



5º Congresso de Pós-Graduação

MÉTODO DE AUTO-AVALIAÇÃO DE PROCESSOS DE SOFTWARE PARA PEQUENAS EMPRESAS

Autor(es)

THIAGO SALHAB ALVES
LUIZ EDUARDO GALVÃO MARTINS

Orientador(es)

Luiz Eduardo Galvão Martins

1. Introdução

Equipes e organizações desenvolvedoras de software muitas vezes adotam diferentes práticas de desenvolvimento, criando o que se costuma chamar de processos ad hoc de desenvolvimento de software. Infelizmente esses processos ad hoc costumam ser pouco controlados, não passíveis de repetição e altamente dependentes da capacidade individual de cada membro da equipe. A partir da década de 1990, vários modelos de maturidade de processos vêm sendo propostos com o objetivo de auxiliar na melhoria da qualidade dos processos de software adotados pelas organizações. Entre eles podemos citar os modelos CMM (PAULK et al., 1993), Spice - ISO/IEC 15504 (CÔRTEZ, 2001), CMMI (CMMI-DEV, 2006) e mais recentemente, no Brasil, o MPS-BR (MPS, 2004). Atualmente as organizações brasileiras que desenvolvem software, estão atentas à necessidade da adoção de processos de software melhores definidos, e percebe-se movimentos das organizações brasileiras em busca de certificações de qualidade de processos de software, notadamente as certificações CMMI de nível 2. Infelizmente essas certificações tem um alto custo para implantação, são muito rígidos e criteriosos e influenciam no fluxo das atividades diárias da organização. Neste sentido, a proposta de desenvolvimento de um método de auto-avaliação de processos de software, apoiado por ferramenta automatizada, demonstra-se pertinente e atual, apresentando-se como um instrumento que oferecerá auxílio às pequenas organizações e equipes de desenvolvimento de software na melhoria de seus processos, sendo de baixo custo para organização, confiável e flexível. Este método se baseia no documento CMMI-DEV (Capability Maturity Model Integration for Development), nos requisitos de avaliação para o CMMI (ARC V1.2 - Appraisal Requirements for CMMI) e no método de avaliação SCAMPI A (Standard CMMI Appraisal Method for Process Improvement A).

Revisão Bibliográfica O CMMI-DEV (Capability Maturity Model Integration for Development) é um modelo de maturidade de melhoria de processos para o desenvolvimento de produtos e serviços. As designações do CMMI para engenharia de sistemas e engenharia de software (CMMI-SE/SW) foram substituídas pelo título "CMMI for Development" para realmente refletir a integração desses corpos de conhecimento e aplicações do modelo com a organização (CMMI-Dev, 2006). De acordo com o documento CMMI-Dev (2006), CMMI

for Development Versão 1.2 é a continuação e atualização do CMMI versão 1.1 e foi otimizado com o conceito de constelações. CMMI-DEV é o primeiro de tais constelações. O CMMI permite acessar a melhoria de processo e avaliação usando duas diferentes representações: contínua e por estágio. Cada representação possui níveis, que são usados em CMMI para descrever um trajeto evolucionário recomendado para uma organização que quer melhorar os processos que usa para desenvolver e manter os produtos e serviços (CMMI-Dev, 2006). Os cinco níveis de maturidade utilizados para a Representação por Estágio são: Inicial, Gerenciado, Definido, Gerenciado Quantitativamente e Otimizado. O Appraisal Requirements for CMMI (ARC) v 1.2 define os requisitos considerados essenciais para classes de métodos de avaliação que pretendem ser usados com o CMMI-DEV, (SCAMPI, 2006a). De acordo com o SCAMPI (2006a), os requisitos são alocados em cada classe, baseado nos atributos associados com essa classe. Assim, um método de avaliação pode ser declarado para ser um ARC classe A, B ou C. Nem todos os métodos de avaliação CMMI são esperados a estar em total conformidade com o ARC, isto é, satisfazer cada requisito ARC. Segundo o SCAMPI (2006a), os grupos de requisitos do ARC para Classes do Método de Avaliação são: Responsabilidades, Documentação do Método de Avaliação, Planejamento e Preparação para Avaliação, Coleção dos Dados de Avaliação, Validação e Consolidação de Dados, Pontuação e Relato de Resultados. O SCAMPI A (Standard CMMI Appraisal Method for Process Improvement A), foi desenvolvido para prover pontuações para modelos CMMI. É aplicado a uma vasta escala de modos de avaliação, incluindo melhoria de processo interno e determinação de capacidade externa (SCAMPI, 2006). Segundo o SCAMPI (2006), o SCAMPI A satisfaz todos os requisitos do Appraisal Requirements for CMMI (ARC) para o método de avaliação Classe A. O Documento de Definição do Método, Classe A SCAMPI v1.2, descreve os requisitos, atividades, e práticas associadas com cada processo que compõem o método SCAMPI A. De acordo com o SCAMPI (2006), o SCAMPI A consiste de 3 fases: Planejar e Preparar para Avaliação, Conduzir Avaliação e Reportar Resultados. Cada fase possui processos essenciais que devem ser atendidos.

2. Objetivos

Desenvolvimento de um método para auto-avaliar processos de software adotados por pequenas organizações, ou pequenas equipes desenvolvedoras de software que desejem confrontar suas práticas atuais de desenvolvimento frente ao CMMI-DEV nível 2 de Maturidade – Representação por Estágio. Os objetivos do método são: Ser utilizado por pequenas organizações ou pequenas equipes de desenvolvimento para uma auto-avaliação. Ajudar organizações a identificar o estado atual de seus processos. Servir como plano de melhoria de processo de desenvolvimento de software. Apresentar baixo custo para a organização. Ser confiável e simples. Tempo flexível (a organização determinará o tempo das atividades da avaliação). Não influenciar no fluxo das atividades diárias da organização (a organização não receberá avaliadores externos).

3. Desenvolvimento

Aprofundamento teórico sobre Melhoria do Processo de Software, adquirindo conhecimentos sobre Maturidade das Organizações, Sintomas de Imaturidade, Prejuízos da Imaturidade, Riscos e Benefícios da Melhoria dos Processos. Esses conhecimentos foram úteis para identificar os pontos negativos de uma empresa imatura e os benefícios da melhoria dos processos. Aprofundamento teórico sobre Modelos de Maturidade e Capacidade de Software, iniciando o estudo com o modelo CMM (Capability Maturity Model) (PAULK et al., 1993), incluindo seminários apresentados ao orientador. Com esse aprofundamento, verificou-se o funcionamento do modelo CMM e o problema dos múltiplos CMMs que existiam, desviando o foco das empresas que necessitavam melhorar seus processos. Aprofundamento teórico através de seminários que apresentaram o modelo CMMI (Capability Maturity Model Integration) em sua versão 1.1 (CMMI, 2002) e também o novo modelo CMMI-DEV (Capability Maturity Model Integration for Development), versão 1.2. Esses dois documentos, principalmente o CMMI-DEV, constituem a base para identificar os objetivos específicos e genéricos assim como as práticas específicas e genéricas para o nível 2 de maturidade, que é o propósito do método a ser implementado neste trabalho. Aprofundamento teórico

através de seminários que apresentaram o ARC (Appraisal Requirements for CMMI) versão 1.2 (SCAMPI, 2006a). Este é um dos principais documentos, pois através dele identifica-se os requisitos essenciais para ser usado com o CMMI. Aprofundamento teórico através de seminários que apresentaram o SCAMPI A (The Standard CMMI Appraisal Method for Process Improvement) (SCAMPI, 2006). Este documento é essencial para prover as pontuações, pois é o único que satisfaz todos os requisitos ARC. Início da escrita da dissertação, começando com a revisão bibliográfica dos assuntos citados acima, buscando deixar clara a funcionalidade dos documentos, através de tópicos que exploram os assuntos. Levantamento dos Requisitos para o desenvolvimento do método de auto-avaliação, projetado para auxiliar equipes de desenvolvimento de software a fazer uma auto-avaliação de suas práticas de desenvolvimento frente ao modelo CMMI-DEV, representação por estágio nível 2. Análise e construção de tabelas de dependência entre Práticas Específicas dentro de uma mesma área de processo. Descrição do método de auto-avaliação de processos de software para pequenas empresas.

4. Resultados

Através do aprofundamento teórico (através de estudos dos modelos CMMI-DEV, ARC e SCAMPI A) e do levantamento dos requisitos para o desenvolvimento do método de auto-avaliação de processos de software, verificou-se como necessidade para a construção do método a verificação de possíveis dependências existentes entre as práticas específicas em uma mesma área de processo do nível 2 de maturidade. Essa verificação foi realizada para as sete áreas de processo do CMMI-DEV nível 2 de maturidade (Gerência de Requisitos, Planejamento do Projeto, Monitoramento e Controle do Projeto, Gerência de Acordo com Fornecedores, Medida e Análise, Garantia da Qualidade do Produto e Processo, Gerência da Configuração) e como resultado obteve-se as dependências entre as práticas específicas em cada área de processo, sendo um fator importante para a pontuação das práticas específicas que vão ser avaliadas na auto-avaliação. Atualmente está sendo elaborado o método de auto-avaliação de processo de software, que terá suporte de uma ferramenta automatizada que será desenvolvida através da linguagem JAVA, sendo um ambiente totalmente visual e de fácil interação com o usuário. Inicialmente o método de auto-avaliação se divide em três fases: Fase I – Preparar para Avaliação 1. Preparar Avaliação 1.1 Determinar o Escopo da Avaliação 1.2 Preparar Equipe de Avaliação 1.2.1. Identificar o Coordenador da Avaliação 1.2.2. Selecionar os Membros do Time de Avaliação 1.3 Obter Lista Inicial das Evidências Objetivas 1.3.1. Obter as Listas Objetivas Iniciais através de Documentos Na fase I, a organização vai se preparar para iniciar o processo de auto-avaliação. Determina-se o escopo da avaliação (projeto significativo ou processo a ser avaliado), identifica-se o coordenador da avaliação (membro da organização com conhecimento básico em CMMI) que vai coordenar as fases da avaliação e seleciona-se os membros da equipe (membros da organização envolvido no projeto a ser avaliado). Sugere-se que a equipe de auto-avaliação seja composta de pelo menos dois integrantes (coordenador e um membro). Com a equipe formada e interada sobre a avaliação, inicia-se o levantamento das evidências objetivas (documentos ou afirmações que comprovem a implementação de uma prática específica ou genérica) relativas às setes áreas de processos. Fase II – Avaliação 2. Conduzir Avaliação 2.1. Classificar Evidências Objetivas 2.2. Gerar Resultados de Avaliação 2.2.1. Pontuar as Práticas Específicas e Genéricas 2.2.2. Pontuar Objetivos Específicos e Genéricos 2.2.3. Pontuar Área de Processo Na fase II, inicia-se a auto-avaliação do projeto. Com a coleta das evidências objetivas para as práticas específicas e genéricas realiza-se a classificação para cada produto típico de trabalho (saídas produzidas pelo uso de uma prática) dessas evidências em artefatos diretos (saídas diretas da implementação de uma prática, ex: documentos, materiais de entrega, etc.), artefatos indiretos (saídas que são o subproduto de uma prática, ex: lista de erros encontrado em um diagrama de classes) e afirmações (declarações orais ou escritas confirmando a prática, ex: apresentações, demonstrações, etc.) Com base nessa classificação e submetendo essas respostas a um conjunto de regras, pode-se pontuar as práticas específicas e genéricas: · Completamente Implementada (CI) · Largamente Implementada (LI) · Parcialmente Implementada (PI) · Não Implementada (NI) · Ainda Não Implementada (ANI) Consequentemente os objetivos são classificados como Não Pontuados, Não Satisfeitos ou Satisfeitos e com base na classificação dos objetivos, as áreas de processo devem ser pontuadas como Satisfeita, Não Satisfeita, Não Aplicada, Fora de Escopo ou Não

Pontuada. Fase III – Resultados 3. Relatar Resultados 3.1. Verificar o atendimento das dependências entre as práticas 3.2. Entregar Descobertas Finais 3.3. Planejar Próximos Passos Na fase III, relata-se os resultados da avaliação para a organização. Esse relato deve apontar para a organização quais práticas, objetivos e áreas de processo merecem mais atenção e quais delas estão em conformidade com o modelo CMMI. É feita a verificação das dependências entre as práticas de uma mesma área de processo, pois se uma prática específica de um objetivo foi classificada como Parcialmente Implementada (PI), Não Implementada (NI) ou Ainda Não Implementada (ANI), essa classificação se deve ao fato de alguma prática que se depende, também ter sido classificada dessa forma. Assim o objetivo é mostrar as relações de dependências entre as práticas em uma mesma área de processo e apontar onde estão os “pontos fracos” e sugerir melhorias. Dessa forma, informa-se através de um relatório, as análises encontradas, permitindo assim que a organização possa criar um plano de melhoria de seus processos.

5. Considerações Finais

Este trabalho está vinculado à dissertação de mestrado em Ciência da Computação que ainda não foi concluída e está em desenvolvimento. Atividades concluídas: Desenvolvimento do trabalho de qualificação e sua aprovação pela banca examinadora. Análise das dependências entre práticas específicas de uma mesma área de processo do nível 2 de maturidade. Proposta das três fases do método de auto-avaliação. Atividades em andamento: Validação das dependências entre as práticas específicas encontradas. Descrição detalhada das três fases do método de auto-avaliação. Atividades a desenvolver: Validação do método de auto-avaliação. Aplicação de um Estudo de Caso para análise do método de auto-avaliação. Implementação de ferramenta através da Linguagem Java que suporte as fases do método de auto-avaliação.

Referências Bibliográficas

CMMI PRODUCT TEAM, CMMI-DEV, Version 1.2, Carnegie Mellon University, Aug/2006.

Disponível em: <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>

CÔRTEZ, M. L. e CHIOSSI, T. C. S, “Modelos de Qualidade de Software”, Editora Unicamp, 2001.

MPS-BR - Melhoria do Processo de Software Brasileiro, Notas de Apresentação do *Workshop* do Projeto melhoria de processo do software Brasileiro (mps BR), Campinas, Outubro/2004.

PAULK, M.C., CURTIS, B., CHRISSIS, M.B., WEBER., C.W., “Capability Maturity ModelSM for Software, Version 1.1”, Carnegie Mellon University, February 1993.

Disponível em:

<http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr24.93.pdf>

SCAMPI UPGRADE TEAM, “Appraisal Requirements for CMMI, Version 1.2”, Carnegie Mellon University, August 2006a

Disponível em: <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr011.pdf>

SCAMPI UPGRADE TEAM, "Standard CMMI Appraisal Method for Process Improvement (SCAMPI) A, Version 1.2: Method Definition Document", Carnegie Mellon University, August 2006.

Disponível em: <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06hb002.pdf>