



AdEPT: Um Framework para Apoiar Práticas de Qualidade em Empresas de Software

Nadja da Nobrega Rodrigues¹, Jose Cazuza Pinheiro Neto²

¹Professora do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas - IFPB. e-mail: nadja.rodrigues@ifpb.edu.br

²Graduando do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas - IFPB. e-mail: cazuzant@gmail.com

Resumo: Processos de desenvolvimento de software e modelos de qualidade têm sido vistos como recursos essenciais para potencializar o sucesso de projetos de software. Entretanto, a incorporação de suas práticas muitas vezes é vista como um desafio, especialmente por empresas de pequeno porte, em virtude de diversos fatores, como exemplo, complexidade, altos custos ou mesmo falta de cultura das empresas para trabalhar com sistematização na geração de produtos de software. Visando potencializar o uso de processos e práticas de qualidade pelas empresas, este artigo retrata o AdEPT, um framework para apoiar práticas de qualidade em empresas de software. O principal objetivo do AdEPT é orientar tanto a adoção, evolução, prospecção e transição de processos de desenvolvimento de software, quanto o seu treinamento, em micro e pequenas empresas. Tecnicamente falando, o AdEPT se baseia no Modelo de Referência do MPS.BR, definindo níveis próprios de qualidade e seus respectivos elementos (processos, subprocessos, papéis, artefatos e ferramentas), o que o diferencia de vários modelos propostos, que não são tão detalhados em termos de definição desses elementos, ou se mostram pesados, em termos de esforços técnicos, para instanciação de projetos no contexto de micro e pequenas empresas. Através do AdEPT, espera-se que as empresas de software se aproximem das boas práticas da Engenharia Software, viabilizando projetos de qualidade, e promovendo-as no sentido de se desenvolverem e se projetarem na indústria de software.

Palavras-chave: Empresas, desenvolvimento de software, qualidade, boas práticas

1. INTRODUÇÃO

As Tecnologias de Informação e Comunicação vêm sendo utilizadas não apenas como ferramentas para automatização organizacional, através de sistemas de informação tradicionais [Rodrigues 2002]. Com o surgimento de novos paradigmas e recursos, houve uma transferência de valores entre os instrumentos organizacionais, e a tecnologia passou a ser vista como elemento imprescindível para elaboração de estratégias e formação de inteligência de negócios. Como consequência, houve uma mudança no perfil das aplicações de software, que passaram a apresentar características que representam maior complexidade para a sua construção. O resultado da percepção dessa constante evolução das tecnologias e do aumento de poder proporcionado por elas pode ser associado ao aumento no nível de exigência dos “clientes de tecnologia” [Rodrigues 2002].

A produção de software vem a ser uma atividade mais particular quando comparada as outras áreas de desenvolvimento de produtos. Os sistemas de software são abstratos e intangíveis. Eles não são restringidos pelas propriedades dos materiais, nem governados pelas leis da física ou pelos processos de manufatura [Sommerville 2011]. Devido a essa singularidade o desenvolvimento de sistemas pode tornar-se uma atividade muito difícil de ser gerenciada. Manter uma previsibilidade constante ao longo de um projeto de software é algo muito cobijado por os profissionais da área.

São muitos os fatores que podem fazer um projeto de software falhar, levando-o ao fracasso total ou a enfrentar problemas, gerando uma profunda insatisfação nos clientes. Kruchten (2003) explica que a maioria dos projetos falha por causa da combinação de causas como gerência ineficiente de requisitos, comunicação ambígua e imprecisa, inconsistências não detectadas, teste insuficiente, automação insuficiente, entre outros. Beck (2004) faz referência a alguns riscos que as empresas de desenvolvimento de sistemas enfrentam, como deslizos no cronograma, erros, negócio mal compreendido, modificações nos negócios, falsa riqueza de funções, entre outros.

É necessário que os profissionais de desenvolvimento de software criem a cultura de incorporar em seus projetos as boas práticas da Engenharia de Software (ES), que aplicadas de forma correta e inteligente, podem minimizar os problemas e fornecer um gerenciamento mais seguro dos riscos da



atividade. Software, em todas as suas formas e em todos os seus campos de aplicação, deve passar pelos processos de engenharia [Presmam 2011]. Torna-se indiscutível a importância de desenvolver software fazendo uso das práticas de ES, que podem contribuir para o êxito nos projetos. Para Sommerville (2011), ES está ligada a resultados de qualidade requeridos em cronograma e orçamento.

Assim como qualquer atividade produtiva, o desenvolvimento de software precisa de sistematização. Para este fim, surgiram os Processos de Desenvolvimento de Software (PDS), definidos como uma sequência de atividades que leva a um produto de software. Os processos auxiliam a força de trabalho da organização a alcançar seus objetivos estratégicos, ajudando-a a trabalhar de forma mais inteligente, com menor esforço e melhor consistência, permitindo a otimização de recursos e uma melhor compreensão das tendências de negócio [Software 2011]. Com a disseminação da cultura de processo, empresas e profissionais de Informática estão se conscientizando da importância de se definir, sistematizar e controlar o processo de software, compreendido como o conjunto de atividades que abrangem todo o ciclo de vida de um software [Associação 1998].

Empresas e profissionais de Informática estão se conscientizando de que atender ou superar as demandas e expectativas dos stakeholders com qualidade e assim chegar ao sucesso de um projeto são alguns dos desafios da área de ES e, especificamente, da aplicação de PDS, ao longo de todo o ciclo de vida de um software. Apesar de pesquisas mostrarem que a utilização de processos bem definidos traz benefícios importantes para uma empresa de desenvolvimento, há uma resistência por partes destas pequenas empresas em adotarem estes processos, uma vez que é uma tarefa onerosa [Zahran, 1998].

Os principais modelos de melhoria de processo, como o CMMI [Ahern et al. 2001] e a ISO 15504 [International 2004], exigem uma infra-estrutura considerável para que possam ser empregados. O custo de implantação desses modelos praticamente inviabiliza a sua pronta utilização em pequenas empresas, exigindo adaptações significativas para que possam ser adotados. O MPS.BR [Associação 2011] ou Melhoria de Processos do Software Brasileiro é simultaneamente um movimento para a melhoria da qualidade (Programa MPS.BR) e um modelo de qualidade de processo (Modelo MPS) voltada para a realidade do mercado de pequenas e médias empresas de desenvolvimento de software no Brasil. Ele é baseado nas normas ISO/IEC 12207 e ISO/IEC 15504 e na realidade do mercado brasileiro, bem como é compatível com o CMMI. Processos ágeis [Rising et al. 2000, Beck e Boehm 2003, Highsmith 2002], por outro lado têm se mostrado como uma alternativa econômica para implantação de PDS ao estabelecer um equilíbrio entre flexibilidade do processo e características inerentes a pequenas empresas de desenvolvimento, como exemplo, equipes pequenas.

A Paraíba já possui várias empresas trabalhando com desenvolvimento de sistemas, tanto públicas como privadas, e um volume representativo de pesquisas relacionadas a essa área. Entretanto, faz-se importante o direcionamento de ações no sentido de trabalhar o conhecimento gerado tanto na indústria como na academia, de forma democrática, buscando aproximá-lo do maior número de empresas, independentemente de seu porte e recursos disponíveis para investimento, incentivando o empreendedorismo e aumentando o alcance das iniciativas de inovação tecnológica no Estado.

Atualmente, na grande João Pessoa, existem muitas pequenas empresas de desenvolvimento de software [Farol Digital 2011], sendo que a maioria destas não possui nenhum processo que oriente a construção de seus produtos, os softwares. Pensando nas demais cidades da Paraíba, distantes dos grandes centros, onde a cultura de geração de software ainda está começando a se firmar, especialmente com a chegada de cursos ligados à área Análise e Desenvolvimento de Sistemas, a realização de atividades dessa natureza não deve ser diferente. Nesse cenário podem ser encontrados diversos tipos de problemas, como consequência dessa falta de sistematização. Dentre eles, podemos citar: improviso no desenvolvimento (que é feito ad hoc), forte dependência dos profissionais (pois apenas eles conhecem e entendem os sistemas que desenvolvem), pouca produtividade (ou não controlada), qualidade de difícil previsão, alto custo de manutenção e alto risco nos projetos.

Deve-se trabalhar de forma mais específica as demandas desse perfil de empresas, como forma de desenvolvê-las tanto em infraestrutura como em capital intelectual. Essa iniciativa se coloca ainda como uma estratégia tanto para garantir a qualidade dos produtos gerados pelas pequenas empresas e profissionais liberais da área de desenvolvimento de software, como para atender à necessidade, a nível de país, de crescimento nos indicadores de desenvolvimento e inovação tecnológica.



Neste sentido, este projeto tem o objetivo de especificar um framework para orientar tanto a adoção, quanto a evolução, prospecção e transição de práticas de PDS, quanto o seu treinamento, em pequenas empresas. O projeto pretende analisar as boas práticas da ES e da Qualidade de Software (QS), seus métodos e ferramentas, em relação aos PDS para que uma pequena empresa de desenvolvimento tenha apoio contínuo (evolutivo) na sistematização na construção dos seus produtos, aumentando a qualidade destes e a produtividade da equipe e, consequentemente, diminuindo o retrabalho, se tornando uma empresa mais competitiva no mercado.

2. MATERIAL E MÉTODOS

Tratou-se de uma pesquisa exploratória, envolvendo levantamento bibliográfico sobre modelos prescritivos e ágeis de PDS, além de práticas e modelos da área de qualidade de software, buscando investigar a ES e a QS, e identificar elementos que poderiam apoiar o desenvolvimento de software de qualidade em micro e pequenas empresas. Ainda foram investigados os principais problemas associados ao desenvolvimento de software em micro e pequenas empresas, através de literatura técnica sobre o assunto e entrevistas junto a duas empresas reais.

Remetendo aos objetivos da pesquisa, o foco principal foi a definição do framework para qualidade, visando orientar tanto a adoção, evolução, prospecção e transição de PDS, quanto o seu treinamento, em micro e pequenas empresas. Na execução destes objetivos, inicialmente foi utilizado o Modelo de Referência do MPS (MR-MPS) como referência para identificar os requisitos de qualidade do framework, para cada disciplina da ES. Com base nesse modelo ainda foram sugeridos os níveis do framework, de forma que as empresas possam ir adotando suas práticas de forma evolutiva, de acordo com suas possibilidades. A incorporação dos elementos do PDS (papéis, atividades, artefatos e ferramentas) ao framework foi feita com base nos elementos dos PDS estudados, que foram escolhidos através do levantamento bibliográfico citado.

3. RESULTADOS E DISCUSSÃO

Remetendo aos objetivos do trabalho, a pesquisa tem como principal resultado o Framework para Adoção, Evolução, Prospecção e Transição de Processos de Software, ou AdePT, um framework para apoiar a qualidade, especialmente em micro e pequenas empresas de desenvolvimento de software. O intuito é criar uma espécie de guia que ajudará as empresas a implantar e evoluir processos e assim sistematizar as atividades relacionadas a ES e a QS, contribuindo para o aumento na qualidade das suas rotinas organizacionais técnicas e consequentemente dos seus produtos. Nesse contexto, a partir do guia, das suas orientações e elementos em geral (inclusive do suporte ao framework, que poderá ser dado pela equipe do projeto, incluindo treinamento) qualquer empresa poderá implantar e usar o AdePT, de acordo com suas particularidades.

Embora o MPS.BR seja um modelo que se propõe a simplificar a implantação de PDS e de QS em empresas, a revisão bibliográfica feita neste trabalho ainda encontrou registros de dificuldades relacionadas à sua implantação. Rodrigues (2009) expõe algumas das dificuldades encontradas na implantação dos modelos de qualidade MPS.BR e CMMI: falta de equipe interna com competente específica, resistência a possíveis mudanças necessárias na cultura organizacional da empresa, falta de uma estratégia pré-definida para implantação, falta de comprometimento gerencial e operacional para a implantação do modelo, indisponibilidade de recursos humanos para atuarem nas atividades de implantação, falta de recursos financeiros, falta de treinamento. Porto et al. (2011) relataram a experiência adquirida na implantação de MPS.BR em uma pequena empresa de desenvolvimento de sistemas, apontando como os principais riscos e dificuldades encontradas a indisponibilidade de tempo da equipe da organização, a resistência à mudança, a falta de disseminação da cultura de qualidade e processos na empresa. Também referente à implantação do MPS.BR, Becker et al. (2011) exemplificaram como dificuldades: pouca disponibilidade de recursos humanos qualificados para atuar na implementação de processos (consultores de implementação), pouca experiência dos recursos humanos das empresas em melhoria de processo, pouca disposição inicial para troca de experiências entre as empresas, pouca disseminação inicial de conhecimentos sobre o modelo de referência MR-MPS, entre outras.



Considerando o material colhido através da literatura referente às dificuldades encontradas na implantação de PDS e modelos de qualidade em empresas, algumas decisões foram tomadas. Inicialmente, este projeto fez algumas adaptações aos níveis do modelo MR-MPS e seus respectivos processos e subprocessos, definindo os níveis, processos e subprocessos do AdEPT. Para cada nível do MR-MPS, surgiram dois níveis no AdEPT, um mais simplificado em termos de esforços técnicos (papéis, subprocessos, artefatos e ferramentas) e outro mais completo com relação ao MR-MPS, e portanto, com maior aderência a este modelo e às suas definições de qualidade (esse nível é chamado de plus, por exemplo, Nível G +). A definição dos níveis e processos do AdEPT pode ser vista na Tabela 1.

Tabela 1 – Níveis e processos do Framework AdEPT

Nível	Processos
A e A+	Otimização do PDS; Processos do Nível B (para A) ou B+ (para A+).
B e B+	Gerência de Projetos (evolução); Processos do Nível C (para B) ou C+ (para B+).
C e C+	Gerência de Riscos; Desenvolvimento para Reutilização; Gerência de Decisões; Processos do Nível D (para C) ou D+ (para C+).
D e D+	Verificação; Validação; Projeto e Construção do Produto; Integração do Produto; Desenvolvimento de Requisitos; Processos do Nível E (para D) ou E+ (para D+).
E e E+	Gerência de Projetos (Evolução); Gerência de Reutilização; Gerência de Recursos Humanos; Definição do Processo Organizacional; Avaliação e Melhoria do Processo Organizacional; Processos do Nível F (para E) ou F+ (para E+).
F e F+	Medição; Garantia de Qualidade; Gerência de Portfólio de Projetos; Gerência de Configuração; Aquisição; Processos do Nível G (para F) ou G+ (para F+).
G e G+	Gerência de Requisitos; Gerência de Projetos.

Além dos processos e subprocessos (que representam as atividades), foram especificados fluxos (que representam a execução dos processos e subprocessos no tempo), artefatos (sendo elaborados para cada subprocesso, de acordo com a sua necessidade), e ferramentas (para automatizar os subprocessos, quando possível, e de acordo com as características de cada um deles). No caso das ferramentas, optou-se por sugerir o uso de ferramentas gratuitas.

Um exemplo de como foi feita a estruturação do AdEPT pode ser visto através da apresentação do nível G do modelo. A Tabela 2 apresenta seus processos e subprocessos; a Tabela 3 apresenta os papéis, artefatos e ferramentas sugeridos por subprocesso, para equipes de empresas que se encontram no nível G de qualidade.

Tabela 2 – Processos e subprocessos do Nível G (Framework AdEPT)

Processo	Subprocessos
Gerência de Projetos (G-GP)	Definir escopo (G-GP-1) Monitorar escopo (G-GP-2) Dimensionar projeto (G-GP-3)



Monitorar dimensionamento do projeto (G-GP-4)
 Definir ou adaptar o PDS (G-GP-5)
 Gerar WBS (G-GP-6)
 Monitorar WBS (G-GP-7)
 Dimensionar esforço do projeto (G-GP-8)
 Dimensionar custos (G-GP-9)
 Gerar cronograma (G-GP-10)
 Monitorar cronograma (G-GP-11)
 Gerar orçamento (G-GP-12)
 Monitorar orçamento (G-GP-13)
 Identificar riscos (G-GP-14)
 Monitorar riscos (G-GP-15)
 Definir perfis para a equipe de projeto (G-GP-16)
 Monitorar perfis da equipe (G-GP-17)
 Definir recursos materiais para a execução do projeto (G-GP-18)
 Monitorar os recursos materiais (G-GP-19)
 Gerar plano de projeto (G-GP-20)
 Avaliar viabilidade do projeto (G-GP-21)
 Monitorar viabilidade (G-GP-22)
 Gerar contrato inicial/ contratos por requisição (G-GP-23)
 Registrar problemas das áreas monitoradas (G-GP-24)
 Traçar ações para correção de problemas (G-GP-25)

Gerência de Requisitos
 (G-GR)

Entendimento dos requisitos macro (G-GR-1)
 Avaliação dos requisitos macro (G-GR-2)
 Montar rastreabilidade dos requisitos (G-GR-3)
 Verificar consistência de requisitos (G-GR-4)
 Gerenciar mudanças nos requisitos (G-GR-5)

Tabela 3 – Subprocessos, papéis, artefatos e ferramentas do Nível G (Framework AdEPT)

Subprocesso	Papéis envolvidos	Artefato sugerido	Ferramenta sugerida
G-GP-1	Gerente de projeto (GP) e time (TM)	Escopo do projeto	Editor de texto gratuito (BrOffice)
G-GP-2	GP e TM	Escopo do projeto (atualizações)	Editor de texto gratuito (BrOffice)
G-GP-3	GP e TM	Contagem de pontos de função	Editor de texto gratuito (BrOffice)
G-GP-4	GP e TM	Contagem de pontos de função	Planilha gratuita (BrOffice)
G-GP-5	Responsável pela qualidade (RQ)	Guia do PDS	Editor de texto gratuito (BrOffice)
G-GP-6	GP	WBS	Ferramenta de gerência gratuita (OpenProj)



G-GP-7	GP	WBS (atualizações)	Ferramenta de gerência gratuita (OpenProj)
G-GP-8	GP e TM	Estimativas dos pacotes da WBS	Ferramenta de gerência gratuita (OpenProj)
G-GP-9	GP	Custos do projeto	Ferramenta de gerência gratuita (OpenProj)
G-GP-10	GP	Cronograma do projeto	Ferramenta de gerência gratuita (OpenProj)
G-GP-11	GP	Cronograma do projeto (atualizações)	Ferramenta de gerência gratuita (OpenProj)
G-GP-12	GP	Orçamento do projeto	Editor de texto gratuito (BrOffice)
G-GP-13	GP	Orçamento do projeto (atualizações)	Editor de texto gratuito (BrOffice)
G-GP-14	GP e TM	Mapa de riscos	Planilha gratuita (BrOffice)
G-GP-15	GP e TM	Mapa de riscos (atualizações)	Planilha gratuita (BrOffice)
G-GP-16	GP e TM	Perfis do time	Editor de texto gratuito (BrOffice)
G-GP-17	GP e TM	Perfis do time (atualizações)	Editor de texto gratuito (BrOffice)
G-GP-18	GP e TM	Mapeamento de recursos do projeto	Editor de texto gratuito (BrOffice)
G-GP-19	GP e TM	Mapeamento de recursos do projeto (atualizado)	Editor de texto gratuito (BrOffice)
G-GP-20	GP	Plano de projeto	Editor de texto gratuito (BrOffice)
G-GP-21	GP e TM	Parecer de viabilidade	Editor de texto gratuito (BrOffice)
G-GP-22	GP e TM	Parecer de viabilidade (atualizações)	Editor de texto gratuito (BrOffice)
G-GP-23	GP	Contrato comercial	Editor de texto gratuito (BrOffice)
G-GP-24	GP	Registro de problemas	Planilha gratuita (BrOffice)
G-GP-25	GP	Plano de ação para correção de problemas	Editor de texto gratuito (BrOffice)
G-GR-1	GP e TM	Documento de requisitos	Editor de texto gratuito (BrOffice)
G-GR-2	GP e TM	Documento de requisitos (atualizado)	Editor de texto gratuito (BrOffice)



G-GR-3	TM	Matriz de rastreabilidade	de Planilha gratuita (BrOffice)
G-GR-4	TM	Registro inconsistência	de Editor de texto gratuito (BrOffice)
G-GR-5	TM	Documento requisitos (atualizado)	de Editor de texto gratuito (BrOffice)

Um diferencial do AdEPT com relação a alguns modelos é que apesar destes modelos servirem como guia para auxiliar empresas na adoção de processos, alguns não definem detalhes específicos de que métodos ou ferramentas utilizar. Além disso, um fator determinante nesta adoção é o treinamento das pessoas. Foi com este intuito que o AdEPT pretendeu gerar um pacote para facilitar a melhoria dos processos de software em micro e pequenas empresas, definindo métodos, ferramentas e procedimentos da ES que serão utilizados por essas empresas na execução das suas atividades e, além disso, elaborando material para treinamento.

6. CONCLUSÕES

Este artigo apresentou o AdEPT, um framework para melhoria de processos de software voltado para as micro e pequenas empresas do ramo de desenvolvimento de software. O objetivo do framework é apoiar a necessidade de essas empresas reforçarem a sistematização das suas rotinas técnicas, e assim aprimorarem o seu processo produtivo, gerando ainda software com mais qualidade, e potencializando os seus aspectos de competitividade na indústria de desenvolvimento de software.

O AdEPT se propõe a ser uma alternativa capaz de fornecer a micro e pequenas empresas uma menor dificuldade para sua aderência aos padrões de qualidade baseados nas boas práticas de ES e QS. Nesse contexto o AdEPT se apresenta como um modelo contendo diretrizes agregadas a processos e subprocessos, artefatos, papéis e ferramentas capazes de guiar as empresas na aquisição de maturidade na execução de suas atividades de uma forma mais simplificada em relação a diversos outros modelos da indústria. Em virtude do seu nível de detalhamento, espera-se que o AdEPT possa servir como base para que as empresas possam usar suas orientações e projetar a sua evolução nos níveis de forma autônoma. Ainda assim, a equipe envolvida no projeto estará a disposição para gerar parcerias com essas empresas, buscando tanto o apoio à escalabilidade destas nos níveis do AdEPT, em busca de mais qualidade para seus PDS, como para obter feedback sobre o uso do framework, visando seu refinamento. Para isto, será montado um website com informações sobre o framework.

Além dos benefícios explícitos do framework, este trabalho sugere a implementação políticas de qualidade em empresas de software. Espera-se que o foco em qualidade seja percebido como uma forma de incentivo para as empresas que pretendem melhorar a cada dia as suas rotinas técnicas.

A primeira versão completa do AdEPT (o guia com todos os elementos construído e o website motado) está prevista para outubro de 2012. Como atividades futuras, está prevista a implantação do AdEPT na Empresa Júnior do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do IFPB – Campus Cajazeiras, empresa que desenvolve software. A ideia é que um consultor do AdEPT acompanhe esta implantação e colha as lições aprendidas e sugestões de melhoria do framework, favorecendo o lançamento de versões mais próximas às reais necessidades das micro e pequenas empresas.

AGRADECIMENTOS

A equipe do projeto agradece ao IFPB – Campus Cajazeiras, por fomentar e apoiar o desenvolvimento do projeto, e à professora Wilza Moreira (Coordenadora de Pesquisa e Extensão do Campus), por incentivar a participação dos pesquisadores em eventos científicos.

REFERÊNCIAS

AHERN, D., CLOUSE, A., TURNER, R. **CMMI Distilled: A Practical Introduction to Integrated Process Improvement**. Hardback, 2001.

ISBN 978-85-62830-10-5

VII CONNEPI©2012



ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - ABNT. **NBR ISO 12207: Tecnologia de Informação – Processos de Ciclo de Vida de Software**. Rio de Janeiro: ABNT, 1998.

ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO - SOFTEX. **MPS.BR - Melhoria de Processo de Software Brasileiro - Guia Geral**. Disponível em: <<http://www.softex.br/home/default.asp>>. Acesso em: 10 de dez 2011.

BECK, K. **Programação extrema explicada: acolha as mudanças**. Porto Alegre: Bookman, 2004.

BECK, K.; BOEHM, B. **Agility Through Discipline: A Debate**. Computer. Jun 2003, p. 45-46.

BECKER, C. A., PRIKLADNICKI, R., GALARRAGA, O. **Cooperativa MPS.BR – Relato de experiências, lições aprendidas, melhores práticas e dificuldades da II e IOGE SOFTEX do RS**. Disponível em: <<http://www.softex.br/portal/softexweb/uploadDocuments/mpsbr/SOFTSUL.pdf>>. Acesso em: 15 nov 2011.

FAROLDIGITAL. **Farol Digital**. Disponível em: <<http://www.faroldigital.org.br/>>. Acesso em: 10 jul 2011.

HIGHSMITH, J. **Agile software Development Ecosystems**. Addison- Wesley, 2002.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION - ISO. **ISO/IEC 15504-1: 2004**. ISO, 2004.

KRUCHTEN, P. **Introdução ao RUP**. Rio de Janeiro: Ciência Moderna Ltda., 2003.

PORTO, J. B. et al. **Experiência de Avaliação MPS.BR Nível F na Qualidade**. Disponível em: <<http://www.softex.br/portal/softexweb/uploadDocuments/mpsbr/T2-Qualita-WE.pdf>>. Acesso em: 24 nov 2011.

PRESMAM, R. S. **Engenharia de Software**. 7ª Edição. Porto Alegre: AMGH, 2011.

RISING, L.; JANOFF, N. S. **The Scrum Software Development Process for Small Teams**. In Software, Jul/Aug 2000, p. 26-32.

RODRIGUES, J. F. **Avaliação da Implantação do Mps.Br: Um Estudo Empírico Sobre Benefícios, Dificuldades e Fatores de Sucesso**. 2009. 191 F. Dissertação (Mestrado em Ciência da Computação) - Universidade Metodista de Piracicaba, Piracicaba, 2009.

RODRIGUES, N. N. **Virtualização bancária: a experiência em João Pessoa – PB**. 2002. 157 f. Dissertação (Mestrado em Administração de Empresas) – Universidade Federal da Paraíba, João Pessoa, 2002.

SOFTWARE ENGINEERING INSTITUTE - SEI. **CMMI for Development**. Disponível em: <<http://www.sei.cmu.edu/cmmi/solutions/dev/>>. Acesso em: 20 dez 2011.

SOMMERVILLE, I. **Engenharia de Software**. 9ª Edição. São Paulo: Pearson Prentice Hall, 2011.

ZAHRAN, S. **Software Process Improvement – Practical Guidelines for Business Success**. Addison – Wesley, 1998, 447 p.