.3. Outros produtos gerados (que foram disponibilizados para o mercado)

Um dos subprodutos gerados pelo projeto e eu agrega grande valor é o módulo de integração com repositório de metadados via padrão de mercado RAS. Esse módulo foi utilizado para integrar o IACS ao DigitalAssets Manager – principal ferramenta de repositório de metadados do mercado nacional.

A integração com um produto comercial deu grande visibilidade ao projeto, além de agregar valor ao negócio dos clientes do DigitalAssets Manager.

4.4. Métodos e/ou algoritmos desenvolvidos

No contexto desse projeto, *indicadores* são heurísticas capazes de identificar ativos de software com potencial para a reutilização. Portanto, os indicadores são partes fundamentais da ferramenta proposta. Muitas abordagens para identificação de componentes e outras atividades na área de engenharia reversa ou reengenharia de sistemas legados são encontrados na literatura. Foram estudadas e validadas diferentes abordagens já publicadas de identificação de grupos de artefatos que sugerem sua utilização e reutilização conjunta.

Os indicadores selecionados para serem implementados no projeto foram:

- **Coesão e acoplamento:** Definimos a metodologia de Coesão e Acoplamento como sendo a análise estrutural (análise de artefatos e interdependências) a fim de encontrar módulos de software que compõem o sistema e caracterizados como grupos significativos (alta coesão) e independentes (baixo acoplamento). Este tipo especial de *clustering* foi explorado em [Mitchell06] com o intuito de encontrar automaticamente abstrações, como subsistemas, a fim de auxiliar o especialista na tarefa de entender a estruturação do código.
- **Análise de dominância:** Análise de Dominância é um percurso em grafos dirigidos de forma a capturar a relação de dominação entre nós, por exemplo, se um nó N é dito dominar outro nó M em um grafo dirigido G, todo caminho da raiz de G até M contém N. A árvore de dominância é uma possível representação dos relacionamentos entre um nó e seu dominador imediato. A aplicação da análise de dominância em estruturas de sistemas e código-fonte existentes já foi tema de metodologias e implementações. Em [Buschsbaum01] podemos encontrar uma proposta de ferramenta que usa esta análise tanto para avaliar qualitativamente a complexidade do sistema quanto para avaliar o impacto de evoluções nos demais subsistemas envolvidos.
- **Matriz de Estrutura das Dependências (Dependency Structure Matrix – DSM):** A Matriz de Dependências é usada para modelagem e análise de sistemas que envolvam entidades como tarefas, pessoas, etc., e relacioná-las informando o fluxo

de informações entre as entidades [Yassine04]. Além de aplicações em várias áreas da engenharia de processos, DSM é útil em análises de sistemas complexos e para representação das interdependências entre os elementos de software. Devido ao grande volume em potencial de informações extraídas destes sistemas, ferramentas são empregadas para que o analista possa especificar um projeto, modelá-lo em DSM e executar automaticamente algoritmos que determinam agrupamentos ou seqüências dos elementos do DSM que melhor caracterizam o fluxo de informações.

4.5. Artigos publicados

O projeto IACS foi mencionado em diversos eventos como SBCARS 2007 (Simpósio Brasileiro de Componentes e Reuso de Software), SBQS 2007 (Simp. Brasileiro de Qualidade de Software), Wire 2007 (Workshop de Introdução de reuso em empresas), 2º Minds-ON (2º Fórum de Inovação DigitalAssets), dentre outros. Além desses, o trabalho originou publicações internacionais bastante relevantes (IEEE e ACM):

- A ferramenta foi apresentada e demonstrada em um dos principais eventos de reuso de software do mundo, o IRI (*The 2007 IEEE International Conference on Information Reuse and Integration – IEEE - Las Vegas/USA*). Sob o título: “Automatic Identification of Reusable Software Development Assets: Methodology and Tool”, [Goncalves07].
- A ferramenta também foi apresentada no evento internacional da ACM, o OOPSLA (***22st Annual ACM Conference on Object- Oriented Programming, Systems, Languages, and Applications, Montréal, Canada***), sob o título: “DigitalAssets Discoverer: Automatic Identification of Reusable Software Components”, [Oliveira07a]. Este evento é considerado um dos mais importantes do mundo na área de orientação de objetos e aplicações. Foi possível realizar exposições práticas do mecanismo do IACS realizando identificação de componentes em sistemas.
- Além de citações e publicações relacionadas, como em [Oliveira07b], “DA Manager®, gerência e avaliação da reutilização de ativos digitais”. perfil prático do projeto pôde ser afirmado através das publicações, que foram, em sua maioria, em salões de demonstração. Nestes salões foi possível instalar e permitir a interação dos participantes do evento com a ferramenta.

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

O projeto contou com pesquisadores do Instituto de Computação da Unicamp, consultores e gerentes de projeto da DigitalAssets e da Ci&T e com alunos de graduação e mestrado. Um dos principais resultados do projeto foi a capacitação de pessoal, gerando especialistas em nas tecnologias relacionadas ao projeto, sobretudo em SOA (*Service Oriented Architecture*), arquiteturas Peer-to-Peer, além de um mestre em ciência da computação, que teve a RCCS como tema de projeto de mestrado.

4.7. Eventuais parcerias ou programas de transferência de tecnologia efetuados

O Laboratório tem parceria com diversas instituições de pesquisa, e vêm realizando projetos em conjunto com C.E.S.A.R., UFPE, UNICAMP, UFPB, IBTA, CenPRA e UFV. O projeto da IACS foi desenvolvido juntamente com a Unicamp, e algumas ferramentas (como o mecanismo de visualização) foram desenvolvidas em laboratórios da Ci&T Labs.

5. Aplicabilidade dos resultados

Atualmente o setor de tecnologia sofre uma pressão que força a divisão de investimentos e o alinhamento de iniciativas de TI com as estratégias de negócio. Para lidar com esses desafios as organizações de tecnologia estão buscando uma variedade de iniciativas de desenvolvimento de software para atingir padronização, componetização, reutilizações

de ativos de software, estratégias de arquitetura orientada a serviços (SOA) e redução de custos. Esse é um mercado de novos paradigmas no desenvolvimento de software e abordagens de Arquiteturas Orientadas a Serviços (SOA) e Reutilização de Software serão formas primordiais para se atingir elevados patamares de produtividade e agilidade frente a mudanças.

Michael Blechar, analista do Gartner, declarou que “as empresas que tiram vantagem da reutilização de software conseguem aumentar produtividade, qualidade e entrega ao mercado em um fator de 5 para 1”. Mas isso só é possível quando times de projeto conseguem localizar e reutilizar facilmente os ativos digitais já previamente desenvolvidos e testados.

O conjunto de aplicativos legado no mercado geral de TI é extremamente significativo e de suma importância para qualquer projeto que envolva um conjunto maior de ações. O projeto IACS, aplicado a esse contexto, permite a antecipação do retorno de investimento e trazendo luz aos investimentos já realizados.

Ainda no escopo do projeto IACS, também foi implementado um mecanismo de empacotamento de ativos em modelo RAS (Export-RAS). Os principais gerenciadores de ativos internacionais utilizam este padrão para intercâmbio de componentes, e essa funcionalidade alavanca as ferramentas nacionais a um padrão internacional de representatividade de ativos, que muitas vezes é requisito obrigatório para escolha de ferramentas.

Esse mecanismo de empacotamento permitiu integrar o IACS ao DigitalAssets Manager, a 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]. Essa integração promove uma maior visibilidade do acervo já existente envolvendo e viabilizando a cooperação entre os diversos papéis envolvidos no processo de reuso, permitindo a definição dos processos de reutilização de código assim como assim como definição das políticas de acompanhamento e seus indicadores que irão, por fim, culminar em serviços com qualidade superior e agilidade suficiente para que empresas possam acompanhar as exigências do mercado atual.

6. Características inovadoras

Os principais tópicos de pesquisa relacionados são:

- Metodologias de Representação de Ativos Digitais e Componentes de Software
- Mecanismos de identificação e classificação de componentes de software
- Algoritmos de identificação automática de componentes.
- Ferramentas escaláveis para exibição gráfica da dependência entre artefatos.

Relacionado às características inovadoras, podemos citar:

- Implementação de um mecanismo totalmente inovador para mineração de componentes de software em parque de aplicações desenvolvida de maneira não componentizada.
- Criação de um mecanismo de análise arquitetural. O IACS possui um mecanismo para captação da complexidade arquitetural e visualização gráfica da árvore de relacionamento entre os artefatos do projeto, possibilitando uma avaliação da qualidade arquitetural. Em momentos de evolução tecnológica, isso torna-se de grande valor, inclusive em empresas usuárias do IACS.
- Integração de diversos algoritmos identificadores de possibilidade de reuso.

Alguns algoritmos já existiam, mas de forma isolada em ferramentas de engenharia reversa e análise ciclômica. Estes algoritmos foram combinados para uma análise moderna para identificação de conjuntos de artefatos e componentes potencialmente reusáveis.

- Geração de uma ferramenta prática e com fluxo definido para identificação de componentes. Mesmo nos principais concorrentes internacionais, não existe ferramenta com características semelhantes (isso está, inclusive, disparando um processo de registro de patente - propriedade intelectual - da solução no Brasil e nos EUA). Vale destacar que a DigitalAssets foi selecionada como uma das 25 empresas que mais inovam no Brasil, resultado da pesquisa “O Brasil que inova”, realizado pelo Monitor Group para a revista Exame (<http://pesquisainovacao-exame.monitor.com/resultados.html>). Boa parte das características inovadoras citadas da empresa estão atribuídas ao IACS.

7. Conclusão e Perspectivas futuras

O Projeto IACS aborda tecnologias de identificação automática de componentes de software em sistemas já existentes, bem como mecanismos e padrões para a representação destes componentes e outros ativos digitais de forma flexível e interoperável em meta-modelos de dados.

Após o processo de registro de patente a ferramenta tende a ser total diferencial competitivo internacional da empresa. Nacionalmente, as POVs (Provas de Valor) da ferramenta têm tido boa receptividade e está sendo trabalhada uma estratégia de trabalho e serviço para atuação dentro de organizações maiores, visando agregar maior valor à solução e às empresas com grande volume de sistemas legados

Referências

[Boehm99] Boehm, B., *Managing Software Productivity and Reuse*, Computer, vol. 32, issue 9, Sept 1999, pp. 111-113.

[Bacili06] Bacili, K., Oliveira, M., *DigitalAssets Manager: sharing and managing software development assets*, OOPSLA'06 Demo Session, ACM, NY, 2006, pp. 700-701.

[Buschsbaum01] Buschsbaum, A., Chen, Y.F., Huang, H., Koutsofios, E., Mocenigo, J., Rogers, A. *Visualizing and Analyzing Software Infrastructures*, IEEE Software, Sept./Oct. 2001.

[D'Souza99] D. F. D'Souza and A. C. Wills. *Objects, Components, and Frameworks With UML: The Catalysis Approach*. Addison Wesley. USA., 1999.

[Goncalves07] Gonçalves, E. M.; Oliveira, M. S.; Bacili, K. R.; *DigitalAssets Discoverer: Automatic Identification of Reusable Software Components*. 22st Annual ACM Conference on Object- Oriented Programming, Systems, Languages, and Applications OOPSLA- October 21-25, 2007 – Montréal, Canada. Demonstration Tools.

[Jacobson97] L. Jacobson, M. Griss, and P. Jonsson. *Software Reuse: Architecture, Process and Organization for Business Success*. Addison Wesley. USA., 1997.

[Oliveira07a] Oliveira, M. S.; Gonçalves, E. M. ; Bacili, K. R.; *Automatic Identification of Reusable Software Development Assets: Methodology and Tool*. The 2007 IEEE International Conference on Information Reuse and Integration IRI2007 - August 13 - 15, 2007 - Las Vegas, USA.

[Oliveira07b] Oliveira, M. S.; Bacili, K. R.; Vahl JR., J. C.; *DA Manager®, gerência e avaliação da reutilização de ativos digitais*. SBCARS 2007 - Brazilian Symposium on Software Components, Architectures and Reuse, August 29 – 31, 2007. Campinas, SP – Brazil.

[Mitchell06] Mitchell, B.S., Mancoridis, S., *On the Automatic Modularization of Software Systems Using the Bunch Tool*, IEEE Trans. on Soft. Eng., vol. 32, no. 3, March 2006.

[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.

[Yassine04] Yassine, A. *An Introduction to Modeling and Analyzing Complex Product Development Processes Using the Design Structure Matrix (DSM) Method*, Quaderni di Management, no. 9, 2004, *english version*.

[Zou03] Zou, Y., *Techniques and Methodologies for the Migration of Legacy Systems to Network Centric Environments*, Ph.D. Thesis, Department of Electrical & Computer Engineering, University of Waterloo, Ontario, Canada, Sept 2003.

[6.12] Um Modelo Sistêmico para Atividade de Avaliação e Testes de Software

Entidade: ¹ Centro de Pesquisa Renato Archer - CenPRA - Rodovia Dom Pedro I, km 143,6 Campinas SP - Brazil e ²Instituto de Computação da Universidade Estadual de Campinas

Autores: Ana C. Guerra¹, Eduardo Vasconcelos Silva² - Ana.guerra@cenpra.gov.br, vasconcelos2001@yahoo.com

Resumo. *Uma proposta para sistematizar o processo de criação de casos de teste encontra-se no uso de uma rede Bayesiana que mapeia a arquitetura de testes a ser implementada, aliada a uma ponderação estatística de riscos a cenários. Como produto final desta proposta sistêmica, além do ferramental gráfico que possibilita a descrição dos casos de testes segundo uma seqüência lógica e simulação de cenários, têm-se uma matriz que reúne todos os casos de testes obtidos da rede e demais oriundos da análise dos requisitos, segundo o enfoque do critério de adequação.*

1. Introdução

O desenvolvimento de produtos de software com boa qualidade ainda é complexo e caro. No momento que a sociedade depende cada vez mais de software, os problemas históricos associados com seu desenvolvimento ainda não foram adequadamente solucionados.

A avaliação é uma atividade no processo de desenvolvimento do software que, praticado com procedimentos bem definidos, tem a característica de aumentar a qualidade dos produtos desenvolvidos. Embora a tarefa de avaliar programas de computador seja tão antiga quanto o primeiro programa produzido, casos de erros nos sistemas em operação ainda são comuns. Ao longo destes anos, surgiram várias técnicas que ajudaram a melhorar o nível de qualidade dos sistemas desenvolvidos. Entretanto, tal melhoria não conseguiu acompanhar o aumento da complexidade dos sistemas de software.

Na medida em que o emprego de sistemas de software cresceu ao ponto em que boa parte de nossa vida depende cada vez mais de software e computadores, passa a ser de vital importância a existência de software confiável – software que fornece resultados corretos quando alimentado com dados válidos e que identifica corretamente dados inválidos. Principalmente quando se trata de aplicações onde um simples defeito pode causar um grande prejuízo ou simplesmente uma catástrofe, tais como software de aplicações espaciais, software de controle de processos na área médica, controle de processos em usinas nucleares.

Uma explicação para este fato de que software tem erro é que avaliar e testar software não é uma atividade trivial[Som92]. A atividade de avaliação e teste exige conhecimentos, habilidades, e infra-estrutura específica. Um bom desenvolvedor ou projetista de software sem esta base dificilmente realizaria uma boa tarefa de avaliação. Outra explicação é que as empresas produtoras de software normalmente se encontram bastante ocupadas nas tarefas rotineiras (principalmente corrigindo defeitos!) para dedicar tempo e esforço necessários para identificar os procedimentos, técnicas e ferramentas de software para essas atividades, tão importantes.

2. Objetivos e Justificativa

Esse artigo tem como objetivo buscar mecanismos que possibilitem um tratamento sistêmico da atividade de avaliação e teste de software, trazendo uma maior racionalidade na criação e execução dos casos de testes, além da melhoria da qualidade do produto. Com este enfoque é ressaltado o modelamento do problema sob uma nova óptica, buscando levantar os cenários afetados, além dos limites do documento de requisitos. Isso assegura ao desenvolvedor de testes, segurança e domínio da corrente implementação, tendo como resultado imediato, possíveis cenários não detalhados no documento de requisitos e sua correta atualização.

O processo de testes é encarado dentro de um contexto não determinístico, como sendo um gerenciamento de risco ' logo um cuidado estatístico especial será utilizado para o modelamento do problema. Outro ponto importante, dentro do modelamento proposto, é o uso extensivo do engenheiro de testes, cuja experiência é incorporada ao modelo através da ponderação adequada dos fatores que influenciam o resultado final. O trabalho tem de maneira implícita a melhoria da qualidade do produto de software colaborando para a criação de produtos mais robustos e confiáveis, através da melhoria do processo, o que indubitavelmente representa vantagens financeiras interessantes.

Será sugerida uma ferramenta que possibilita a simulação de forma computacional. Também é abordado o conceito de critério de adequação, outro elemento básico para a formulação do trabalho, que reúne elementos como caso de teste, especificação funcional, e software dentro de um equacionamento baseado em critérios. Assim será proposto a unificação da rede Bayesiana que modela o domínio de entrada com o critério de adequação, sugerindo uma sistematização no processo de criação de testes. O produto final desta sistematização é uma planilha que mapeia todos os casos de testes, visando melhorar a cobertura e eliminando duplicações. Foi realizada uma análise quantitativa de quatro projetos de testes em andamento, focando no esforço de análise e cobertura e mostra a implementação prática de um design de testes.

3. Conceitos em Testes de Software e Metodologia

Teste é uma forma de verificação dinâmica que consiste em executar o programa com um conjunto de dados de entrada e determinar se ele se comporta conforme o esperado, isto é, de acordo com sua especificação. Em geral, é impossível testar um programa exaustivamente; o importante é selecionar um conjunto finito de casos de testes que permitam testá-lo adequadamente [Mar00]. Pode-se afirmar que um teste bem-sucedido é aquele que descobre um erro ainda não descoberto, logo um bom caso de testes é aquele que tem alta probabilidade de encontrar um novo erro [Mye79]. Conceitos de erro, de falha e de defeito, são utilizados segundo o IEEE Std. Glossary of Software Engineering Terminology, padrão 610.12/1990, definidos como: erro: engano cometido por um desenvolvedor (analista, projetista, programador); falha: manifestação do erro (uma especificação ou código incorretos); defeito: evento notável ao usuário, ativação da falha.

3.1 Estudo Redes Bayesiana e Critérios de Adequação

Critério de adequação em síntese define um relacionamento entre especificação, casos de teste e programa e redes Bayesiana que proverá um ferramental para o modelamento

do teste de software, possibilitando inclusive simulações estatísticas. Um dos conceitos extremamente valiosos para a elaboração teórica do presente trabalho. Sua contribuição concentra-se na cobertura de requisitos [Pre02].

Segundo [Zhu96], critérios de adequação são regras que definem se um trecho de software foi adequadamente testado. Um vasto número de critérios de adequação tem sido propostos e investigados, podendo ser citado os seguintes: Critério baseado em fluxo de controle; Critério baseado em fluxo de dado; Critério baseado em texto de programa; Critério baseado em falhas.

4. Resultados Obtidos

A abordagem proposta visa buscar um novo paradigma para o design de testes de software incorporando elementos gráficos e uma ponderação de riscos envolvidos. Nesta proposta faz-se necessário um estudo preliminar profundo sobre os novos requisitos a serem implementados e realizar um levantamento de quais casos de usos são afetados. Uma documentação de apoio como diagramas UML de casos de uso ou mesmo um diagrama de sequência, serão materiais de inestimável valia para o correto levantamento dos cenários.

4.1 Produto Obtido no projeto

Através do levantamento consistente dos cenários, e suas inter-relações, tem-se os fundamentos necessários para o modelamento Bayesiana. Nesta etapa, a preocupação está voltada para o modelamento do problema de software, buscando levantar todas as transações, ações de software e partições de entrada relacionadas. Deste conjunto de elementos constitui a arquitetura de testes, um documento estritamente direto quanto ao levantamento dos casos de testes necessários, possibilitando inclusive, simulação de cenários e conseqüentes estimativa dos testes mais indicados a identificar falha. A Metodologia foi gerada e os resultados testados e analisados [Vasconcelos06].

4.2 Casos de Testes e Rede Bayesiana

Quando o modelamento de uma transação finaliza, obtém-se a arquitetura de testes e os casos de testes a serem aplicados, de forma muito direta e objetiva. A rede deve ser simulada partindo dos nós de mais alta hierarquia e para cada partição de entrada todas as partições de entrada do nó sucessor devem ser verificadas. Um novo caso de teste deve ser criado quando a probabilidade for diferente de zero e a interação corrente fizer sentido lógico. O grau de detalhamento do caso de teste está fora do escopo desse trabalho, mas vale aqui dizer que essa tarefa sofre influência do grau de conhecimento da pessoa encarregada de executar a tarefa de teste, ou se a execução será manual ou automática.

4.3 Cenários não previsíveis

A construção da Rede Bayesiana constitui um ferramental muito eficiente no entendimento adequado do problema e busca de situações que deveras não estão contempladas nos requisitos. Nestes casos, o comportamento esperado não é documentado, o que acaba por exigir uma revisão dos requisitos. Desta forma a documentação do projeto, por exigência da técnica, torna-se mais robusta, completa e adequada, reduzindo assim, os problemas com manutenção no futuro.

4.4 Outros resultados

Desse projeto foi gerada dissertação de mestrado e serão apresentados artigos em congressos e revistas oportunamente. O projeto foi implantado na empresa com sucesso e se encontra em operação, além da capacitação de pessoas no assunto.

5. Aplicabilidade dos resultados

De forma a tornar elucidativo o uso dessa nova técnica foram levantadas métricas relativas ao esforço de análise, cobertura de testes e razão entre o esforço as técnicas bem como a razão de cobertura. Esses são dados reais obtidos na criação de testes de software na área de telecomunicação. Uma simulação da técnica proposta na criação de testes sobre um sistema fictício de cadastro de cartões de crédito foi realizada para validar os resultados.

A seguir é exemplificado o uso da técnica na criação de casos de testes para um sistema de banco de dados, onde foi criada uma rede e populado parcialmente a matriz de adequação. O sistema fictício de validação de cartões de crédito em questão possui os seguintes requisitos:

Identificação	Descrição
Req_1	O sistema realizará a validação de cartões de crédito das operadoras A, B e C somente, recusando as demais operadoras;
Req_2	A identificação da operadora será baseada no número do cartão, a saber: Operadora A: número do cartão inicia-se com zero; Operadora B: número do cartão não se inicia com zero e possui no total dez dígitos; Operadora C: número do cartão não se inicia com zero e possui no total doze dígitos. O sistema devera suportar até 100 validações simultâneas;
Req_3	Para as operadoras B e C deve-se validar o código de segurança do cartão (baseado num algoritmo que envolve o próprio número do cartão);
Req_4	Deve-se validar também se o cartão possui a data de validade não vencida somente para os cartões das operadoras B e C.

Tabela 1 – Requisitos Sistema de Cartão

Baseado nestes requisitos, uma proposta para o modelamento do problema numa rede Bayesiana. Neste experimento, toda documentação de entrada foi um documento extremamente simples contendo os quatro requisitos. Com a aplicação da técnica sistêmica proposta, foram criados treze casos de teste, resultando num valor médio de quatro testes por requisito. Os testes baseados nos critérios positivo e negativo são os mais imediatos e obtidos da rede Bayesiana pela simples varredura seqüencial dos nós (e suas partições de entrada). Nas simulações de cenários, partições com 0% de probabilidade são casos de testes que se enquadram no critério negativo. Com relação aos critérios Limite, Volume e Interação, a matriz de adequação exige o foco em cada requisito segundo prismas diferentes, buscando identificar casos de teste que sejam aplicáveis.

6. Características Inovadoras

É indiscutível o caráter inovador e os benefícios oriundo da nova proposta para melhoria da qualidade de software por meio da avaliação. Estudos mostram também que o custo despendido na manutenção em campo pode chegar a 90% do total investido no projeto [Pig97].

Outro ponto de grande importância é a necessidade de um método sistêmico que agregue ao processo de desenvolvimento de avaliação e testes, mas especificamente aqueles baseados exclusivamente em requisitos funcionais, onde ferramentas estatísticas de simulação de cenários bem como critérios de adequação [Zhu96] irão suprir uma deficiência efetiva.

7. Conclusões e Perspectivas Futuras

Foram abordadas aqui a rede Bayesiana e o Critério de Adequação, fundamentos para a proposta sistêmica apresentada. A busca de uma sistematização do processo de criação de casos de teste teve em seus primórdios dois limitantes básicos: deveria agregar valor tecnicamente proporcionando um ferramental leve para o modelamento da arquitetura de testes e não poderia embutir uma sobrecarga relevante ao esforço de criação. Os dados experimentais obtidos sinalizam que o modelamento gráfico da arquitetura de testes acaba por vislumbrar cenários não cobertos no documento de requisito e que devem ser incluídos na documentação, de forma a evitar comportamento dúbio.

Essa contribuição torna-se ainda mais acentuada quando o desenvolvimento de testes inicia-se paralelamente ao design da arquitetura de software. Assim, a arquitetura do sistema acaba por sofrer influência direta da atualização dos requisitos, resultando numa implementação mais robusta.

O uso sistemático de uma metodologia para criação de casos de teste é também uma maneira de eliminar a subjetividade do processo, e permitir uma melhor controle na fase de criação, um ganho gerencial interessante no tocante a estimativas, com melhores previsões, quanto a disponibilidade dos casos de testes.

Foi feita uma análise quantitativa da técnica com dados reais. Os dados e gráficos comparativos sinalizam uma tendência positiva referente a uma melhor cobertura (quatro vezes e meio) com menor esforço de análise (um quarto), aliado a um sólido conhecimento da arquitetura de teste. Esses dados mostram-se muito atraentes frente ao desafio atual, da busca da qualidade ao menor custo, com certeza um diferencial competitivo.

Neste trabalho não foi abordado técnicas e vantagens do uso de testes exploratórios. Uma proposta para um trabalho futuro seria a união da corrente proposta sistêmica com o uso extensivo de testes exploratórios. O uso de testes exploratórios tornaria menor o esforço de escrita de testes que compartilham casos de usos comuns, com pequenas alterações do estado inicial, bem como testes de alta complexidade e não usuais. No estágio do design da matriz de adequação podem-se detalhar os testes que seriam os prováveis candidatos para exploratórios.

Referências

- [Mye79] Myers, G.J.. The Art of Software Testing. John-Wiley & Sons, 1979.
- [Som92] Sommerville, Ian. Software Engineering – 4th. Edition. Addison-Wesley Inc., 2000.
- [Zhu96] Zhu,Hong. A formal Analysis of the Subsume
- [Pig97] Pigoski, Thomas M; Practical Software Maintenance: Best Practices for Managing Your Software Investment. Wiley Computer Publishing, 1997.
- [Mar00] Martins, Eliane.; Manutenção e Ferramentas CASE. IC-UNICAMP, 2000.
- [Vasconcelos06] Um Modelo Sistêmico para Atividade de Avaliação e Teste de Software. Autor: Eduardo de Vasconcelos Silva, Orientador: Ana Cervigni Guerra. Co-Orientador: Rogério Drummond Burnier P de M Filho. Dissertação de mestrado - DEPARTAMENTO DE SISTEMAS DE COMPUTACAO - Unicamp 2006.
- [Woo02] Wooff, David A. Goldstein, Michael. Coolean,Frank P. A. Bayesian Graphical Models for Software Testing. IEEE Transactions on Software Engineering Vol 28, N° 5, May 2002.
- [Pre02] Pressman, Roger S.; Engenharia de Software , 5. Edição. McGraw-Hill,Inc., 2002.

[6.13] Uma ferramenta de suporte ao MA-MPS

Entidade: Faculdade de Informática - Pontifícia Universidade Católica do Rio Grande do Sul - Av. Ipiranga, 6.681 - Prédio 32, CEP: 90619-900 - Porto Alegre - RS

Autores: Marcelo Hideki Yamaguti¹, Dimitrius Fraga², Daniel Alves Gonzalez³ - 1yamaguti@pucrs.br, 2dimitriusfraga@hotmail.com, 3daniel-gonzalez@uol.com.br

Abstract. *In this work is presented a tool for supporting an Appraiser Organization in the execution of MA-MPS (the assessment method of MPS.BR) during the on-site assessment in an organizational unit.*

Resumo. *Este trabalho apresenta uma ferramenta de apoio a uma Instituição Avaliadora na execução do Método de Avaliação do MPS.BR (MA-MPS), durante a avaliação on-site em uma unidade organizacional a ser avaliada.*

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.

Mas mesmo com essa preocupação, não havia como se ter certeza de que a empresa encarregada do desenvolvimento do software cumpriria os requisitos necessários para que o software adquirisse uma qualidade satisfatória.

Sendo assim, começaram a surgir no mercado, modelos de qualidade de software como CMMI (*Capability Maturity Model Integration*), MPS.BR (Melhoria do Processo de Software Brasileiro). Esses modelos visam melhorar o processo de desenvolvimento de software da organização fabricantes.

A maioria dos modelos de avaliação de qualidade de software classifica as empresas de desenvolvimento de software com escalas de maturidade em relação aos processos de desenvolvimento.

Um dos modelos de qualidade é o MPS.BR, que tem como foco avaliar os processos de desenvolvimento de software das micro, pequenas e médias empresas.

Hoje o sistema de avaliação da IA-MPS (Instituição Avaliadora) não é realizado de forma *on-line*. O avaliador vai até a organização que está tentando atingir um nível de maturidade do MPS.BR, analisa as evidências e pontua todos os processos relevantes de acordo com o método de avaliação. As informações coletadas são preenchidas em planilhas e posteriormente enviadas via e-mail.

O objetivo desse Trabalho de Conclusão é otimizar esse serviço, criando uma ferramenta que auxilie a IA-MPS.BR no momento em que essa está avaliando, gerando uma avaliação mais rápida e segura.

2. Objetivos e Justificativa

O objetivo principal do projeto é a criação de uma ferramenta que auxilie uma Instituição Avaliadora na execução do Método de Avaliação do MPS-BR (MA-MPS), durante a avaliação *on-site* na unidade organizacional a ser avaliada.

Esta ferramenta deve permitir a sua execução pela Web (*on-line* e remota) ou localmente de forma *off-line*.

O MA-MPS define um processo para a avaliação de uma unidade organizacional dentro do MPS.BR, mas não há uma definição rígida quanto ao uso de ferramentas.

A ferramenta proposta visa otimizar o processo de avaliação, tendo em vista que a mesma irá substituir o método atual que utiliza planilhas preenchidas manualmente e que

posteriormente são enviadas à Instituição Avaliadora via e-mail, permitindo que a equipe de avaliação possa executar suas atividades de forma eficiente e correta.

3. Metodologia de Execução

O projeto foi desenvolvido em seis etapas:

- a) **Capacitação no MA.MPS** – estudo e discussão do método de avaliação proposto no MPS.BR
- b) **Levantamento de requisitos** – identificação das principais características e funcionalidades esperadas para uma ferramenta de suporte ao MA.MPS
- c) **Descrição dos requisitos** – utilizando-se UML foram descritas as funcionalidades através de casos de uso e suas respectivas descrições detalhadas.
- d) **Projeto da solução** – foram definidas as soluções para a implementação da ferramenta em termos de arquitetura, tecnologias de informação (linguagem de programação, banco de dados, servidor Web, servidor de aplicação, etc.)
- e) **Implementação** – realizou-se a geração do código da ferramenta, a instanciação do banco de dados e a configuração dos servidores.
- f) **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).

O resultado obtido foi uma ferramenta que funciona de forma *on-line*, através da Internet. Possibilitando que representantes de empresas consigam inserir documentos referentes aos seus projetos e salvá-los, através de arquivos enviados via Internet, na base de dados do servidor da IA-MPS. Esses dados salvos anteriormente serão acessados via *web* pelos avaliadores no momento em que os mesmos estiverem avaliando os projetos.

A ferramenta foi desenvolvida durante um trabalho de conclusão de curso de graduação em Ciência da Computação e pode ser obtida na Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul.

5. Aplicabilidade dos Resultados

A disponibilização de uma ferramenta de suporte ao processo MA-MPS, visa a eficácia e eficiência na execução das atividades da equipe de avaliação de empresas segundo o modelo proposto no MPS.BR.

Neste contexto, o projeto visa atingir organizações que pretendam atuar como Instituições Avaliadoras dentro do projeto MPS.BR, tendo desta forma, um alcance nacional.

Esta automatização de atividades visa reduzir o erro e melhorar a eficiência na execução das atividades da equipe de avaliação.

As limitações atuais da ferramenta é que a mesma provê suporte parcial a uma avaliação inicial e não implementou o modo *off-line* que permitiria o sistema funcionar em caso de problemas de acesso ao sistema Web.

6. Características Inovadoras

Já existem ferramentas e artefatos propostos para suporte ao MA-MPS, entretanto, o projeto se propôs a gerar um produto integrado que permita a sua utilização baseada na Web (de forma *on-line* e remota) e também de forma *off-line* e local.

A forma off-line e local não pode ser concluída.

7. Conclusão e Perspectivas Futuras

O sistema permite que o Representante da Empresa que está em processo a avaliação, consiga se logar ao sistema, incluir os indicadores referentes aos resultados de cada processo de cada projeto, bem como incluir os artefatos de cada indicador.

Os Avaliadores Adjuntos e Avaliadores Líder possuem login e senha diferenciados ao qual permite aos mesmos navegar em todas as páginas do sistema. Os avaliadores, após se logarem escolherão a empresa a ser avaliada na página de abrangência. Na página de avaliação, os avaliadores irão ter acesso a todos os indicadores e artefatos que foram incluídos anteriormente pelo representante da empresa, podendo assim avaliar os documentos, os indicadores, resultados e processos.

Também ao final dessa página, será informado o grau que a empresa alcançou.

Está também disponível, aos avaliadores no momento da avaliação, a inclusão da descrição de possíveis entrevistas com representantes das empresas.

O sistema funciona de forma *on-line*, através da Internet. Possibilitando que representantes de empresas consigam inserir documentos referentes aos seus projetos e salvá-los, através de arquivos enviados via Internet, na base de dados do servidor da IAMPS.

Esses dados salvos anteriormente serão acessados via *web* pelos avaliadores no momento em que os mesmos estiverem avaliando os projetos.

Fica pendente como trabalho futuro a avaliação desde o primeiro contato da unidade organizacional a ser avaliada com a Instituição Avaliadora até o relatório e a documentação gerados pela IA-MPS em relação aos resultados alcançados na avaliação.

Ou seja, englobando desde a avaliação inicial. Também fica pendente a implementação da forma *off-line*.

8. Referências Bibliográficas

CMMI. **Capability Maturity Model Integration**, Version 1.1. Pittsburgh: Carnegie Mellon University, 2007. Disponível em: <<http://www.sei.cmu.edu/cmmi/models/model-components-word.html>>.

CÔRTEZ, Mario Lúcio. **Modelos de Qualidade de Software**. Campinas: UNICAMP, 2001. 148 p.

JAVA. **Sun Microsystems - BR**. Disponível em: <<http://br.sun.com/>>.

MPS.BR. **Melhoria do Processo do Software Brasileiro**. Campinas: Sociedade SOFTEX, 2007. Disponível em: <http://www.softex.br/mpsbr/_guias/default.asp>.

PRESSMAN, Roger S.; **Engenharia de Software**. São Paulo: Markon, 1995. 1056 p.

ROCHA, Ana Regina C.; MALDONADO, José Carlos; WEBER, Kival Chaves. **Qualidade de Software: Teoria e Prática**. São Paulo: Prentice Hall, 2001. 310 p.

SOFTEX. **Site oficial**. Campinas: Sociedade SOFTEX, 2007. Disponível em: <http://www.softex.br/portal/_home/default.asp>.

SEI. **Software Engineering Institute**. Disponível em: <http://www.sei.cmu.edu>. Acesso em: 27 mar. 2007.

TSUKUMO, Alfredo et al. **Qualidade de Software: Visões de Produto e Processo de Software**. Piracicaba: II ERI da SBC, 1997.

[6.15] Projeto COMPGOV: Biblioteca Compartilhada para Componentes de E-gov

Entidade: 1 Centro de Estudos e Sistemas Avançados do Recife (C.E.S.A.R), Recife, Pernambuco, Brasil e 2Reuse in Software Engineering (RiSE), Recife, Pernambuco, Brasil

Autor: Eduardo Santana de Almeida^{1, 2} - esa@cesar.org.br

Abstract. *Software reuse is a key aspect for companies obtain the benefits related to productivity, quality and cost reduction. However, in order to achieve these benefits a combination of technical and non-technical issues should be addressed. This paper presents a summary of the COMPGOV project - an important effort in software reuse joining universities and companies - discussing since its motivation and origins until the main results.*

Resumo. *Reuso de software é um aspecto chave para empresas obterem melhorias relacionadas à produtividade, qualidade e redução de custos.*

Entretanto, alcançar esses benefícios envolve uma combinação sistemática de fatores técnicos e não técnicos. Este artigo apresenta um resumo do projeto COMPGOV, o qual corresponde a uma grande iniciativa em reuso de software envolvendo universidades e empresas, discutindo desde a sua motivação e origem até os principais resultados obtidos.

1. Introdução

Uma das razões mais convincentes para a adoção de abordagens de desenvolvimento de software, com ou sem objetos, é a premissa de reutilização. A idéia é construir software através da utilização de artefatos ou conhecimento já existentes primariamente pela montagem e substituição de suas partes interoperáveis. Estes componentes abrangem desde controles de interfaces de usuários, como *listboxes* e *HTML browsers*, até componentes para distribuição e domínios específicos [1]. As implicações de redução de tempo de desenvolvimento e melhoria da qualidade do produto tornam esta abordagem muito atrativa.

Segundo [19], a reutilização no contexto da engenharia de software não é uma idéia recente. De acordo com Pressman, os programadores têm utilizado idéias, abstrações e processos desde os primeiros dias da computação, mas as primeiras abordagens à reutilização eram as da prática corrente. Atualmente, sistemas complexos e de alta qualidade baseados em computador necessitam ser construídos em curtos períodos de tempo. Isso indica uma abordagem mais organizada e voltada à reutilização.

Um dos grandes pesquisadores na área de reutilização, Martin Griss, visualiza a mesma como uma simples e bem conhecida idéia [3]. A partir da necessidade de se construir um novo software, deve-se utilizar componentes previamente construídos.

Deste modo, tem-se a redução nos custos de desenvolvimento, testes, documentação e manutenção.

A reutilização é composta por uma variedade de técnicas que tem o objetivo de reaproveitar ao máximo o trabalho de análise, projeto e implementação já concluído. O objetivo é não reinventar a mesma idéia cada vez que um novo projeto tiver que ser desenvolvido, mas sim organizar o trabalho já realizado e implantá-lo imediatamente em um novo contexto. Deste modo, mais produtos podem ser entregues em menores tempos, uma vez que as melhorias realizadas em um segmento de projeto refletir-se-ão em todos os projetos nos quais este está sendo utilizado e, por fim, tem-se uma melhoria de qualidade, visto que os componentes reutilizados já foram bem testados.

Conforme discutido amplamente na literatura [4, 5, 6], uma solução de reutilização de software no contexto de empresas, envolve uma combinação de aspectos técnicos, como: processos, métricas, ferramentas e ambientes, assim como aspectos não técnicos, tais como: educação na área, criação de uma cultura, treinamentos e comprometimento organizacional.

Neste contexto de reutilização de software foi desenvolvido o projeto COMPGOV. O projeto envolveu o desenvolvimento de soluções na área formada pela combinação de aspectos técnicos. Devido a sua amplitude e com base nos casos de sucesso e falhas publicados na literatura [3-11], onde a combinação de academia e indústria é essencial, foi definido um consórcio formado por universidades (Universidade Federal de Pernambuco, Universidade Federal da Paraíba e Universidade Estadual de Campinas) e empresas (Centro de Estudos e Sistemas Avançados do Recife – C.E.S.A.R, Ci&T Software e Centro de Pesquisas Renato Archer - CePRA). Os aspectos não técnicos, como: treinamentos, criação da cultura e comprometimento organizacional ficaram a cargo das empresas participantes do projeto, porém, não como um objetivo oficial.

2. Objetivos e Justificativa

A solução de reutilização de software proposta para o projeto COMPGOV foi baseada em cinco aspectos chaves:

- ♦ **Processos de Reutilização:** um aspecto importante em um programa de reutilização diz respeito a como desenvolver artefatos reutilizáveis. Assim, o projeto investigou três áreas principais relacionadas a processos de reutilização: Engenharia de Domínio [12], Linhas de Produto [13] e Desenvolvimento Baseado em Componentes. Com base nas pesquisas realizadas, foram desenvolvidos três processos: o primeiro para desenvolver artefatos reusáveis (desenvolvimento *para* reuso), o segundo para desenvolver aplicações reutilizando os artefatos (desenvolvimento *com* reuso) e, por fim, um processo de teste de componentes para aumentar a qualidade dos artefatos desenvolvidos;
- ♦ **Ferramentas de Apoio:** uma vez definido os processos para se desenvolver software reutilizável, é também importante para um programa de reutilização, ter meios para automatizar parte das suas tarefas. Deste modo, o projeto investigou a área de ambientes e ferramentas de apoio a reutilização visando o desenvolvimento de ferramentas de apoio aos processos de desenvolvimento de soluções reutilizáveis e teste de componentes;
- ♦ **Processos de Certificação de Componentes:** atualmente, um aspecto chave na área de desenvolvimento baseado em componentes corresponde a como garantir a qualidade dos artefatos (componentes) desenvolvidos [14]. No projeto COMPGOV, essa preocupação era ainda maior, uma vez que o mesmo visava à criação de um repositório de componentes público. Assim, um processo de certificação bem definido era um requisito chave na solução, para permitir que apenas componentes com um bom grau de qualidade fosse armazenado no repositório;
- ♦ **Repositório de Componentes:** o principal objetivo do projeto era o desenvolvimento de um repositório. Um repositório de componentes pode ser entendido como uma base para o armazenamento, a busca e a recuperação de artefatos reutilizáveis. No projeto COMPGOV, essa definição envolvia alguns aspectos adicionais, como: a criação de repositórios distribuídos, que com isso adicionava outros requisitos não funcionais, como, por exemplo, segurança e escalabilidade e a definição de um modelo de negócios para o repositório; e
- ♦ **Modelo de Negócios:** por fim, uma vez desenvolvido o repositório de componentes, o projeto tinha como objetivo também, a definição de um modelo de negócios para os componentes desenvolvidos e armazenados no repositório.

Assim, os modelos de negocio existentes foram analisados de modo a definir uma solução inicial para o repositório.

Vale ressaltar que esses aspectos são também apontados na literatura como fatores importantes para programas de sucesso com reuso [5, 6, 9].

Com base nos objetivos definidos anteriormente, foram definidos as instituições responsáveis. Entretanto, uma segunda instituição deveria auxiliar e acompanhar a conclusão. A responsabilidade de gerenciamento e conclusão do projeto ficou a cargo do C.E.S.A.R. A Tabela 1 mostra a estrutura definida para o desenvolvimento do projeto.

Tabela 1. Estrutura para Desenvolvimento do Projeto

Objetivo	Instituição
Processo de Desenvolvimento <i>para</i> reuso	UFPE e UNICAMP
Processo de Desenvolvimento <i>com</i> reuso	UNICAMP e UFPE
Processo de Teste de componentes	UNICAMP e CenPRA
Ferramentas de Apoio	UNICAMP e Ci&T
Processo de Certificação de Componentes	CenPRA e UFPE
Repositório de Componentes	UFPB e UFPE
Implementação do Repositório	C.E.S.A.R e Ci&T
Modelo de Negócios	Ci&T e C.E.S.A.R

3. Metodologia de Execução

A metodologia para o desenvolvimento do projeto consistiu de reuniões de acompanhamento, discussões através de lista de *emails* e workshops. As reuniões de acompanhamento foram realizadas através da utilização de programas para conferências remotas, juntamente com processos de acompanhamento e gerenciamento de projetos.

Adicionalmente, uma estrutura de desenvolvimento remoto foi definido entre as empresas C.E.S.A.R e Ci&T para o desenvolvimento do repositório.

Além disso, o projeto contou também com workshops de acompanhamento e troca de conhecimento entre as instituições envolvidas. Os workshops foram realizados em Recife e Campinas, com todas as instituições participantes.

4. Resultados Obtidos

O projeto COMGOV apresentou diversos resultados relacionado ao escopo do projeto, dentre os quais devem ser destacados:

- ♦ **Produtos:** como discutido anteriormente, um dos aspectos principais objetivos do projeto consistia no desenvolvimento de um repositório de componentes. Deste modo, o repositório distribuído foi desenvolvido, juntamente com um ambiente de apoio ao processo de desenvolvimento *com* reuso e uma ferramenta para teste de componentes;
- ♦ **Métodos:** quatro processos foram desenvolvidos no contexto do projeto. Dois processos foram desenvolvidos para o desenvolvimento de software reutilizável: o primeiro, que corresponde ao desenvolvimento *para* reuso e o segundo, que contempla o desenvolvimento *com* reuso. Além disso, foram definidos dois processos para garantir a qualidade dos artefatos desenvolvidos: um processo de teste de componentes e o outro para certificação de componentes do repositório;

- ♦ **Artigos:** devido ao seu caráter inovador, o projeto teve um impacto significativo na comunidade de reutilização de software nacional e mundial. O projeto gerou cerca de 50 artigos publicados em importantes eventos e periódicos nacionais e internacionais;
- ♦ **Recursos Humanos:** outro aspecto importante foi a formação de recursos humanos na área. Neste aspecto, o projeto foi importante para a consolidação de disciplinas existentes na pós-graduação da UFPE, juntamente com a formação de cerca de sete mestres e dois doutores;
- ♦ **Dissertações e teses:** com base nos desafios do projeto, foram geradas dissertações nas áreas de métodos e processos para desenvolvimento de componentes, teste de componentes, ferramentas de apoio e processos de reutilização. Uma das dissertações foi também premiada como a melhor dissertação de mestrado na área de qualidade de software [15]. Além disso, outras pesquisas oriundas dos problemas e desafios identificados ao longo do projeto vêm sendo exploradas em dissertações de mestrado e teses de doutorado em algumas instituições, como, por exemplo, a UFPE e UNICAMP; e
- ♦ **Parcerias:** nenhuma nova parceria foi estabelecida com base no projeto. Entretanto, vale ressaltar que o mesmo contribuiu para estreitar os laços entre as empresas e grupos de pesquisas participantes do projeto. Esse aspecto é importante, pois pode contribuir para a formação de novos projetos em conjunto no futuro.

5. Aplicabilidade dos Resultados

Conforme discutido na seção anterior, o projeto apresentou resultados importantes no contexto de pesquisa, desenvolvimento e inovação na área de reutilização de software.

Mesmo outros projetos internacionais de reutilização espalhados pelo mundo [7, 11, 16], não apresentaram resultados tão significativos. Por outro lado, mesmo ainda em caráter de protótipos, as ferramentas desenvolvidas, juntamente com os processos definidos, se refinados e calibrados podem ser de grande valia para as empresas interessadas em obterem os benefícios inerentes de reutilização. Acredita-se que este objetivo será mantido, uma vez que as universidades e empresas participantes continuam interessadas nos tópicos.

Outro aspecto crucial que deve ser destacado é que o projeto serviu também de insumo para a criação de novas empresas na área, como: a *Digital Assets* (DA) e o *Reuse in Software Engineering* (RiSE).

6. Características Inovadoras

A área de reutilização apresenta uma série de processos de reuso publicados [17], principalmente, envolvendo os aspectos de desenvolvimento *para* e *com* reuso.

Entretanto, existem poucos processos experimentados de fato na prática. Além disso, poucos processos contemplam características de teste e certificação de componentes.

O mesmo pode ser dito dos repositórios existentes no mercado [18]. No entanto, os repositórios desenvolvidos não contemplam algumas características importantes, como: distribuição e replicação, juntamente com questões relacionadas a negócios (modelo de negócios). Assim, pode ser dito que os processos desenvolvidos durante o projeto, assim como o modelo do repositório em si, são inovações relevantes. Porém, os mesmos precisam ser mais experimentados e refinados antes de serem levados de fato ao mercado.

7. Conclusão

A idéia de reutilização de software foi proposta há cerca de quarenta anos atrás, quando McIlroy propôs uma indústria de componentes como uma possível solução para o

problema da crise do software [19]. Ao longo dos anos, universidades e empresas vêm tentando alcançar os benefícios oriundos da reutilização como aumento de produtividade, qualidade das aplicações e redução de custos. No entanto, uma solução de reutilização para ser efetiva, principalmente, em larga escala envolve uma combinação de aspectos técnicos e não técnicos.

Além dessa combinação sistemática, a literatura tem mostrado que um projeto desse tipo para ser conduzido apenas por uma universidade é improvável de obter sucesso. Nesse contexto foi definido o projeto COMPGOV. O projeto envolveu a participação de universidades e empresas trabalhando juntos no desenvolvimento de soluções na área de métodos, processos e ferramentas de reutilização. Devido aos resultados alcançados, o projeto pode ser considerado um sucesso, principalmente, devido ao seu impacto na comunidade de pesquisa mundial com uma série de artigos, protótipos e estudos de caso que contribuíram para o avanço da área. Além disso, no contexto industrial, o projeto também foi importante para o surgimento de novas empresas com foco em reutilização de software.

Agradecimentos

O autor do artigo gostaria de agradecer a todas as instituições participantes do projeto pelas discussões que contribuíram muito para a consolidação e o surgimento de novas idéias.

Referências

- [1] C. W. Krueger, **Software Reuse**, *ACM Computing Surveys*, Vol. 24, No. 02, June, 1992, pp. 131-183.
- [2] M. L. Griss, **Software Reuse Experience at Hewlett-Packard**, *16th IEEE International Conference on Software Engineering (ICSE)*, Sorrento, Italy, May, 1994, pp. 270.
- [3] D. Card, E. Comer, **Why Do So Many Reuse Programs Fail?**, *IEEE Software*, Vol. 11, No. 05, September/October, 1994, pp. 114-115.
- [4] W. B. Frakes, S. Isoda, **Success Factors of Systematic Software Reuse**, *IEEE Software*, Vol. 12, No. 01, September, 1994, pp. 15-19.
- [5] W. B. Frakes, K. C. Kang, **Software Reuse Research: Status and Future**, *IEEE Transactions on Software Engineering*, Vol. 31, No. 07, July, 2005, pp. 529-536.
- [6] M. Aoyama, **CBSE in Japan and Asia**, in *Component-Based Software Engineering: Putting the Pieces Together*, G. T. Heineman, B. Councill, Addison-Wesley, 2001, pp. 818.
- [7] D. Bauer, **A Reusable Parts Center**, *IBM Systems Journal*, Vol. 32, No. 04, September, 1993, pp. 620-624.
- [8] R. L. Glass, **Reuse: What's Wrong with This Picture?**, *IEEE Software*, Vol. 15, No. 02, March/April, 1998, pp. 57-59.
- [9] M. L. Griss, **Making Software Reuse Work at Hewlett-Packard**, *IEEE Software*, Vol. 12, No. 01, January, 1995, pp. 105-107.
- [10] B. McGibbon, **Status of CBSE in Europe**, in *Component-Based Software Engineering: Putting the Pieces Together*, G. T. Heineman, B. Councill, Addison-Wesley, 2001, pp. 818.
- [11] K. Czarnecki, U. W. Eisenecker, **Generative Programming: Methods, Tools, and Applications**, Addison-Wesley, 2000, pp. 832.
- [12] P. Clements, L. Northrop, **Software Product Lines: Practices and Patterns**, Addison-Wesley, 2001, pp. 608.
- [13] B. Meyer, C. Mingins, H. Schmidt, **Providing Trusted Components to the Industry**, *IEEE Computer*, Vol. 31, No. 05, May, 1998, pp. 104-105.

- [14] A. Alvaro, **Software Component Certification: A Component Quality Model**, Universidade Federal de Pernambuco, Brasil, 2005.
- [15] S. D. Kim, **Lessons learned from a nationwide CBD promotion project**, *Communications of the ACM*, Vol. 45, No. 10, October, 2002, pp. 83-87.
- [16] E. S. Almeida, A. Alvaro, D. Lucrédio, V. C. Garcia, S. R. L. Meira, **A Survey on Software Reuse Processes**, *IEEE International Conference on Information Reuse and Integration (IRI)*, Las Vegas, Nevada, USA, August, 2005, pp.66- 71.
- [17] E. S. Almeida, A. Alvaro, V. C. Garcia, J. C. C. P. Mascena, V. A. A. Burégio, L. M. Nascimento, D. Lucrédio, S. R. L. Meira, **CRUISE: Component Reuse in Software Engineering**, C.E.S.A.R e-books, 2007, pp. 202.
- [18] R. S. Pressman, **Software Engineering: A Practitioner's Approach**, McGraw-Hill, 2005, pp. 880.

[6.18] Programa Integrado de Melhoria de Processos de Desenvolvimento de Software

Entidade: 1CPM Braxis / Unitech - Av. ACM, 2487, CEP: 40280-630, Salvador, BA

Autores: Likiso Hattori¹, Carol Passos¹ -

{likiso,carol.passos,Renata.dias,livia.graca}@cpmbraxis.com

Abstract. *This article presents the Process Improvement Integrated Program of CPM Braxis/Unitech whose goal is leveraging the quality and productivity of their services, searching through international certifications (ISO and CMMI) enable to an insertion in the international market. The work is based on concepts related to the theme of Quality Software and intends to demonstrate how an adherent process to international standards can contribute to gains in quality and productivity.*

Resumo. *Este artigo apresenta o Programa Integrado de Melhoria de Processos da CPM Braxis/Unitech cujo objetivo é alavancar a qualidade e a produtividade dos seus serviços, buscando através de certificações internacionais (ISO e CMMI) habilitar-se a uma inserção no mercado internacional. O trabalho fundamenta-se nos conceitos relacionados ao tema Qualidade de Software e tem como objetivo de demonstrar como um processo aderente a padrões internacionais pode contribuir com ganhos em termos de qualidade e produtividade.*

1. Introdução

Motivadas pelos avanços da tecnologia da informação e pelas constantes mudanças no mercado, as organizações têm dado maior ênfase à estruturação e padronização de seus processos. A pressão por diferenciais, a competição acirrada pelo mercado e o ambiente de negócio cada vez mais complexo são fatores que influenciam neste cenário e conduzem para uma estratégia focada em processos de negócio. A compreensão do papel dos modelos e normas de referência neste contexto tem contribuído para o desenvolvimento de novos mecanismos para melhoria dos processos organizacionais.

A melhoria da qualidade do processo de desenvolvimento de software de uma organização garante uma maior consistência dos produtos finais com os requisitos especificados pelo cliente [Pádua 2003]. O principal produto desta abordagem é a estruturação de um processo integrado e padronizado, que define as fases, atividades e boas práticas relacionadas ao desenvolvimento de software. Neste contexto, o Programa Integrado de Melhoria de Processos da CPM Braxis/Unitech tem como objetivo de alavancar a qualidade e a produtividade dos seus serviços, buscando através de certificações internacionais, como ISO e CMMI, habilitar-se a uma inserção no mercado internacional.

A existência de um processo estruturado e padronizado é um fator primordial para o desenvolvimento de software com qualidade e com a adoção de padrões internacionais, como normas e modelos de processo, pode-se reduzir o esforço necessário para realizar um trabalho e aumentar a qualidade e consistência dos resultados finais [Chrissis, et al 2003].

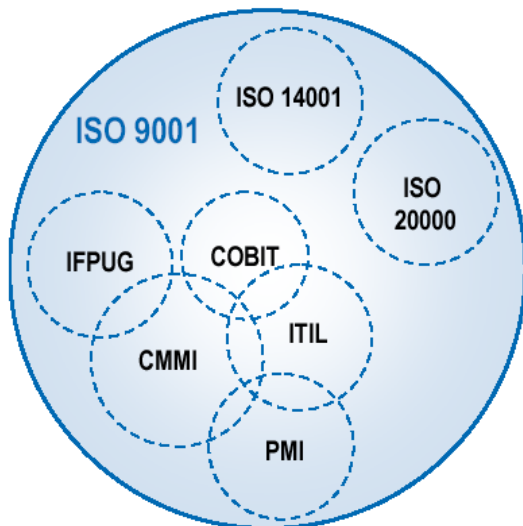
Este artigo descreve as características e componentes deste programa, sua arquitetura e implementação, além de um resumo dos resultados já obtidos por ele. O trabalho fundamenta-se nos conceitos relacionados ao tema Qualidade de Software e tem como objetivo demonstrar como um processo aderente a padrões internacionais pode contribuir com ganhos em termos de qualidade e produtividade e consequente abertura de novos mercados para a organização no Brasil e no exterior.

2. Objetivos e Justificativas

O Programa Integrado de Melhoria de Processos da CPM Braxis/Unitech tem foco na melhoria da qualidade dos processos de desenvolvimento de software em consonância com padrões internacionais (ISO, CMMI, entre outros) e as respectivas certificações. O seu principal objetivo é definir e implementar um processo aderente a padrões internacionais e demonstrar como este processo pode contribuir com ganhos em termos de qualidade dos produtos gerados e produtividade das equipes de projeto.

A existência de um processo estruturado e padronizado é um fator primordial para o desenvolvimento de software com qualidade. A adoção de padrões internacionais de referência promove a redução do esforço necessário para realizar um trabalho ao mesmo tempo que aumenta a qualidade e consistência dos resultados produzidos. Além disso, é uma maneira de medir a maturidade e a capacidade dos processos de uma organização, visto que eles criam uma estrutura para a realização de avaliações confiáveis e consistentes.

A melhoria da qualidade e produtividade no processo de desenvolvimento de software da organização pode ser alcançada através de um processo padrão, desempenhado por indivíduos devidamente capacitados e motivados, o que permite que redundâncias sejam evitadas, garantindo qualidade no custo e prazo planejados.



3. Metodologia

O programa engloba toda a organização CPM Braxis/Unitech, sendo que os processos das áreas funcionais são aderentes às Normas ISO-9001 e ISO-14001 e, em um futuro próximo, a ISO-20000 e os processos relacionados ao desenvolvimento de software são também aderentes ao Modelo CMMI. Vale ressaltar que, em janeiro de 2005, a empresa

conquistou o nível 2 do SW-CMM, em abril de 2006, alcançou o CMMI e os níveis 4 e 5 foram atingidos ainda no final de 2007.

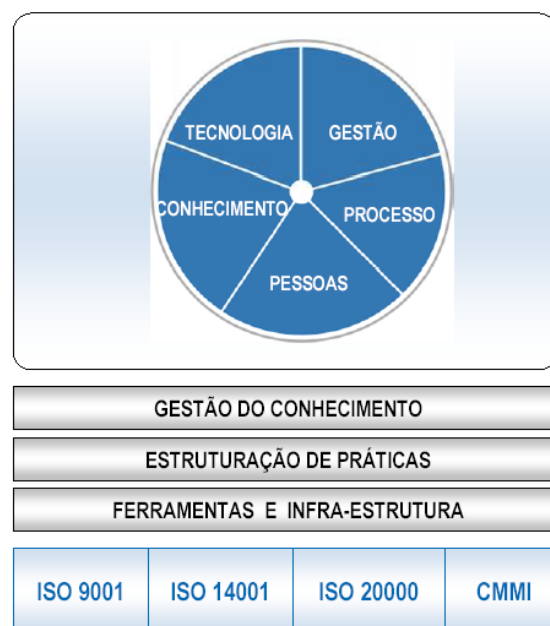
Figura 1. Normas e Modelos adotados pelo Programa Integrado de Melhoria de Processos da CPM Braxis/Unitech.

O programa define a estrutura e as atividades a serem realizadas pela CPM Braxis/Unitech em diversos projetos e ações que irão concretizar o processo de melhoria do processo de desenvolvimento de software, cujo objetivo principal é ser uma organização madura e atingir, assim, um novo patamar internacional e reconhecimento do mercado.

Este programa foi estruturado em 5 dimensões. A dimensão **Gestão** para direcionar o planejamento e acompanhamento de ações que viabilizam a estratégia de negócio da organização. A dimensão **Processo** se apoia na definição, coleta e análise de indicadores estabelecidos para medir a melhoria contínua dos processos. Para a dimensão **Pessoas**, fica claro que o programa provoca mudanças positivas sobre a natureza do trabalho da equipe e na postura do colaborador, que passa a ter maior sensibilidade sobre a sua contribuição junto a objetivos estratégicos, aprofundando sua responsabilidade e compreendendo o papel de suas tarefas (maior comprometimento).

Na dimensão **Tecnologia**, novas ferramentas de apoio foram introduzidas, buscando viabilizar a execução das práticas recomendadas pelo programa de melhoria. Por fim, para a dimensão **Conhecimento**, a estruturação de um processo para geração, captura, classificação, disseminação e (re) utilização do conhecimento derivado de processos e práticas organizacionais foi crucial para o desenvolvimento do programa conforme o esperado.

Figura 2. Programa Integrado de Melhoria de Processos da CPM Braxis/Unitech suas dimensões e infra-estrutura de apoio.



Cada uma destas dimensões tem influência nos resultados obtidos pelo programa de uma forma definitiva e relacionada.

4. Resultados Obtidos

Os objetivos do Programa Integrado de Melhoria de Processos da CPM Braxis/Unitech estão vinculados aos objetivos de negócio da organização. Desta forma, os indicadores

definidos demonstram as realizações e os resultados de cada projeto do programa de uma forma consolidada.

A medição e análise dos resultados obtidos denota a evolução da maturidade do processo de desenvolvimento de software. Alguns resultados mais relevantes deste programa, até o presente momento, foram:

- **Produtividade**

- Redução de 10% do esforço (h/h) de gestão de projeto;
- Redução de 57% do esforço de rastreabilidade de requisitos;
- Redução de 45% do esforço de codificação;

- **Qualidade**

- Redução de 10% dos erros de codificação;
- Aumento de 30% na taxa de efetividade de testes.

O programa de melhoria de fato está contribuindo para a construção de diferenciais competitivos esperados pela empresa.

5. Aplicabilidade dos Resultados

Os resultados obtidos pelo Programa Integrado de Melhoria de Processos de Desenvolvimento de Software, conforme mostrados no item anterior, estão sendo aplicados diretamente nos processos produtivos da empresa contribuindo para construir os diferenciais competitivos necessários para a sua globalização. Esta estratégia permitiu um crescimento com qualidade, produtividade e rentabilidade.

A estratégia proporcionou que a empresa conquistasse novos clientes, principalmente no mercado globalizado onde viabilizou contratos para serviços *offshore* para empresas americanas e européias.

6. Características Inovadoras

O Programa Integrado de Melhoria de Processos da CPM Braxis/Unitech é visto como um fator estruturante de competitividade e inovação de práticas. A idéia principal deste programa é gerar uma nova forma de fazer algo que já existe, buscando uma maior produtividade, eficiência e eficácia do processo. As normas e modelos de referência foram institucionalizados, permitindo uma reflexão sobre as suas práticas e o redirecionamento criativo de tarefas e comportamentos. Com foco na melhoria contínua, o programa traduz o conhecimento da organização em produtividade, qualidade e, portanto, em uma maior capacidade de atingir seus objetivos estratégicos.

Assim, é possível uma perspectiva global e sistemática para potencializar a competitividade da organização.

7. Conclusão e Perspectivas Futuras

As organizações mais competitivas são aquelas que promovem melhorias em processos que já atendem aos objetivos requeridos. Seguindo este paradigma, a proposta deste trabalho é o estabelecimento de um programa que viabilize a estruturação e melhoria contínua de processos integrados e aderentes a padrões internacionais.

Neste artigo foram apresentados os aspectos que compõem o Programa Integrado de Melhoria de Processos da CPM Braxis/Unitech, como seus objetivos, justificativa e relevância, abrangência e metodologia, além dos principais resultados obtidos.

Como trabalhos futuros relacionados a este programa podemos citar a institucionalização do processo aderente a ISO-20000 para certificação em 2008 que está em execução e a preparação para futuras certificações, como: e-SCM para outsourcing que a empresa já

desenvolveu e utiliza uma metodologia aderente e está criando processos adequados para futura avaliação, assim como a ISO-27001 para segurança corporativa.

8. Referências Bibliográficas

CHRISSIS,M.; KONRAD,M.; SHRUM,S. (2003) "CMMI® – Guidelines for Process Integration and Product Improvement". SEI Series, Addison-Wesley, 1st edition.

PÁDUA, Wilson. (2003) "Engenharia de Software - Fundamentos, Métodos e Padrões". Editora LTC, 2st edition..

DAVENPORT, Thomas H.; PRUSAK Lawrence. (1998) "Conhecimento Empresarial: Como as Empresas Gerenciam o seu Capital Intelectual". Editora Campus, 7a edição.

COLLINS, H. (2003) "Enterprise Knowledge Portals". Amacom.

TERRA, J. C. (2001) "Portais Corporativos e Gestão do Conteúdo". Negócio, 2a edição.

WENGER, E. (2002) "Communities of practice: a brief introduction". Harvard Business School Press.

[6.21] Ambientes de Engenharia de Software Orientados a Corporação - Uma Infra-estrutura Computacional para a Definição, Execução e Melhoria de Processos de Software em Corporações

Entidade: COPPE/UFRJ - Universidade Federal do Rio de Janeiro
Programa de Engenharia de Sistemas e Computação
Av. Horácio Macedo, 2030, Prédio do Centro de Tecnologia, Bloco H, Sala 319, Caixa Postal 68511 CEP 21941-914 - Rio de Janeiro, RJ

Autores: Gleison Santos, Ana Regina Rocha, Guilherme Horta Travassos - gleison, darocha, ght@cos.ufrj.br

Abstract. *The definition of software processes based on national or international reference models is important, but not sufficient to guarantee the quality of software products and processes. In order to increase the efficacy and efficiency of those processes and to increase the quality of software products, it is fundamental to guarantee the adequate execution of software processes. This work presents an approach to guarantee software processes quality with the support of Knowledge Management in the TABA Workstation, an enterprise-oriented software development environment.*

Resumo. *Corporações compostas por organizações que desenvolvem software de forma independente e com certo grau de independência precisam gerenciar a diversidade de processos existentes dentro do âmbito corporativo. Como forma de aumentar a competitividade e controle sobre o desenvolvimento de software, as corporações também devem possuir mecanismos que possibilitem às organizações alcançarem maiores níveis de maturidade. Este artigo apresenta uma abordagem para definição e criação de Ambientes de Engenharia de Software Orientados a Corporações que visam atender a estes objetivos.*

1. Introdução

Uma característica cada vez mais comum no cenário de desenvolvimento de software é a existência de corporações com organizações subordinadas que desenvolvem software de forma independente entre si. Estas organizações muitas vezes possuem diferentes necessidades e realidades em relação ao desenvolvimento de software, por exemplo, nichos de mercado com características diferentes ou diferentes níveis de maturidade. Por exemplo, corporações cujas organizações apenas realizam manutenção de sistemas embarcados e outras que desenvolvem sistemas de informação para automatização de atividades de apoio e outras que desenvolvem software para terceiros; corporações em que uma das organizações têm o desenvolvimento de software aderente a um nível de maturidade do MPS.BR [SOFTEX 2007] ou do CMMI [Chrissis *et al.* 2006] e as demais desenvolvem software de forma *ad hoc*; ou ainda, corporações cujas organizações mesmo estando aderentes a algum modelo de maturidade tenham seus processos institucionalizados em níveis distintos de maturidade e capacidade. Outro fator que tem aumentado a complexidade do desenvolvimento de software ao longo do tempo é o número de processos que são executados para a conclusão de um projeto de software. Organizações desenvolvedoras de software têm necessidade de executar diversos processos integrados ao processo de desenvolvimento para a execução de um projeto de software (por exemplo, processos de apoio como Garantia de Qualidade) e, também, processos executados de forma independente dos processos de desenvolvimento em si, mas também relacionados à produção de software ou gerência e estruturação do desenvolvimento (por exemplo, processos como Aquisição e Gerência de Projetos de Melhoria e Gerência de Portfólio de Projetos).

A definição e construção dos Ambientes de Desenvolvimento de Software Orientados a Organização [Villela 2004] supriu a necessidade de se apoiar a utilização do conhecimento organizacional, e não só o conhecimento de domínio provido por especialistas (conforme identificado por OLIVEIRA [1999]), durante o desenvolvimento de software e, também, a necessidade de se apoiar o desenvolvimento de software em

uma organização em particular. Entretanto, esta abordagem tem limitações ao: (i) apoiar apenas a definição e execução de processos de desenvolvimento ou manutenção de software enquanto vários outros processos precisam ser definidos e executados no contexto geral de produção de software em uma organização; e (ii) considerar apenas a existência de organizações independentes enquanto há corporações que precisam gerenciar os processos de software nas suas organizações subordinadas;

Os Ambientes de Engenharia de Software Orientados a Corporação visam suprir estas limitações e podem ser considerados uma evolução dos Ambientes de Desenvolvimento de Software Orientados a Organização em dois sentidos: (i) Evolução de Ambientes de *Desenvolvimento* de Software para Ambientes de *Engenharia* de Software, o que significa estender a definição dos ambientes para outros processos além do de desenvolvimento e de manutenção, apoiando a execução e a gerência dos demais processos de Engenharia de Software; e (ii) Evolução de Ambientes Orientados a *Organização* para Ambientes Orientados a *Corporação*, o que significa permitir a configuração de ambientes para corporações e, a partir desses, para as organizações que as compõem.

Dessa forma, o objetivo deste artigo é descrever os *Ambientes de Engenharia de Software Orientados a Corporação* (AESCorp), que fornecem o apoio computacional que possibilita a uma corporação, em relação aos processos de software, gerenciar a diversidade e os estágios de maturidade de cada uma das organizações que a compõem de forma adequada às suas necessidades. Além disso, esse apoio computacional permite às corporações e organizações serem capazes de gerenciar e controlar os diversos processos de software de que dispõem e/ou necessitem, bem como o conhecimento organizacional envolvido.

A próxima seção apresenta uma revisão da literatura sobre ambientes de desenvolvimento de software e apoio para definição, execução e melhoria de processos de software. A seção 3 apresenta as características dos AESCorp. A seção 4 apresenta a definição e criação dos AESCorp no contexto da Estação Taba enquanto a seção 5 descreve modificações realizadas na Estação Taba para que isso fosse possível. Por fim, a seção 6 apresenta as considerações finais e perspectivas futuras.

2. Apoio Computacional para Definição, Execução e Melhoria de Processos de Software

De acordo com a importância de as iniciativas de melhoria de processos para as organizações serem competitivas e, também, nas dificuldades que as organizações têm em implementar tais iniciativas devido ao custo e tempo necessários, é preciso realizar programas de melhoria mais acessíveis para a maior parte das organizações independentemente das características de cada organização [Amescua *et al.* 2006]. Uma forma de reduzir os custos associados com uma iniciativa de melhoria de processos de software é adotar ferramentas adequadas para automatizar certas tarefas e, assim, reduzir o esforço necessário para realizá-las. Apesar do grande número de publicações relatando ferramentas CASE e ambientes de desenvolvimento de software, não são muitos os relatos sobre a utilização de tal apoio ferramental em contextos de iniciativas de melhoria de processos de software ou que situam tal apoio ferramental como adequado a tais contextos. Entretanto, é comum em relatos sobre fatores de sucesso e dificuldades relacionadas à execução de iniciativas de melhoria de processos (por exemplo, [Niazi *et al.* 2005] [Santos *et al.* 2007]) a identificação da necessidade de uma infra-estrutura adequada como um importante fator de sucesso. A maior parte das organizações com baixos níveis de maturidade no desenvolvimento não possuem uma infra-estrutura adequada para iniciar a iniciativa de melhoria de processos de software [Santos *et al.* 2007]. O escopo do apoio ferramental para auxiliar na execução de iniciativas de melhoria de processos em organizações encontrados na literatura é variado. Os relatos mencionam desde a utilização de ferramentas no contexto de um processo específico, divulgação dos processos, adoção de modelos de maturidade etc.

Os requisitos de modelos de maturidade e capacidade, por seu aspecto abrangente e o pressuposto de haver diferentes tipos de processos distribuídos em diferentes níveis, apresentam uma maior complexidade em relação ao apoio ferramental por gerar a necessidade de automatização de um grande número de tarefas especializadas. Dessa forma, uma abordagem comum em organizações com estes requisitos é a adoção de ambientes (ou, pelo menos, um conjunto de ferramentas integradas que se comportam, de fato, como um ambiente no sentido clássico da acepção) para apoiar as tarefas necessárias.

FMESP [Canfora *et al.* 2006] provê o apoio necessário para a representação e gerência do conhecimento relacionado a processos de software a partir das perspectivas de modelagem e medição, ao integrá-las. O ambiente possui componentes para descrição de ontologias (e é baseado no uso de ontologias de processo e de medição), modelagem de processos e apoio à medição de software. Foi utilizado numa organização espanhola dedicada ao desenvolvimento e manutenção de software que obteve, como resultado, uma certificação ISO 9000. SoftPM [Wang e Li 2005] é um sistema integrado de apoio às atividades de gerentes de projetos, alta gerência, engenheiros, testadores, membros da área de garantia da qualidade e outros membros de áreas de apoio. Ajuda a compartilhar os dados coletados, entender o cronograma, esforço e qualidade do projeto e auxilia a comunicação entre os participantes do projeto. A Plataforma para Gerência da Qualidade é composta por quatro ferramentas (gerência de projetos, biblioteca de ativos de processo, garantia da qualidade e medição e análise) que podem ser combinadas de forma a aumentar a aderência ao CMM/CMMI. Há duas outras plataformas para engenharia de produtos e de apoio a serviços. Segundo os autores, o ambiente já foi utilizado com sucesso em mais de 100 empresas de software chinesas com uma redução de esforço de cerca de 35% em relação às atividades do grupo de processos, área de qualidade e gerentes de projetos. SPP [Jun *et al.* 2007] é baseado na estrutura do processo definido para a organização e tem o objetivo de apoiar a execução deste processo. Foi utilizado para a implantação de CMMI níveis 2 e 3 em organizações de software chinesas com foco no desenvolvimento de software embutido. ImPProS [Oliveira e Vasconcelos 2006] é um ambiente para melhoria de processos que apóia a adoção do IDEAL. Seu conjunto de funcionalidades inclui definição, simulação, execução, avaliação, melhoria e reuso de processos, análise e tomada de decisão com base nas avaliações dos processos, gerência de conhecimento relativa a processos e conversão da estrutura de processos com base em normas e modelos de qualidade. No entanto, não foram encontrados na literatura relatos da aplicação no ambiente na indústria [ImPProS 2008]. WebAPSEE [Costa *et al.* 2007] fornece apoio automatizado para a gestão de processos de software, sendo projetado para permitir a integração de vários serviços relacionados com uma visão bastante ampla do meta-processo de software. A ferramenta apóia desde a concepção e levantamento de requisitos do processo até a realização da análise *post mortem* dos processos, passando pelo controle da execução dos processos de forma flexibilizada. O WebAPSEE é, hoje, utilizado em projetos [Paxiúba *et al.* 2007], porém não foram encontrados na literatura relatos da aplicação prática na indústria.

A Estação Taba é composta por um conjunto de ambientes e ferramentas que auxiliam os engenheiros de software na definição, execução, avaliação e melhoria dos processos de desenvolvimento e manutenção. Estes ambientes, também, possuem funcionalidades de gerência de conhecimento integradas às atividades dos processos apoiados de forma a preservar o conhecimento organizacional e aumentar a institucionalização destes processos. As ferramentas disponíveis, atualmente, apóiam os níveis G, F, E, D e C do MR-MPS e os níveis equivalentes do CMMI [Montoni *et al.* 2007]. Atualmente, há várias teses de doutorado em andamento que visam a definição de ferramentas (e, possivelmente, alterações em ferramentas existentes) para a adequação aos níveis A e B do MR-MPS e equivalentes do CMMI. Os resultados de uso da Estação Taba incluem o seu papel importante para a obtenção de avaliações CMMI e MPS.BR em diversas empresas brasileiras. Como dados quantitativos, pode-se mencionar a redução do tempo

gasto em retrabalho nos projeto de 44% para 7% decorrente de iniciativa de melhoria de processo, realizada com o uso de ferramentas de apoio da Estação Taba em uma destas empresas [Ferreira *et al.* 2006] [Ferreira *et al.* 2007].

3. Características de um Ambiente de Engenharia de Software Orientado a Corporação

3.1 Evolução de Ambientes de Desenvolvimento de Software para Ambientes de Engenharia de Software

A principal questão relacionada à evolução dos Ambientes de *Desenvolvimento* de Software (ADS) para os Ambientes de *Engenharia* de Software (AES) é a existência de uma grande diversidade de processos existentes no contexto de organizações que produzem software. Cada um destes diferentes tipos de processos de software deve poder ser definido, executado e melhorado. A execução e gerência de tais processos devem ser apoiadas através da utilização de ambientes centrados em processo específicos a cada um destes processos. Um ADS é, dessa forma, apenas um entre os possíveis AES.

Definição de Processos - A definição de um processo de software, ou de um conjunto de processos inter-relacionados, geralmente é feita com base no contexto em que este processo está inserido e, também, no conhecimento já existente sobre processos em geral. Exemplos de conhecimento especializado sobre processos de software incluem modelos de maturidade e capacidade e normas internacionais relacionadas a processos de software, boas práticas e lições aprendidas, conhecimento sobre engenharia de software, conhecimento de especialistas em processos ou no contexto em que o processo será utilizado etc. O contexto em que o processo está inserido inclui o contexto de execução dos processos, aspectos culturais que podem influenciar a utilização dos processos, experiências anteriores relacionadas à execução (*ad hoc* ou não) dos processos, versões anteriores dos processos, necessidades e objetivos definidos para os processos. A estrutura de um processo pode ser bastante dinâmica e os resultados esperados da execução de uma atividade podem depender da execução de outros processos em contextos diferentes ou complementares. Por exemplo, se houvesse neste processo atividades referentes a garantia da qualidade de processo e produto pode-se imaginar a possibilidade de estas atividades comporem, de certa forma, um processo específico para a equipe de garantia da qualidade da organização. Assim, o processo de garantia da qualidade seria formado por atividades presentes em um macro-processo específico para a área de garantia da qualidade e também nas atividades de garantia da qualidade presentes nos processos dos diversos projetos de desenvolvimento de software na organização. O processo de garantia da qualidade teria, então, uma interface de comunicação e consumo e geração de produtos intermediários com vários outros processos.

Dessa forma, para Ambientes de Engenharia de Software é importante que outros processos além dos de desenvolvimento e manutenção possam ser definidos e, caso os processos se relacionem entre si, deve ser possível a identificação desse relacionamento ou dependência entre suas atividades.

Execução de Processos - Uma vez que um processo for definido é importante prover apoio à sua execução. Esta execução é tipicamente feita através de um ambiente específico centrado em processo e composto por um conjunto de ferramentas adequado ao propósito do ambiente e ao apoio necessário para a execução do processo no qual o ambiente é baseado. Estes ambientes devem possuir, também, uma base de dados onde informações sobre a execução do processo e das ferramentas são armazenadas. Além disso, podem acessar elementos externos como outras ferramentas, ambientes e bases de dados para dar apoio à execução do processo.

Os Ambientes de Engenharia de Software podem ser utilizados para apoiar a execução de quaisquer dos processos de software que tenham sido definidos no contexto de uma

organização e não apenas os de desenvolvimento e manutenção.

Melhoria de Processos - Uma das principais fontes para a melhoria de processos de software é a análise do comportamento dos processos ao longo de uma ou várias execuções. Esta análise pode levar em consideração, por exemplo, informações coletadas durante a execução do processo, necessidades do negócio e conhecimento adquirido externamente. Após a melhoria ter sido identificada, uma nova versão melhorada do processo é definida e disponibilizada para uso e a versão anterior é descontinuada.

Ambientes de Engenharia de Software possuem mecanismos que permitem a identificação de melhorias e disponibilização de novas versões dos diferentes processos de software em uso. Além disso, visto que pode ser necessário implantar melhorias em processos ainda em execução, é possível a alteração de processos em execução sem a perda de informações e dados uma vez que uma necessidade de melhoria tenha sido identificada.

Gerência dos Ativos de Processos - A definição, a execução e a melhoria de processos podem se beneficiar da existência e gerência adequada de ativos de processo, ou seja, artefatos que sejam considerados úteis para atender as necessidades de negócio da organização. Processos de software são, obviamente, parte do conjunto de ativos de processos de uma organização. Outros exemplos incluem artefatos, roteiros de documentação, itens de conhecimento (como diretrizes, lições aprendidas, melhores práticas etc.), ferramentas etc. Uma base responsável pela gerência de tais artefatos, geralmente denominada biblioteca de ativos de processos, deve ser capaz de armazenar os documentos considerados relevantes, garantir que eles sejam disponibilizados aos interessados e garantir que suas evoluções sejam controladas através de métodos apropriados de gerência de configuração e de garantia da qualidade.

Nos Ambientes de Engenharia de Software esta biblioteca contém mecanismos para apoiar todos os processos em uso e está integrada às funcionalidades existentes para a definição, execução e melhoria de processos de software.

Gerência de Conhecimento - A gerência de conhecimento é um importante mecanismo de apoio à execução de processos e pode, também, fornecer subsídios à melhoria de processos. A institucionalização da gerência de conhecimento pode garantir uma maior competitividade da organização e melhor capacitação de seus colaboradores. Através da institucionalização de uma rede de especialistas e de um mecanismo de apoio à troca de informações pode-se garantir que o conhecimento seja prontamente disponibilizado e compartilhado na organização aumentando a eficiência na realização das tarefas.

Em Ambientes de Engenharia de Software o ferramental disponível apóia a execução das atividades presentes nos diversos processos de software existentes. Para isso existe um mecanismo que permite a aquisição de itens de conhecimento ao longo das atividades do processo sendo executado, seu empacotamento, disponibilização e manutenção. Além disso, o acesso ao conhecimento acumulado ao longo do tempo é disponibilizado através de mecanismos e ferramentas específicos, de acordo com a atividade sendo executada. Durante as ações realizadas para a melhoria de processos tais itens de conhecimento estão disponíveis para que se investiguem possíveis pontos de melhoria.

3.1 Evolução de Ambientes Orientados a Organização para Ambientes Orientados a Corporação

A evolução dos Ambientes Orientados a *Organização* para os Ambientes Orientados a *Corporação* está na criação de um ambiente adaptado às necessidades de uma Corporação na gerência de suas atividades relacionadas a software e, também, à gerência de suas organizações que desenvolvem software. O aspecto mais importante a

ser gerenciado é a diversidade de estágios de maturidade em desenvolvimento de software em que estas organizações podem estar. O controle das atividades desempenhadas pelas organizações deve permitir à corporação identificar e prover mecanismos necessários para que elas convirjam para possuírem maturidade equivalente na execução de suas atividades relacionadas a software. Para que isto ocorra, devem ser observados aspectos relacionados à definição, execução e melhoria de processos, gerência de ativos de processos, gerência de conhecimento e à infraestrutura de apoio à execução destas atividades.

Definição de Processos - Deve ser de interesse da Corporação a definição de regras e requisitos mínimos para a definição dos processos a serem utilizados pelas organizações que a compõem. Devido isso, pode haver algum grau de ingerência em relação a um conjunto mínimo de procedimentos ou atividades a serem executadas em cada organização sob determinado contexto. Como a diversidade de processos nas organizações pode ser grande, a corporação pode definir processos, com o nível de detalhe que achar adequado, que possam ser incorporados nos diversos processos definidos nas organizações. Por outro lado a corporação pode ser a responsável por uma definição mais completa dos processos a serem executados em cada organização, limitando a estas apenas adaptações simples para deixá-los mais aderentes à sua cultura organizacional específica. Uma Organização que esteja subordinada a uma Corporação deve respeitar os limites estabelecidos para a definição de seu conjunto de processos-padrão. Além disso, deve garantir que todos os procedimentos e atividades estabelecidos tenham sido adequadamente incorporados a estes processos. Regras para a adaptação dos processos-padrão para uso nos projetos devem ser desenvolvidas pela organização em concordância às regras e limitações impostas pela corporação.

Em Ambientes Orientados a Corporação é possível a definição de processos padrão pela Corporação e sua posterior adaptação por parte das Organizações que as compõem para definir seus próprios processos padrão ou quando for necessário executar tais processos. O escopo e abrangência das adaptações que podem ser realizadas devem ser definidos pela Corporação de acordo com seus objetivos e necessidades.

Execução de Processos - A execução de processos em corporações não acontece apenas no contexto da produção efetiva de software. Processos podem ser definidos para um grande escopo de atividades relacionadas a software, por exemplo, atendendo a necessidades relativas à existência de programas de melhoria de processos, programa de medição ou um escritório de projetos. Estes processos, geralmente gerenciais, também devem poder ser executados e, para tal, necessitam de apoio ferramental adequado.

Dessa forma, os Ambientes Orientados a Corporação utilizam Ambientes de Engenharia de Software de forma a possibilitar a execução de processos de software seja no contexto Corporativo, seja no contexto Organizacional.

Melhoria de Processos - Uma Corporação, devido à maior diversidade de processos e à necessidade de lidar com organizações diferentes, com diferentes versões de processos e com diferentes níveis de maturidade, precisa ter maior controle sobre o ciclo de evolução de seus ativos de processos e, principalmente, de seus processos-padrão. Uma corporação deve ser capaz de identificar melhorias nos processos com base nas informações coletadas a partir da utilização dos processos nas diversas organizações. Além disso, a corporação deve ser capaz de disponibilizar novas versões de processos (e ativos de processo) às suas organizações, influenciando, dessa forma, iniciativas de melhoria de processos em andamento nas organizações ou forçando, na prática, o início de uma melhoria de processos nas organizações. Apesar de uma Organização subordinada a uma Corporação poder desenvolver suas próprias estratégias para melhoria de processos de software (como uma organização independente o faria) deve sempre respeitar as regras definidas pela corporação. Durante a análise das oportunidades de melhorias pela corporação é importante a coleta de dados e

informações gerados a partir da execução dos processos nas organizações, assim como do *feedback* fornecido por aqueles que executaram os processos ou membros das equipes responsáveis por analisar e avaliar as oportunidades de melhoria nas organizações ou do acesso a dados presentes em base de dados externas que sejam consideradas úteis. Dessa forma, é necessário que a corporação possa ter acesso de forma eficiente a estes tipos de dados para auxiliar na melhoria de seus processos. A utilização de um vocabulário comum relacionado a processos de software na corporação também auxiliaria a diminuir o viés causado por falhas na comunicação e entendimento dos conceitos relacionados à execução dos processos e, dessa forma, facilitaria a identificação de melhorias no contexto corporativo.

Os Ambientes Orientados a Corporação possibilitam que a Corporação identifique melhorias nos processos de software a partir de informações relacionadas ao uso dos processos por suas organizações e, também, possibilitam que a Corporação defina melhorias que devem obrigatoriamente ser implantadas nas organizações. Além disso, possuem mecanismos de acesso e integração de dados que permitem à corporação consultar os repositórios de dados das organizações e dos projetos para possibilitar a análise das informações disponibilizadas em cada um destes contextos visando a análise da institucionalização e da melhoria dos processos em uso.

Gerência dos Ativos de Processos - Do mesmo modo que a diversidade de processos no contexto corporativo é maior do que em uma organização isolada, a quantidade de ativos de processo relacionados a estes processos também é maior e estes devem ser gerenciados e controlados. Apesar de os ativos de processo nas organizações que compõem a corporação poderem variar enormemente devido às características e particularidades destas organizações (por exemplo, organizações com níveis de maturidade diferentes ou atuando em nichos de mercado diferentes), o alinhamento às diretrizes corporativas deve ser respeitado. Dessa forma, assim como acontece na definição de processos pelas organizações, a gerência dos ativos de processo nas organizações também pode sofrer influência das regras e necessidades da corporação.

Assim, uma Corporação deve garantir que a criação e a evolução dos ativos de processo por parte de suas organizações sejam realizadas seguindo as diretrizes gerais definidas para tal. Pode, também, prover ativos de processo às organizações sem permitir que estes sejam alterados. De qualquer forma, níveis de mudanças permitidas devem ser estabelecidos de forma a garantir que evoluções de um ativo não o tornem incompatível com os objetivos e necessidades corporativos. Uma Organização subordinada a uma Corporação deve respeitar as diretrizes definidas pela corporação para a criação e evolução dos seus ativos de processo. Os Ambientes Orientados a Corporação possuem uma biblioteca de ativos de processos que garanta à corporação a gerência dos seus ativos de processos e, além disso, permita à corporação um maior controle sobre a gerência e evolução do conteúdo das bibliotecas de ativos de processos de suas organizações subordinadas. O uso desta biblioteca deve permitir, também, às corporações analisar os dados obtidos pela organização na utilização de tais ativos de forma a contribuir para a melhoria dos ativos corporativos e, possivelmente, a partir disso, uma melhoria dos ativos em todas as organizações.

Gerência de Conhecimento - Uma Corporação deve possuir mecanismos eficientes para que as suas organizações compartilhem conhecimento entre si e contribuam para a base de conhecimento corporativa. Mecanismos eficientes para a disseminação de conhecimento devem ser instituídos para potencializar o uso e abrangência da base de conhecimento corporativo. Assim, além de construir sua própria base de conhecimento, tipicamente com conteúdo adquirido durante a execução de seus processos, uma Organização subordinada a uma Corporação pode se beneficiar do conhecimento corporativo consolidado na base corporativa ou presente apenas em bases de outras organizações.

Dessa forma, em Ambientes Orientados a Corporação é importante garantir que o

conhecimento acumulado pelas organizações durante a execução de seus processos possa ser analisado pela corporação e incorporado à sua base de conhecimento. Também deve ser possível para a corporação disponibilizar novos itens de conhecimento a organizações que possam vir a se beneficiar deles e utilizar estes itens de conhecimento durante a identificação de oportunidades de melhorias nos processos.

4. Definição e Criação de Ambientes de Engenharia de Software Orientados a Corporação na Estação Taba

A infra-estrutura provida pelos Ambientes de Engenharia de Software possibilita apenas a execução de processos de software específicos. Entretanto, a gerência das atividades relacionadas a software nas corporações envolve mais do que apenas executar processos específicos e, portanto, requer outro tipo de infra-estrutura de apoio. Os Ambientes Orientados a Organização, conforme definidos por VILLELA [2004], por sua vez, apóiam a gerência das atividades de organizações que desenvolvem software. Além disso, possibilitam a geração de ambientes de apoio à execução de processos de desenvolvimento e manutenção. Estes ambientes são adequados a organizações independentes (ou seja, não subordinadas a uma corporação) e estão em uso em várias organizações apoiando a implantação e melhoria de processos de software com MPS.BR. e CMMI. Entretanto, não são adequados a corporações e suas organizações. Dentre suas limitações está a não existência de mecanismos que permitam a uma organização evoluir de forma eficiente seus processos sem a configuração um novo ambiente organizacional com novas versões dos processos. Além disso, por não levarem em consideração a existência da organização num contexto corporativo, não consideram a existência de organizações com diferentes níveis de maturidade no desenvolvimento de software e a necessidade de se prover mecanismos que permitam a evolução destas organizações visando ao ganho de maturidade, possivelmente, para um nível único definido pela corporação.

Para ter-se os Ambientes Orientados a Corporação foi necessário prover uma infra-estrutura que permitisse a contemplação de todos os itens identificados anteriormente e também a monitoração das organizações para assegurar que as diretrizes corporativas relativas à definição, execução e melhoria dos processos estejam sendo seguidas. Além disso, esta infra-estrutura deve estar adequada à gerência da diversidade de processos de software existente no âmbito de uma corporação e também à gerência dos diferentes níveis de maturidade das organizações que a compõem. Os Ambientes Orientados a Corporação também devem ser capazes de prover o apoio ferramental necessário para a gerência das atividades das organizações e para a execução dos diferentes processos existentes no contexto corporativo e organizacional.

Com base nos cenários de evolução dos Ambientes de *Desenvolvimento* de Software para Ambientes de *Engenharia* de Software e de Ambientes Orientados a *Organização* para Ambientes Orientados a *Corporação* foram definidos os *Ambientes de Engenharia de Software Orientados a Corporação* cujos objetivos podem ser sumarizados como: (i) apoiar os engenheiros de software na execução de suas atividades; (ii) apoiar a definição, execução e melhoria dos diferentes processos de engenharia de software das organizações segundo suas características específicas e as diretrizes definidas pela corporação; (iii) apoiar a corporação na gerência da diversidade de processos e de níveis de maturidade possuídos pelas organizações; (iv) apoiar a gerência de conhecimento nas corporações, incluindo o aprendizado corporativo em Engenharia de Software a partir do aprendizado adquirido pelas organizações nos projetos de software; e (v) prover uma infra-estrutura que permita a gerência de configuração dos ativos de processo da corporação visando à adoção de abordagens de melhorias de processos.

Assim, podemos definir os Ambientes de Engenharia de Software Orientados a Corporação como o conjunto, ou família, de ambientes que possibilitam a gerência, execução, monitoração e melhoria de processos de uma corporação e das organizações que compõem esta corporação, além de possibilitar a evolução controlada dos ativos de

processos envolvidos nestas iniciativas. O modelo para construção de Ambientes de Engenharia de Software Orientados a Corporação (AESCorp) foi dividido em definição e geração de ambientes, definição de processos, melhoria de processos, execução de processos, gerência de ativos de processo e gerência de conhecimento.

Definição e Geração de Ambientes - Para compor a infra-estrutura relacionada aos AESCorp foi necessária a definição de quatro tipos diferentes de ambientes, conforme pode ser visto na Tabela 1.

Tabela 1. Ambientes

Ambiente	Descrição
Meta-Ambiente	O Meta-Ambiente é um ambiente capaz de gerar outros ambientes. É utilizado em uma entidade externa à corporação ou às organizações, sendo utilizado por profissionais especializados em engenharia de software, que fazem a manutenção e evolução dos ativos de processos e demais recursos disponíveis neste ambiente. Sua principal função é apoiar a configuração de ambientes para corporações ou organizações específicas e, também, prover as funcionalidades e controlar as características e recursos disponíveis nos demais ambientes. Dessa forma, a principal funcionalidade deste ambiente é configurar os Ambientes Corporativos (no caso de configuração para Corporações) e os Ambientes Organizacionais (no caso de configuração para Organizações Independentes).
Ambiente e Corporativo	Este ambiente é configurado no Meta-Ambiente a partir do processo-padrão corporativo. Sua função é auxiliar os engenheiros de software na gerência das atividades relacionadas aos processos de software existentes na Corporação. A partir deste ambiente pode ser feita a disponibilização de apoio ferramental e ativos de processo para as Organizações que compõem a Corporação através dos Ambientes Organizacionais.
Ambiente Organizacional	Este ambiente é configurado a partir do Ambiente Corporativo e é utilizado por profissionais de nível gerencial e pelo grupo de processos da organização responsáveis pela construção e disponibilização do apoio (computacional ou relativo a ativos de processo) necessário para controlar todas as atividades realizadas pela Organização relacionadas aos seus processos de software. Para organizações que não estejam subordinadas a uma Corporação, estes ambientes são configurados no Meta-Ambiente.
Ambiente de Projeto	Ambientes de Projeto são ambientes centrados em processo e caracterizam-se pela existência de um processo a ser executado, independentemente do seu tipo ou finalidade, e são gerados com o objetivo de executar e controlar as atividades de um projeto específico. Estes ambientes são utilizados pelo gerente/líder do projeto e demais membros da equipe do projeto, dando apoio às várias atividades dos processos do ciclo de vida de um software através de ferramentas específicas, descrição de procedimentos e <i>templates</i> , dentre outros. Apesar de o senso mais comum relacionado a um Ambiente de Projeto se referir a um Ambiente de Desenvolvimento de Software, a utilização de processos para desenvolvimento (ou manutenção) de software não é a única possibilidade para tais ambientes. Um novo ambiente pode ser gerado, por exemplo, para controlar as atividades relacionadas a Garantia da Qualidade em uma organização. A necessidade de execução de processos não é exclusiva dos Ambientes Organizacionais. Uma Corporação pode, por exemplo, necessitar de apoio para executar um processo para a gerência do seu programa de melhoria a partir do Ambiente Corporativo. A definição de um Ambiente de Projeto está, portanto, subordinada às regras e processos-padrão definidos no ambiente do qual é originado.

A hierarquia dos ambientes começa com o Meta-Ambiente, capaz de gerar os Ambientes Corporativos, isto é, ambientes adequados às necessidades de diferentes corporações. Estes Ambientes Corporativos são gerados a partir do processo-padrão corporativo. De posse de seu Ambiente Corporativo, uma Corporação pode configurar Ambientes Organizacionais para cada uma de suas organizações ou unidades organizacionais. Os Ambientes Organizacionais são instâncias dos Ambientes Corporativos e são configurados customizando-se o processo-padrão corporativo para a organização. Da mesma forma, cada organização pode instanciar Ambientes de Projeto através da adaptação de um dos processos-padrão da organização para a execução em um projeto específico. De posse do seu Ambiente, cada equipe de projeto pode, então, executar o processo definido para o projeto presente neste ambiente.

Definição de Processos nos AESCorp - Um item chave na geração dos diferentes

tipos de ambientes é a escolha dos processos a serem disponibilizados em cada um deles. De forma geral, cada ambiente contém um conjunto de processos que é adaptado para definir os processos do ambiente do próximo nível. Esta adaptação segue regras específicas próprias de cada nível de ambiente e das características de cada processo.

A abordagem para definição de processos apresentada, proposta inicialmente por OLIVEIRA [1999] e adaptada por VILLELA [2004], pode ser vista na Figura 1. Segundo a proposta do modelo, que estabelece etapas e produtos intermediários, a norma ISO/IEC 12207 [ISO/IEC-12207 1998] [ISO/IEC-12207:Amd1 2002] [ISO/IEC-12207:Amd2 2004] é a base para a definição de qualquer processo-padrão. Para a definição do processo-padrão, são, ainda, consideradas as características do desenvolvimento de software na organização, que são relacionadas ao ambiente de trabalho, conhecimento e experiência das equipes envolvidas e à própria cultura e experiência da organização no desenvolvimento de software. Também podem ser considerados modelos de maturidade como o MR-MPS [SOFTEX 2007] e o CMMI [Chrissis *et al.* 2006].

Antes do início da definição do processo-padrão é preciso identificar as características desejadas para ele, por exemplo, que atividades, ou grupo de atividades, são ou não obrigatórias no contexto corporativo ou organizacional, atividades necessárias para aderência a um modelo de maturidade, atividades comuns a um determinado tipo de organização (por exemplo, organizações que fazem pesquisa ou que são fábricas de software). Também devem ser levantados os itens de conhecimento de especialistas, externos ou internos à Organização/Corporação, que podem influenciar na definição dos processos (por exemplo, lições aprendidas e melhores práticas aplicáveis ao contexto corporativo). De posse destas informações, é possível definir um processo padrão mais genérico para a corporação a ser utilizado por todas as organizações, que pode ou não conter os elementos básicos definidos anteriormente. A definição do processo padrão deve levar em consideração também os diferentes processos de ciclo de vida existentes, normas e modelos de maturidade de software e características da corporação/organização. Esta etapa é realizada, inicialmente, no Meta-Ambiente e deve poder ser modificada no Ambiente Corporativo, caso necessário.

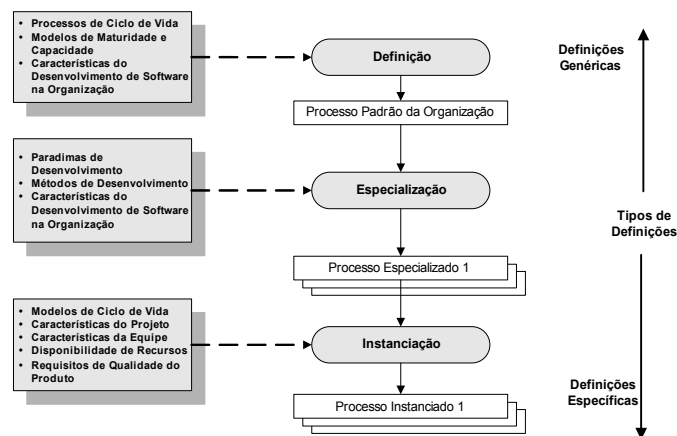


Figura 1 – Abordagem para Definição de Processos [Villela 2004]

A definição dos processos padrões para cada uma das organizações que compõem a corporação deve levar em consideração os diferentes ativos de processos existentes na organização, normas e modelos de maturidade e capacidade, características da organização. Estes processos padrões devem respeitar as características obrigatórias definidas pela Corporação e podem conter outros elementos não contemplados anteriormente e que sejam considerados adequados ao novo contexto. Esta etapa da definição dos processos é realizada, inicialmente, também no Meta-Ambiente e pode ser modificada no Ambiente Corporativo ou no Ambiente Organizacional, caso necessário.

Cada processo padrão pode dar origem a vários processos especializados de acordo com os diferentes paradigmas e métodos de desenvolvimento utilizados comumente pela organização, critérios definidos pela Organização para caracterização de projetos (por exemplo, tamanho de projeto, tipo de cliente, ramo de negócio relacionado ao projeto) e características do desenvolvimento de software na organização, como os tipos de software desenvolvidos na organização (por exemplo: sistemas especialistas, sistemas de informação e sistemas de controle de processos) e dos paradigmas de desenvolvimento adotados (por exemplo: orientado a objetos e estruturado). Neste momento, novas atividades podem ser definidas e incluídas nos processos especializados e a descrição de atividades já definidas no processo-padrão pode ser adequada. Estes processos especializados, no entanto, devem respeitar as características obrigatórias definidas pela Corporação e pela Organização e podem, também, conter outros elementos não contemplados anteriormente e que sejam considerados adequados ao novo contexto. É importante observar que um processo especializado também se comporta como um processo padrão e pode ser utilizado como tal (por exemplo, como base para uma nova especialização), pois possui apenas um novo refinamento para situações específicas.

Para ser utilizado em um projeto, o processo especializado mais adequado para um determinado tipo de software deve ser instanciado para atender às características do projeto específico e dar origem ao processo definido para o projeto. Cada processo especializado pode dar origem a vários processos instanciados de acordo com os diferentes modelos de ciclo de vida existentes e em uso pelas organizações, características específicas do projeto e da equipe, disponibilidade de recursos e requisitos de qualidade do produto ou do processo. Estes processos instanciados são os processos executados de fato e podem ser gerados tanto nos Ambientes Corporativos (geralmente associados com processos gerenciais no contexto corporativo, como o processo de gestão do programa de melhoria de processos da corporação) quanto nos Ambientes Organizacionais (neste caso, tanto para processos gerenciais, como a execução de um programa de medição, quanto para os relacionados à construção de software, como os processos de desenvolvimento ou manutenção). A definição dos processos instanciados deve, portanto, ser realizada antes do início de um projeto e deve respeitar as características obrigatórias definidas pela Corporação e pela Organização, além de poder conter adaptações pertinentes ao contexto dos projetos. Devem ser considerados neste momento, portanto, o tamanho e a complexidade do produto bem como as características de qualidade desejadas, a expectativa de vida útil, as características da equipe de desenvolvimento e demais características do projeto. Neste momento, também devem ser selecionados o modelo de ciclo de vida, os métodos e ferramentas, além de atividades relacionadas ao conhecimento de domínio específico a ser utilizado nos projetos.

Melhoria de Processos - A etapa de definição não é o final do ciclo de vida de um processo. Após a sua definição o processo deve ser executado e, a partir dos dados coletados sobre essa execução, deve ser melhorado. A estratégia para melhoria de processos deve estar adequada aos objetivos organizacionais e corporativos e ao cenário de melhoria de processos adequado à estrutura de definição de processos em uso. Dessa forma, os AESCorp não apóiam apenas a definição e execução dos processos, mas, também, a contínua melhoria destes processos. Dois cenários de melhoria são tratados: (i) a Corporação identifica uma melhoria num dos processos em uso pelas Organizações e disponibiliza para uso uma nova versão de processo padrão corporativo que deve, então, ser utilizada nos novos projetos pelas Organizações a partir deste momento; e (ii) a Organização decide aplicar uma melhoria em processos ainda em execução em projetos para resolver questões pertinentes mais rapidamente ou reduzir o número de versões diferentes dos processos em execução.

Execução de Processos - A execução dos processos é realizada apenas em Ambientes de Projetos, pois são os únicos de fato centrados em processo. Nos AESCorp, os

Ambientes de Projeto possibilitam o acesso a um conjunto de ferramentas, internas e/ou externas, adequadas ao propósito do ambiente e ao apoio necessário para a execução do processo para o qual o ambiente foi gerado. Os Ambientes de Projeto, também, possuem uma base de dados onde informações sobre a execução do processo e de suas ferramentas são armazenadas. Além disso, podem acessar elementos externos como outras ferramentas, ambientes e bases de dados para dar apoio à execução das atividades previstas. O conjunto de ferramentas e Ambientes de Projeto disponibilizados contemplam as fases de planejamento, apoio à execução e finalização do processo. Além disso, fornecem mecanismos para que ferramentas adequadas à execução de tarefas relacionadas a um processo específico (por exemplo, geração de informações sobre rastreabilidade entre requisitos) sejam definidas, criadas ou acessadas externamente.

Gerência de Ativos de Processo - Uma questão referente a melhoria de processos de software é como garantir um controle adequado sobre as diferentes versões dos processos e como prover algum mecanismo eficiente para garantir a rastreabilidade das mudanças ocorridas nestes processos. A identificação das mudanças ocorridas no processo, e dos motivos que levaram a estas mudanças pode ser útil, por exemplo, para que a corporação mantenha o controle sobre a evolução dos processos e também para identificar novas oportunidades de melhoria nos processos corporativos. Os AESCorp possuem uma biblioteca de ativos de processo responsável pela gerência e controle destes ativos. Parte deste controle está relacionada a mecanismos relacionados à gerência de configuração. Os AESCorp fornecem mecanismos para o controle de versão, controle de modificação e rastreabilidade de versões e mudanças, além de possuir bases independentes de ativos de processo e de versões de ativos de processo.

Gerência de Conhecimento - Conhecimentos adquiridos nas organizações devem ser armazenados no repositório de conhecimento organizacional e devem ficar associados a informações de contexto de forma a ser possível avaliar sua adequação à realidade dos projetos, das organizações e da corporação ao longo do tempo, além de permitir que possam ser reavaliados e reclassificados. Uma vez que a base de conhecimento, pelo menos em sua versão inicial, tiver sido construída, deve-se garantir que os itens de conhecimento sejam prontamente compartilhados e disponibilizados aos membros da corporação e das organizações e dos executantes dos processos. Possuir mecanismos que permitam incorporar conhecimentos e experiências registrados nos demais ambientes como, por exemplo, evoluções em lições aprendidas e novas melhores práticas. Outra importante fonte de conhecimento dentro da estrutura corporativa são os especialistas em áreas específicas, como, por exemplo, no próprio domínio de processos de software. Dessa forma, deve-se garantir que haja a identificação de tais especialistas e que eles possam ser encontrados e contatados visando a uma maior disseminação do conhecimento.

5. Alterações na Arquitetura da Estação Taba para a criação dos AESCorp

A evolução tecnológica e as mudanças econômicas ocorridas nos últimos anos vêm alterando significativamente as características das organizações e, conseqüentemente, dos projetos de desenvolvimento de software. Observa-se, por exemplo, uma tendência de descentralização das equipes dos projetos, com seus integrantes trabalhando distribuídos em localidades distintas. Além disso, numa corporação, é comum que as organizações estejam em localidades diferentes e, também neste contexto, aumenta a necessidade de acesso aos dados e às ferramentas entre a corporação e as organizações e mesmo dentre as próprias organizações de uma mesma corporação. Esse cenário impõe alterações na forma de gerenciar projetos de software para garantir o seu sucesso.

A arquitetura da Estação TABA, originalmente, não foi concebida para apoiar a descentralização do uso das ferramentas, ambientes e bases de dados. Sendo assim, foi necessária uma reformulação dessa arquitetura com o objetivo de suprir tal

deficiência e, assim, atender às novas expectativas. Foi realizada, então, uma nova implementação da Estação Taba na web utilizando-se a tecnologia JEE.

Com a nova implementação, foram revistos os seus componentes e a filosofia de integração foi adaptada à nova realidade. Há um conjunto de serviços básicos que fazem parte da infra-estrutura mínima dos ambientes. Estes serviços são responsáveis por prover funcionalidades como integração de ferramentas internas e externas, padronização da interface com usuários e prover acesso a diferentes fontes de dados, como bases de dados e conhecimento internas e externas. Estes serviços são responsáveis pela implementação da estratégia de integração adotada e são compartilhados por todas as ferramentas, componentes e ambientes da Estação Taba.

6. Conclusão

Este artigo apresentou os *Ambientes de Engenharia de Software Orientados a Corporações* (AESOrg) e os seus principais requisitos e características de infra-estrutura, englobando apoio à definição, execução e melhoria de processos, definição de ambientes, gerência de ativos de processos e gerência de conhecimento. Foram realizadas alterações na Estação Taba para permitir a construção de tais ambientes, entre elas o TABA Web.

Neste momento, está sendo realizada a configuração de um AESCorp para uma corporação, onde uma das organizações possui o Nível E de maturidade MPS e outra de suas organizações vai iniciar a implantação do Nível G. Com os resultados desta experiência será possível avaliar o apoio ferramental provido pelos Ambientes de Engenharia de Software Orientados a Corporação.

Referências

- Amescua, A., Garci?a, J., Sa?nchez-Segura, M.I., et al. (2006), "Approaching software process improvement to organizations", *WSEAS Transactions on Computers*, v. 5, n. 3, pp. 507-514.
- Canfora, G., Garci?a, F., Piattini, M., et al. (2006), "Applying a framework for the improvement of software process maturity", *Software - Practice and Experience*, v. 36, n. 3, pp. 283-304.
- Chrissis, M.B., Konrad, M., Shrum, S. (2006), *CMMI (Second Edition): Guidelines for Process Integration and Product Improvement*, Addison Wesley Professional.
- Costa, A., Sales, E., Reis, C.A.L., et al. (2007), "Apoio a Reutilização de Processos de Software através de Templates e Versões". In: *Simpósio Brasileiro de Qualidade de Software - SBQS 2007*, pp. 47-61, Porto de Galinhas - PE, Junho 2007.
- Ferreira, A.I.F., Santos, G., Cerqueira, R., et al. (2007), "Applying ISO 9001:2000, MPS.BR and CMMI to Achieve Software Process Maturity: BL Informatica's Pathway". In: *29th Int. Conference on Software Engineering (ICSE)*, pp. 642-651, Minneapolis, USA, May.
- Ferreira, A.I.F., Santos, G., Cerqueira, R., et al. (2006), "Taba workstation: Supporting software process improvement initiatives based on software standards and maturity models". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 4257 LNCS, pp. 207-218, Joensuu.
- ImPProS, "Ambiente de Implementação Progressiva de Processo de Software". In: <http://www.cin.ufpe.br/~imppros/>.
- ISO/IEC-12207 (1998), "Tecnologia de Informação - Processos de ciclo de vida de Software", ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, v. ISO/IEC-12207.

- ISO/IEC-12207:Amd1 (2002), "Information Technology - Amendment 1 to ISO/IEC 12207", *The International Organization for Standardization and the International Electrotechnical Commission*, v. ISO/IEC-12207:Amd1.
- ISO/IEC-12207:Amd2 (2004), "Information Technology - Amendment 2 to ISO/IEC 12207", *The International Organization for Standardization and the International Electrotechnical Commission*, v. ISO/IEC-12207:Amd2.
- Jun, D., Rui, L., Yi-min, H. (2007), "Software Processes Improvement and Specifications for Embedded Systems". In: *Software Engineering Research, Management & Applications, 2007. SERA 2007. 5th ACIS International Conference on*, pp. 13-18.
- Montoni, M., Santos, G., Rocha, A.R., et al. (2007), "MPS Model and TABA Workstation: Implementing Software Process Improvement Initiatives in Small Settings". In: *Fifth Workshop on Software Quality held in conjunction with the 29th Int. Conference on Software Engineering (ICSE)*, Minneapolis, USA, May.
- Niazi, M., Wilson, D., Zowghi, D. (2005), "A framework for assisting the design of effective software process improvement implementation strategies", *Journal of Systems and Software*, v. 78, n. 2, pp. 204-222.
- Oliveira, K.M. (1999), *Modelo para Construção de Ambientes de Desenvolvimento de Software Orientados a Domínio*, Tese de D. Sc., COPPE, UFRJ, Rio de Janeiro, Brasil.
- Oliveira, S.R.B., Vasconcelos, A.M.L. (2006), "A Continuous Improvement Model in ImPProS". In: *Computer Software and Applications Conference, 2006. COMPSAC '06. 30th Annual International*, v. 2, pp. 370-371.
- Paxiúba, C., Pereira, M., Reis, C.A.L., et al. (2007), "Acompanhamento e Avaliação de Projetos através da Monitoração de Eventos em um Ambiente de Gestão de Processos de Software." In: *Simpósio Brasileiro de Qualidade de Software - SBQS 2007*, pp. 15-29, Porto de Galinhas - PE, Junho 2007.
- Santos, G., Montoni, M., Figueiredo, S., et al. (2007), "SPI-KM - Lessons Learned from Applying a Software Process Improvement Strategy Supported by Knowledge Management", *Product-Focused Software Process Improvement*.
- SOFTEX (2007), "MPS.BR – Guia Geral".
- Villela, K. (2004), *Definição e Construção de Ambientes de Desenvolvimento de Software Orientados à Organização*, Tese de D. Sc., COPPE, UFRJ, Rio de Janeiro, Brasil.
- Wang, Q., Li, M. (2005), "Measuring and improving software process in China". In: *2005 International Symposium on Empirical Software Engineering, ISESE 2005*, pp. 183-192, Queensland.

[6.22] Um Framework de Engenharia de Requisitos para Desenvolvimento de Produtos de Software

Entidade: Centro de Informática - Universidade Federal de Pernambuco (UFPE) - Caixa Postal 50732-970 - Recife - PE - Brazil

Autor: Carina Alves - {cfa}@cin.ufpe.br

Resumo. *Este artigo apresenta as experiências e resultados obtidos durante o programa de melhoria do processo de engenharia de requisitos junto a empresas de produtos de software.*

1. Introdução

Atualmente, o desenvolvimento de produtos de software representa um importante e crescente segmento da indústria de software. Produtos de software também são conhecidos pacotes de prateleira ou produtos COTS (Commercial Off The Shelf). Empresas que desenvolvem produtos de software visando um amplo mercado consumidor, geralmente com clientes distribuídos em várias localidades, sofrem dificuldades ainda maiores para atender satisfatoriamente clientes com diferentes realidades organizacionais e necessidades específicas. Diante da crescente demanda por produtos de qualidade e da crescente exigência dos mercados consumidores, empresas de produtos de software têm reconhecido a importância de programas de melhoria do processo de desenvolvimento de software.

De acordo com pesquisas do Gartner Group, aproximadamente 60-70% das falhas ocorridas em projetos de desenvolvimento de sistemas de software são devidas à inadequação do processo de aquisição, análise e gerenciamento de requisitos. Em geral, qualquer empresa desenvolvedora de sistemas de software necessita entender da melhor forma possível quais são os requisitos de seus clientes. Um dos principais objetivos do processo de engenharia de requisitos envolve o entendimento das reais necessidades e metas de clientes e usuários.

2. Objetivos e Justificativa

O objetivo do projeto foi estimular e apoiar a troca de conhecimento na área engenharia de requisitos entre universidade e empresas do setor de TI. Em particular, este projeto de pesquisa realizou uma pesquisa empírica sobre práticas e processos de engenharia de requisitos conduzidos por empresas desenvolvedoras de produtos de software. O projeto foi financiado pelo CNPQ e obteve apoio local do Softex-Recife e do Centro de Informática da Universidade Federal de Pernambuco.

O projeto de pesquisa foi guiado para investigar os seguintes aspectos:

- Examinar como empresas desenvolvedoras de produtos de software conduzem o processo de engenharia de requisitos;
- Entender os principais desafios e problemas enfrentados por empresas de produtos de software baseadas em Recife, Pernambuco;
- Investigar quais práticas de requisitos são atualmente adotadas por essas empresas;
- Conduzir um programa de melhoria do processo de engenharia de requisitos em quatro empresas de software baseadas em Recife, Pernambuco.

3. Metodologia de Execução

A pesquisa desenvolvida foi dividida em três fases, onde para cada fase foram estabelecidas metas, atividades e prazos para obtenção de resultados. A Figura 1 apresenta uma visão geral das fases do projeto, foram elas: Planejamento e Revisão da Literatura, Estudo empírico, Programa de Melhoria em Engenharia de Requisitos. A seguir, apresentamos cada uma das fases do projeto.

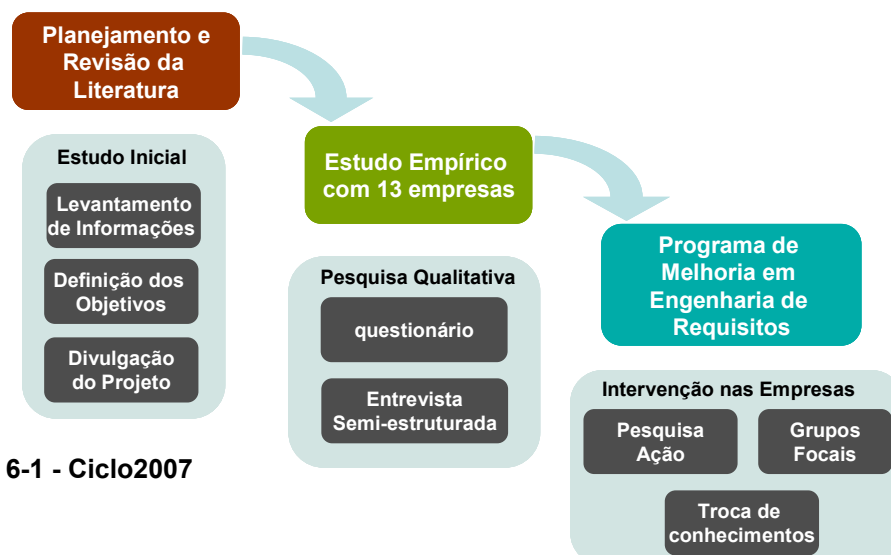


Figura 1 – Fases do Projeto de Pesquisa

Fase 1 - Planejamento e Revisão da Literatura

A pesquisa teve início com a análise bibliográfica da área e investigação de estudos semelhantes conduzidos por pesquisadores junto a empresas de software de outros países. O resultado desse estudo bibliográfico originou o levantamento de dez hipóteses relacionadas às características e práticas correntes no desenvolvimento de pacotes de software. A próxima atividade da pesquisa caracterizou-se pela identificação das empresas que iriam participar do estudo. Nesta etapa, contamos com o apoio do SOFTEX Recife para divulgar a pesquisa junto às empresas associadas. Assim, várias empresas foram convidadas via telefone e/ou e-mail para participar de um seminário sobre o projeto da pesquisa. Dentre as empresas participantes, treze delas concordaram em participar do estudo. Em paralelo, elaboramos um questionário a fim de obter informações gerais sobre as empresas e entender as principais características sobre os produtos oferecidos e o processo de desenvolvimento adotado pelas mesmas.

Fase 2 – Estudo Empírico

Na segunda fase do estudo realizamos um estudo empírico junto a treze empresas de produtos de software baseadas em Recife. O objetivo do estudo foi obter um melhor entendimento sobre o processo de engenharia de requisitos e levantar os principais desafios e problemas enfrentados por empresas desenvolvedoras de produtos de software para um amplo mercado consumidor. O estudo empírico foi conduzido segundo normas e premissas do método qualitativo. Este método de pesquisa é constituído de passos e técnicas que propiciam a compreensão, investigação e análise de questões “por quê” e “como” para melhor entender o objeto de estudo.

O estudo empírico foi dividido em duas etapas. A primeira etapa foi caracterizada pela aplicação do questionário junto a cada empresa participante. A segunda etapa foi caracterizada pela realização de entrevistas semi-estruturadas nas empresas participantes. Os entrevistados foram indicados pelo representante da empresa. Vale salientar que todas as entrevistas foram guiadas pelas mesmas perguntas básicas. Dependendo do andamento de cada entrevista, fizemos perguntas mais específicas para esclarecer pontos que consideramos relevantes.

As entrevistas foram realizadas por dois pesquisadores, sendo que um deles foi responsável por fazer as perguntas e o outro foi responsável por realizar anotações durante a entrevista assim como transcrever a entrevista. A equipe total foi composta por sete pesquisadores, onde a coordenadora do projeto conduziu todas as entrevistas. Após a realização das entrevistas, as mesmas foram transcritas e posteriormente revisadas por dois pesquisadores objetivando assegurar a exatidão e a uniformidade do processo. A análise das entrevistas foi realizada através de leitura e discussão das transcrições. Cada pesquisador ficou responsável por ler todas as entrevistas e sublinhar as seções por ele consideradas relevantes. Posteriormente, cinco reuniões foram realizadas com intuito de promover a discussão das informações encontradas e possibilitar a formulação de conclusões preliminares. A fim de auxiliar a atividade de análise e interpretação dos dados foi utilizada a ferramenta Nvivo, que permite a análise qualitativa dos dados obtidos, auxiliando na organização dos registros da pesquisa e das interpretações dos mesmos.

Os principais resultados do estudo incluem os seguintes aspectos:

- Os maiores desafios enfrentados pelas empresas estudadas durante o processo de desenvolvimento de software são cumprir prazo de entrega, entender os requisitos dos clientes e carência de mão de obra especializada;

- 58,3% das empresas afirmaram não possuírem um processo bem definido de engenharia de requisitos;
- Os problemas mais freqüentes relacionados ao processo de engenharia de requisitos são: falta de um processo definido de ER, dificuldade de entender as reais necessidades dos usuários, marketing deficiente, dificuldade de interação com clientes e dificuldade em gerenciar requisitos.

A partir dos resultados obtidos neste estudo empírico podemos concluir que, de um modo geral, as empresas compreendem a importância de conduzir um processo bem definido de requisitos para melhorar a qualidade final do software e melhor satisfazer seus clientes. No entanto, elas ainda enfrentam dificuldades em definir um processo de engenharia de requisitos específico e apropriado para sua realidade e também escolher métodos e ferramentas adequadas para suas necessidades. Dessa forma, acreditamos que iniciativas para introduzir e disseminar boas práticas em engenharia de requisitos podem representar um importante passo para melhorar a qualidade dos processos de empresas de produtos de software.

Ao final do estudo, os resultados foram apresentados em um evento organizado no Softex-Recife. A coordenadora do projeto apresentou uma palestra mostrando os principais resultados da pesquisa para a comunidade acadêmica e profissional assim como as empresas participantes do estudo. Nessa ocasião, foi entregue um relatório técnico para as empresas participantes.

Fase 3 – Programa de Melhoria em Engenharia de Requisitos

O Programa de Melhoria em Engenharia de Requisitos foi concebido para realizar transferência de conhecimento para as empresas participantes e que demonstraram interesse em melhorar seus processos de ER. O programa foi realizado com quatro empresas pré-selecionadas do grupo de treze inicialmente estudadas. O critério de seleção foi garantir o comprometimento dos colaboradores das empresas participantes e escolher empresas que já estivessem conduzindo projetos de melhoria da qualidade de seus processos. O programa de melhoria em ER foi conduzido durante o período de quatro meses. A Tabela 1 apresenta um perfil das quatro empresas participantes.

Empresa	Área de atuação	Nº Staff	Nº Staff em TI	Nº clientes
TCI	Business Process Outsourcing, Enterprise Content Management	300	60	100
MV	Gestão Hospitalar MV 2000	320	288	200
Procenge	ERP Pirâmide	79	44	162
Facilit	Gestão de conhecimento para Web Communis	30	25	15

Tabela 1 – Descrição das empresas participantes do Programa de Melhoria em ER

O programa envolveu a colaboração direta entre pesquisadores do projeto e empresas, caracterizando-se como uma ação de intervenção. Cada empresa selecionou um ou mais colaboradores para interagir com os pesquisadores. Este processo envolveu a condução de um projeto piloto para acompanhar a evolução do aprendizado e utilizar os novos conhecimentos em um projeto real de cada empresa. O programa foi dividido em nove iterações, cada iteração cobriu uma área específica da engenharia de requisitos. A

concepção do programa foi baseada em pesquisas já consolidadas, tais como [Sommerville 200] e [Robertson 2006].

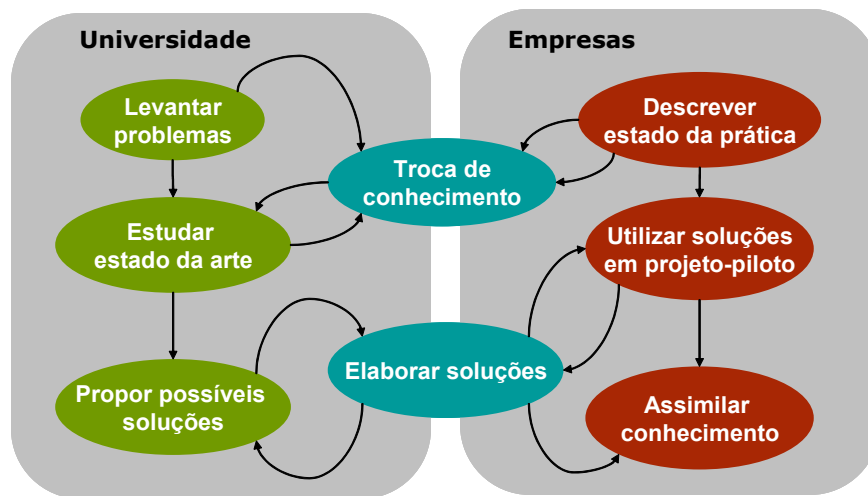


Figura 2 – Modelo de Transferência Tecnológica

O modelo de transferência tecnológica adotado foi inspirado em trabalhos de [Gorschek 2006]. De acordo com a Figura 2, o modelo inicia com a identificação das áreas de melhoria através de observação direta nas empresas e negociação com os participantes. Nosso objetivo foi enfocar nos pontos considerados desafios levantados no estudo anterior. Dessa forma, foram escolhidas as seguintes áreas: elicitación de requisitos, priorização de requisitos para plano de versão e especificação de requisitos. A etapa seguinte foi o estudo da literatura disponível sobre as áreas a serem investigadas. Neste momento também foi realizada troca de conhecimento entre as empresas para garantir que as áreas correspondem às suas reais necessidades de melhoria. Em seguida, as possíveis soluções foram apresentadas durante reuniões individuais com cada empresa.

A Figura 3 apresenta as nove iterações do programa. Estes encontros eram chamados *reunião da iteração*. Durante as reuniões, era apresentado o pacote da iteração, que consiste na descrição da iteração, incluindo os objetivos da iteração e áreas cobertas naquela iteração, além de *templates* para a empresa preencher durante a execução do projeto piloto e artigos relevantes sobre os tópicos estudados. Por exemplo, a primeira iteração cobriu a definição do escopo e metas do projeto, e identificação dos stakeholders. Em cada reunião da iteração, o objetivo dos pesquisadores era apresentar a área do processo de ER a ser investigada, discutir boas práticas para conduzir aquela atividade e estimular que os colaboradores de cada empresa conseguissem instanciar e aplicar tais conhecimentos diretamente no projeto piloto que eles estavam elaborando. Após cada reunião, os colaboradores recebiam a “tarefa de casa” que era estudar o assunto coberto naquela iteração e preencher os *templates* a fim de preparar os artefatos de requisitos do projeto piloto. Além dos encontros individuais com cada empresa, foram organizados três grupos focais com todas as empresas. O objetivo dos grupos focais era estimular a troca de conhecimento entre as empresas. Os grupos focais foram especialmente bem aceitos pelas empresas que viam naqueles encontros, a oportunidade de aprender com outras empresas e dividir aprendizados.

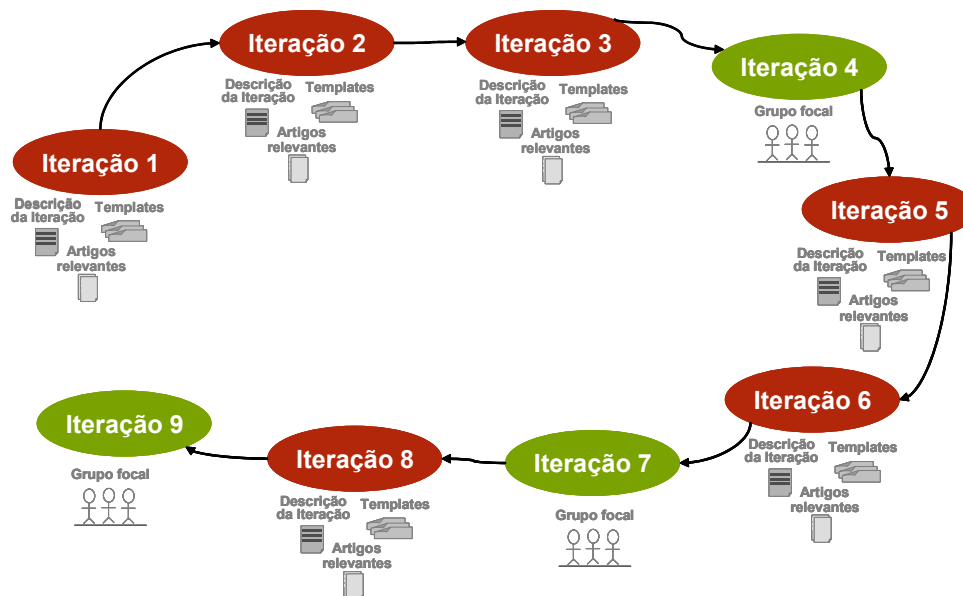


Figura 3 – Iterações do Programa de Melhoria em ER

4. Resultados Obtidos

4.1. Recursos Humanos Capacitados

O projeto de pesquisa teve a colaboração de três bolsistas de iniciação científica, que foram financiados pelo CNPQ, para participar do projeto. Além destes bolsistas, o projeto contou com a participação de dois bolsistas do PET que trabalharam no projeto como voluntários. Também tivemos a colaboração de uma estudante de mestrado que realizou sua dissertação no contexto do projeto.

Por outra perspectiva, as equipes das empresas participantes do programa de melhoria em requisitos também receberam conhecimentos sobre os avanços mais recentes na área de engenharia de requisitos e tem atuado como disseminadores de informação dentro de suas empresas.

4.2. Artigos Publicados

Ao longo do projeto de dois anos foram publicados diversos artigos. Em particular, os artigos [1] [2] e [3] foram escritos em parceria com colaboradores das empresas participantes do estudo. Isto demonstra a efetiva colaboração e obtenção de resultados entre a academia e empresas.

1. Heimann, V. Alves, C. (2008a) "Melhorando o Processo de Engenharia de Requisitos em Empresas de Produtos de Software - Um Estudo de Caso." 11 Iberoamerican Workshop on Requirements Engineering and Software Environments. Recife, Brasil.
2. Alves, C. Valença, G. Sotero, T. Mendes, J. (2008b) "Requirements Engineering Process Improvement: A Knowledge Transfer Experience." 23rd Annual ACM Symposium on Applied Computing. Fortaleza, Brasil.
3. Alves, C. Ramalho, G. Damasceno. (2007a) "Challenges in Requirements Engineering for Mobile Games Development: The Meantime Case Study." 15th IEEE International Requirements Engineering Conference, Delhi, India.
4. Alves, C. Pereira, S. Valença, G. Pimentel, J. Andrade, R. V. C. L. (2007b) "Preliminary Results from an Empirical Study in Market-Driven Software Companies". 10th Workshop of Requirements Engineering, Toronto, Canada.

5. Alves, C. (2007c) “Um Framework de Engenharia de Requisitos para Desenvolvimento de Produtos de Software.” Encontro de Produtividade e Qualidade em Software, Porto de Galinhas, Brasil.
6. Relatório Técnico - Um Estudo Empírico sobre Práticas de Engenharia de Requisitos junto a empresas de Pacotes de Software.

4.2. Dissertações e Trabalhos de Graduação Gerados

- Silvia Cássia Pereira – Dissertação de mestrado - Título: Um Estudo Empírico sobre Engenharia de Requisitos em Empresas de Produtos de Software. Centro de Informática, Universidade Federal de Pernambuco.
- João Pimentel – Trabalho de Graduação – Título: Adequação de um Processo de Reuso de Requisitos a uma Empresa Desenvolvedora de Software. Centro de Informática, Universidade Federal de Pernambuco.
- Daniel Penaforte – Trabalho de Graduação – Título: Uma Análise Sistêmica dos Problemas Enfrentados por Empresas de Software durante o Processo de Requisitos. Centro de Informática, Universidade Federal de Pernambuco.

5. Aplicabilidade dos Resultados

O principal objetivo do programa de melhoria do processo de engenharia de requisitos é a direta troca de conhecimento entre academia, através da disseminação de avanços obtidos na literatura acadêmica, e empresas de software, que enfrentam inúmeros desafios durante as fases iniciais de requisitos. Vale salientar que as quatro empresas participantes do programa de melhoria estão conduzindo programas como MPS-BR e CMMI. Isso demonstra o interesse destas empresas na melhoria de seus processos. As equipes de cada empresa participante têm liderado iniciativas para disseminar as boas práticas aprendidas durante o programa e auxiliar a implantação de novos programas de requisitos e melhorar o processo atual de suas empresas.

6. Características Inovadoras

Acreditamos que o programa é uma das primeiras iniciativas realizadas no País com foco na melhoria do processo de engenharia de requisitos. Apesar da grande importância e impacto que processos de requisitos têm em relação à qualidade de produtos de software, poucas empresas têm adotado práticas efetivas para conduzir esta fase de forma apropriada. Dessa forma, consideramos que este projeto investigou um tema de relevância para melhoria do estado da prática em requisitos.

7. Conclusão e Perspectivas Futuras

Os resultados já alcançados confirmam a relevância de realizar iniciativas como este projeto para estimular a transferência de conhecimento em áreas da engenharia de software. As empresas participantes demonstraram grande interesse em adotar boas práticas em engenharia de requisitos e iniciar programas específicos nesta área. Consideramos que o programa possa ser realizado junto a empresas de outras regiões do Brasil e assim, disseminar que novos projetos de pesquisa sejam realizados em parceria com empresas do setor.

8. Referências Bibliográficas

- Robertson, S. Robertson, J. (2006). “Mastering the Requirements Process, Addison Wesley, 2 edition.”
- Sommerville, I. Sawyer, P. (2000). “Requirements Engineering: A Good Practice Guide. Wiley, 3 edition.”
- Gorschek, T. Wohlin, C. Garre, P. Larsson, S. (2006). “A Model for Technology Transfer in Practice. IEEE Software.”