

Evolução da qualidade de software : Um caminho viável

Sérgio França^{*,1}, Tatu Nakanishi^{**,2}, João Bosco Schumann Cunha^{**,2}

(1)Área de Sistemas de Informação

SRC – Serviço de Recursos Computacionais
Instituto Nacional de Pesquisas Espaciais (INPE)

(2)Centro de Ciências Exatas e de Tecnologia
Universidade de Mogi das Cruzes (UMC)

(*)Mestrado, e-mail: franca@sir.inpe.br (**) Orientadores

Resumo

Muitos setores do INPE – Instituto Nacional de Pesquisas Espaciais estão envolvidos diretamente com o desenvolvimento de software. Poucos, porém, estabeleceram parâmetros formais de Engenharia de Software para orientar tal desenvolvimento. Este artigo propõe a implementação de um conjunto de atividades que visam não somente atender as orientações da Engenharia de Software, mas, principalmente, estabelecer um *framework* que sirva de base para muitas organizações desenvolvedoras (departamentos, setores, grupos, etc.) que têm nível técnico e condições de trabalho semelhantes aos aqui descritos, do departamento SRC – Serviço de Recursos Computacionais do INPE. A maioria das atividades propostas é prevista no modelo CMM – Capability Maturity Model, com forte ênfase em seu nível 2 (Repetível) de maturidade, porém com a inserção de algumas atividades previstas em outros níveis do modelo CMM e outras do modelo SPICE – Software Process Improvement & Capability dTermination. Uma característica marcante no trabalho é a apresentação de modelos e documentos de exemplos práticos, assim como a seleção de um conjunto de ferramentas que possibilitem realizar as atividades propostas.

Palavras-Chave: engenharia de software, CMM, ferramentas, prática

Introdução

Este trabalho apresenta um conjunto de atividades que podem ser implementadas em um ambiente de desenvolvimento de software no sentido de obter melhorias consideráveis nos processos de desenvolvimento atualmente em uso, assim como uma maneira prática de implementá-las. Para tanto, baseia-se nos modelos CMM – Capability Maturity Model (Modelo de Maturidade da Capacidade) e SPICE – Software Process Improvement & Capability determination (Melhoria no processo de software e determinação de capacidade).

Tais modelos – conhecidos como modelos de melhoria de processos – descrevem de maneira completa “o quê” deve ser feito; este trabalho consiste, porém, em mostrar, na prática, “como fazer” para implementar tais medidas de maneira clara e simples, para um determinado caso.

Os modelos CMM e SPICE são baseados em evoluções gradativas no processo de desenvolvimento. Tais evoluções são caracterizadas por **níveis de maturidade**. Embora ambos apresentem propostas para uma evolução até níveis de maturidade muito avançados – a ponto de estarem plenamente implementados em poucas organizações na atualidade – este trabalho pretende apresentar, de maneira realista, as características desejáveis para a obtenção de um ambiente equivalente ao segundo nível do modelo CMM, com o acréscimo de algumas atividades de outros níveis assim como alguns aspectos correspondentes no modelo SPICE.

Componentes para a implementação do trabalho proposto

Para o perfeito entendimento do que se pretende obter, assim como para possibilitar que o trabalho proposto seja plenamente implementado, é necessário que se considere três componentes básicos :

- um ambiente que atualmente necessita de melhorias em seu processo de desenvolvimento;
- o entendimento e a aceitação do que se propõe como melhoria da qualidade do software desenvolvido, baseada nos modelos CMM e SPICE;
- uma explicação detalhada de cada um dos passos que este trabalho recomenda, assim como maneiras práticas e tangíveis de execução das atividades propostas.

Razões para o desenvolvimento do trabalho

Por que apresentar um trabalho sobre este assunto ? Basicamente, porque as organizações que desenvolvem software reconhecem que há atualmente a necessidade de aplicar técnicas e métodos de gerência de projetos diferentes dos que têm sido utilizados até o presente, visando aumento de produtividade e maior controle e previsibilidade sobre o processo de desenvolvimento.

Nota : o termo **organização** será usado doravante no sentido de **qualquer unidade desenvolvedora de software** : software houses, departamentos, equipes de desenvolvimento, etc.

Estas técnicas e métodos são, naturalmente, abrangidos pelo processo de Engenharia de Software, que recomenda a adoção de um padrão, ou seja, um modelo adaptável às características de cada ambiente, a fim de reduzir custos e obter o desejado aumento de produtividade através da **melhoria constante dos processos** de desenvolvimento e, conseqüentemente, da melhoria da qualidade dos produtos gerados.

Tanto CMM quanto SPICE propõem, portanto, diretrizes para, a partir do conhecimento da real situação da organização, implementar de maneira progressiva os procedimentos necessários para que a mesma evolua em termos de maturidade e capacidade dos processos de software.

Parece haver consenso na comunidade de Engenharia de Software no sentido de que ambos modelos têm como ponto forte a identificação precisa dos problemas encontrados no desenvolvimento, assim como têm como ponto fraco o uso de generalizações tão abrangentes que dificultam a identificação do que é apropriado para cada caso em cada organização. Considerando as peculiaridades de uma organização e ambiente específicos, concluímos pela necessidade do uso de recomendações de ambos modelos, nos aspectos em que cada um deles melhor se adequar aos interesses do ambiente onde serão aplicados.

A melhoria na qualidade do software desenvolvido

Os modelos CMM e SPICE são reconhecidos mundialmente como duas das melhores propostas para a capacitação e aumento da maturidade das equipes de desenvolvimento de software. Ambos visam um único objetivo maior, que poderia ser identificado como a razão da existência da própria Engenharia de Software : produzir software com padrões de qualidade cada vez melhores. Por qualidade pode-se entender vários fatores, tais como satisfação dos usuários, facilidade de manutenção, atendimento das restrições de custo, cumprimento de prazos, etc.

Outra característica que marca estes dois modelos é a proposta de melhorias gradativas, considerando que, no mundo real, uma revolução brusca e acentuada na melhoria da qualidade dos produtos desenvolvidos quase nunca se mostra viável.

Contribuição deste trabalho

Veremos, no entanto, que embora os modelos apresentem atividades bem definidas quanto aos objetivos a atingir, o mesmo não acontece no que diz respeito à maneira como estas devem ser implementadas. É neste ponto que este trabalho se encaixa : pretende apresentar, dentre as atividades selecionadas como essenciais para um ambiente, sugestões sobre como proceder para a sua implementação, levando em consideração todas as peculiaridades de um ambiente existente na prática e não apenas na teoria.

Considerando que o ambiente aqui descrito representa a realidade de muitas organizações, este trabalho pode tornar-se útil para qualquer organização que se encontre com características semelhantes às aqui descritas. Não há, entretanto, a pretensão de se estabelecer “o caminho”, mas “**um caminho**” a ser seguido.

Uma preocupação constante que deve haver por parte das organizações que se proponham a implementar o *framework* aqui apresentado é com relação à boa aceitação pelas pessoas envolvidas. Neste aspecto, considerando que mudanças radicais podem provocar rejeições também radicais, ou seja, a fim de facilitar a aceitação por parte das pessoas envolvidas, a abordagem escolhida para a implementação das atividades foi montada visualizando-se três estágios com exigências de rigor, qualidade e automação crescentes. As atividades selecionadas como essenciais, assim como sua distribuição nos três estágios propostos podem ser vistas na tabela 1.

Estágios	1.o Estágio	2.o Estágio	3.o Estágio
Atividades	- Baixo Custo - Registros preliminares - Treinamento mínimo	- Treinamento específico - Ferramentas CASE	- Gerência (avaliação sobre os estágios anteriores) - Evolução
1) Gerência de Requisitos	- Conscientizar sobre necessidades de Ger. de Requisitos - Registro Preliminar de Requisitos	- Seleção de CASE - Treinamento específico - Registro com CASE	- Verificar rastreabilidade (em todo o processo)
2) Planejamento do Projeto de Software	- Docto “Estimativas Preliminares” - Planejamento Simplificado - Aprovação do Planejamento	- Seleção de técnica e ferramenta de medição (Métricas)	- Promover uso de ferramenta de work-flow para o planejamento - Determinação da capacidade para atividades específicas (SPICE)
3) Supervisão e Acompanhamento do Projeto de Software	- Adequar produtos - Medir produtos de software	- Treinamento de gerentes (ênfase nas áreas-chave do processo)	- Implementar alterações no Plano de Desenvolvimento de software - Propor métricas alternativas
4) Garantia da Qualidade	- Treinamento de gerentes (Modelos, CMM e Qualidade) - Planejamento da GQ - Revisão de produtos	- Padrões de código - Registro de informações medidas com CASE - Testes funcionais	- Gerência de mudanças de tecnologias
5) Programa de Treinamento	- Treinamento Específico Emergencial Mínimo	- Cronograma Assunto versus Disponibilidade	
6) Gerência de Configuração	- Documento de “Responsabilidades” - Controle de Config. Não-automatizado	- Seleção de ferramenta	- Verificar resultados
7) Conduzir Processo de Avaliação	- Definir Assessor Qualificado - Definir Fronteiras da Avaliação - Avaliar Processos e Produtos do estágio	- Definir Fronteiras da Avaliação	- Registro de avaliações - Propor e implementar alterações

Tabela 1 : os três estágios propostos com suas respectivas atividades