

"GESTÃO DE CONFIGURAÇÃO SOFTWARE: UMA VISÃO ESTABELECIDADA PELAS NORMAS E MODELOS DE QUALIDADE".

MARCELO NOGUEIRA
Universidade Paulista
Rua Dr. Bacelar 1212 – 4º - CEP 04026-002 – São Paulo – SP.
marcelo@noginfo.com.br

Resumo

As empresas de desenvolvimento de sistemas de pequeno porte têm seus recursos materiais e humanos tanto quanto limitados e sobrecarregados, diante da demanda a ela submetida. Porém a não implementação da gerência de configuração nos projetos de sistemas, podem causar sérios danos e prejuízos, quando requisitos são alterados por qualquer natureza e não são controlados e rastreados. A confiabilidade do produto de software é influenciada pelo processo de desenvolvimento do software. Um processo repetitivo de controle, orientado no sentido de evitar defeitos, poderá provavelmente desenvolver um sistema confiável. A não utilização dessas práticas fundamentais da Engenharia de Software ou por falta de cultura ou até por desconhecimento da sua existência, levam projetos de suma relevância ao insucesso e aumentando ainda mais a quantidade de casos de fracasso no desenvolvimento de sistemas, conseqüentemente desperdiçando tempo, recursos financeiros e não atendendo as necessidades dos clientes, desviando os objetivos do desenvolvedor.

Palavras-chave: Gerência de Configuração, Engenharia de Software, Pequenas Empresas.

Abstract

The systems development companies of small size have their material and human resources as much as limited and overloaded, in front of the demand to her submitted. However the not implementation of the configuration management in the projects of systems, they can cause serious damages and prejudices, when requisites are changed by any nature and are not controlled and traced. The reliability of the software product is influenced by the development process of the software. A repetitive process of control, guided in the sense of avoiding defects, probably can develop a reliable system. The not utilization of these fundamental practices of the Software Engineering or for lack of culture or until for ignorance of your existence, they carry projects of vanishes relevance to the failure and even increasing the quantity of failure cases in the systems development, consequently wasting time, financial resources and not attending the needs to clients, diverting the goals of the developer.

Key words: Configuration management, Software Engineering, Small size companies.

1. INTRODUÇÃO

Num ambiente de desenvolvimento de software altamente complexo, a gestão de configuração de software tem caráter essencial quando as modificações e alterações são inevitáveis durante todo o ciclo de vida do software. Estas alterações geram confusão aos desenvolvedores. A confusão aparece quando estas alterações não são analisadas antes de serem feitas, não são registradas antes de serem codificadas, e não são relatadas aos envolvidos no sentido de controlar e evitar erros.

Tratando-se de sistemas, onde a confiabilidade é fator de sucesso em sua utilização, a falta de controle nas alterações pode trazer transtornos de ordem humana, cumprimento de objetivos e prejuízos nos negócios.

A gestão de configuração de software é um tema simultaneamente abrangente e especializado, dentro das melhores práticas da Engenharia de Software que possibilitará o desenvolver consolidar os conhecimentos no desenvolvimento de software, bem como preparar os micros, pequenos e médios desenvolvedores para encarar com confiança os novos desafios no mundo dos negócios, e também reforçar as competências profissionais, mantendo-se atualizado em relação ao potencial dos

sistemas de informação e das novas tecnologias numa perspectiva empresarial e competitiva globalmente.

A partir do conhecimento adquirido o desenvolvedor será elemento multiplicador de soluções, contribuindo e agregando valor aos sistemas com aplicação de metodologias e tecnologias adequadas, capazes de gerir com sucesso as informações relevantes aos negócios aplicáveis, trazendo às organizações, vantagens competitivas.

No estudo da Engenharia de Software, o autor Roger S. Pressman [PRESSMAN02], demonstra preocupação com a “Crise do Software” que atualmente ele intitula como “Aflição Crônica”, chegando a determinar números expressivos sobre a não finalização de projetos de sistemas começados e não terminados.

Num mundo cada vez mais de recursos financeiros escassos, como é possível aceitar tal desperdício de tempo e dinheiro. O mesmo autor também aponta para o possível problema causador de tal absurdo: “A falta de adoção de métodos, ferramentas e procedimentos no desenvolvimento de software e a difícil relação de entendimento entre o usuário com o analista desenvolvedor”.

Várias técnicas de gestão de projetos foram criadas e o desenvolvedor mesmo assim ainda tem dificuldades de realizar projetos de sistemas livres de manutenções e correções constantes, condenando diretamente a qualidade do produto entregue.

A adoção da Gestão de Configuração de Software no desenvolvimento de Sistemas, pelo micro, pequeno e médio desenvolvedor de software, direciona como estruturar e gerenciar e controlar por todo o ciclo de vida do software, proporcionando acompanhamento de todo o processo, permitindo que o software venha representar a realidade da empresa modelada para geração de um sistema adequado, atendendo assim seus requisitos e as necessidades desta empresa.

Adotar modelos de gestão, permite padronizar os processos de desenvolvimento de software, aumentando a aderência a uma ferramenta de apoio, agregando a elas facilidades de relacionamento e compreensão do Usuário e do Desenvolvedor, utilizando métodos sistêmicos que diminuirão erros, e por consequência a manutenção exagerada, eliminação de custos indevidos, extinção do tempo desnecessário dedicado ao re-trabalho, obtendo qualidade no software, bem como criar a real possibilidade de extrair de um sistema, informações relevantes que venham não só para contribuir com a decisão, mas para ser um fator de excelência empresarial, permitindo novos negócios, permanência e sobrevivência num mercado atuante.

2. RELEVÂNCIA

Segundo o Ministério da Ciência e Tecnologia [MCT02], a participação de mercado das micros, pequenas e médias empresas desenvolvedoras de software correspondem 65,1% do total (Tabela1). Diante da dimensão deste mercado ainda em franca expansão, que motivou a prestar atenção especial para este tema.

Tabela 1- Porte das Organizações, segundo comercialização bruta anual (2000).

<i>Porte das Organizações</i>	
Portes	%
Micro	16,5
Pequena	25,3
Média	23,3
Grande	34,9

Atualmente com a visão global permitindo a participação nas exportações de software para outros países, cada vez mais a qualidade no processo de desenvolvimento e do produto de software ganha maior observação e adoção das melhores práticas e soluções tecnológicas.

3. ENGENHARIA DE SOFTWARE

Segundo Rezende [REZENDE99], Engenharia é a arte das construções, embasada no conhecimento científico e empírico, adequada ao atendimento das necessidades humanas.

Engenharia de Software é a metodologia de desenvolvimento e manutenção de sistemas modulares, com as seguintes características [REZENDE99]:

- Adequação aos requisitos funcionais do negócio do cliente e seus respectivos procedimentos pertinentes;
- Efetivação de padrões de qualidade e produtividade em suas atividades e produtos;
- Fundamentação na tecnologia da informação disponível, viável e oportuna;
- Planejamento e gestão de atividades, recursos, custos e datas.

Segundo Pressman [PRESSMAN95], Engenharia de Software é:

- “O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais”;
- “Descendente da engenharia de sistemas e de hardware. Abrange um conjunto de 3 elementos fundamentais (métodos, ferramentas e procedimentos), que possibilita, ao gerente, o controle do processo de desenvolvimento do software e oferece ao profissional uma base para a construção de software de alta qualidade”.

Segundo Martin [MARTIN91], Engenharia de Software é:

- “É o estudo dos princípios e sua aplicação no desenvolvimento e manutenção de sistemas de software”;
- “Tanto a engenharia de software como as técnicas estruturadas são coleções de metodologias de software e ferramentas”;

Como conclusão, pode-se relatar que engenharia de software é um conjunto de práticas para desenvolvimento de soluções de software, ou seja, roteiro que pode utilizar diversas técnicas.

A seqüência de passos preestabelecidos permite optar e variar de técnicas e ferramentas na suas diversas fases [REZENDE99].

4. OBJETIVOS DA ENGENHARIA DE SOFTWARE

De um modo geral, considera-se que os objetivos primários da Engenharia de Software são o aprimoramento da qualidade dos produtos de software e o aumento da produtividade dos engenheiros de software, além do atendimento aos requisitos de eficácia e eficiência, ou seja, efetividade [MAFFEO92].

A Engenharia de Software visa sistematizar a produção, a manutenção, a evolução e a recuperação de produtos intensivos de software, de modo que ocorra dentro de prazos e custos estimados, com progresso controlado e utilizando princípios, métodos, tecnologia e processos em contínuo

aprimoramento. Os produtos desenvolvidos e mantidos, seguindo um processo efetivo e segundo preceitos da Engenharia de Software asseguram, por construção, qualidade satisfatória, apoiando adequadamente os seus usuários na realização de suas tarefas, operam satisfatória e economicamente em ambientes reais e podem evoluir continuamente, adaptando-se a um mundo em constante evolução [FIORINI98].

Associando a esses objetivos, o termo engenharia pretende indicar que o desenvolvimento de software deve submeter-se a leis similares às que governam a manufatura de produtos industriais em engenharias tradicionais, pois ambos são metodológicos [MAFFEO92].

Com base nos objetivos da Engenharia de Software, fica evidente a necessidade da adoção da gestão sistêmica das configurações do software como princípios de qualidade para padronizar e gerenciar os processos de desenvolvimento de software.

5. CRISE DO SOFTWARE

Para generalizar o termo, ocorre quando o software não satisfaz seus envolvidos, sejam clientes e/ou usuários, desenvolvedores ou empresa [REZENDE99].

A expressão “Crise do Software”, que começou a ser utilizada na década de 60, tem historicamente aludido a um conjunto de problemas recorrentemente enfrentados no processo de desenvolvimento (Construção, implantação e manutenção) de software [MAFFEO92].

Esses problemas não se referem apenas a programas que não funcionam. Na verdade, a chamada “Crise do Software” abrange todos os problemas relacionados a [REZENDE99]:

- Como sistemas computacionais são construídos e implantados, referindo-se aqui ao processo de substituir sistemas antigos, desativando sistemas correntemente em operação, ou ao processo de instalar um sistema inteiramente novo;
- Como é provida a manutenção da quantidade crescente de software construído, associado a sistemas computacionais cada vez mais complexos;
- Como fazer face à crescente demanda para construção de software, visando satisfazer ao conjunto enormemente variado de anseios por informatização, atualmente detectado na sociedade moderna;
- Como administrar as questões comportamentais, envolvendo os clientes e/ou usuários e a política, cultura e filosofia empresarial.

Apesar da enorme variedade de problemas que caracterizam a crise do software, engenheiros de software e gerentes de projetos para desenvolvimento de sistemas computacionais tendem a concentrar suas preocupações no seguinte aspecto: “A enorme imprecisão das estimativas de cronogramas e de custos de desenvolvimento” (Tabelas 2, 3, 4 e 5) [LEE02].

Tabela 2- Custos em projeto de software por fase de desenvolvimento [LEE02].

<i>Custos em projeto de software por Fase de Desenvolvimento</i>	
Etapa de Trabalho	%
Análise de Requisitos	3
Desenho	8
Programação	7
Testes	15
Manutenção	67

Tabela 3- Custos para correção de erros de software [LEE02].

Custos para Correção de Erros de Software				
Fase de desenvolvimento do software	% de Desvios (\$)	Erros Introduzidos (%)	Erros encontrados (%)	Custo Relativo para Correção
Análise de Requisitos	5	55	18	1,0
Desenho	25	30	10	1,0 - 1,5
Teste do Código e da Unidade	10			
Teste de Integração	50	10	50	1,0 - 5,0
Validação e Documentação	10			
Manutenção Operacional		5	22	10 – 100

Tabela 4- Excedentes de Custo [LEE02].

Excedentes de Custo	
% Excedente de Custo	% de Respostas
<20%	15,5%
21% - 50%	31,5%
51% - 100%	29,6%
101% - 200%	10,2%
201% - 400%	8,8%
>400%	4,4%

Tabela 5- Excedente de Prazo [LEE02].

Excedente de Prazo	
% Excedente de Prazo	% de Respostas
<20%	13,9%
21% - 50%	18,3%
51% - 100%	20,0%
101% - 200%	35,5%
201% - 400%	11,2%
>400%	1,1%

6. ANTICRISE DO SOFTWARE

Segundo Rezende [REZENDE99], pode-se resumir que a anticrise é a união e trabalho conjunto e harmonioso de três elementos: Empresa (Alta Administração), Cliente e/ou usuário e a unidade de informática (Desenvolvedores de soluções).

E na prática, cabe principalmente à unidade de informática aceitar este conceito e fazer o possível para a efetivação desta tese (Anticrise), utilizando-se de todos os recursos disponíveis para tal.

A Unidade de informática é um dos principais agentes de mudança nas organizações, preocupando-se com o negócio empresarial, auxiliando efetivamente os gestores nos processos de tomada de decisão, tanto operacionais, como gerenciais e estratégicas.

7. QUALIDADE DE SOFTWARE

Atingir um alto nível de qualidade de produto ou serviço é o objetivo da maioria das organizações. Atualmente não é mais aceitável entregar produtos com baixa qualidade e reparar os problemas e as deficiências depois que os produtos foram entregues ao cliente. [SOMMERVILLE03]

Segundo Machado [MACHADO01], para muitos engenheiros de software, a qualidade do processo de software é tão importante quanto à qualidade do produto. Assim na década de 90 houve uma grande preocupação com a modelagem e melhorias no processo de software. Abordagens importantes como as normas ISO 9000 e a ISO / IEC 12207, o modelo CMM (Capability Maturity Model) e o SPICE (Software Process Improvement and Capability dEtermination) sugerem que melhorando o processo de software, podemos melhorar a qualidade dos produtos.

A qualidade é consequência dos processos, das pessoas e da tecnologia. A relação entre a qualidade do produto e cada um desses fatores é complexa. Por isso, é muito mais difícil controlar o grau de qualidade do produto do que controlar os requisitos.[PÁDUA03]

Segundo Machado [MACHADO01], a globalização da economia vem influenciando as empresas produtoras e prestadoras de serviços de software a alcançar o patamar de qualidade e produtividade internacional para enfrentarem a competitividade cada vez maior.

Segundo o Ministério da Ciência e Tecnologia [MCT02], ainda que divulgadas na década de 90, o conhecimento e utilização das normas e modelos para qualidade de software, estão distantes de tornar-se realidade nas empresas desenvolvedoras de software, conforme (Tabelas 6, 7, 8 e 9).

Tabela 6- Conhecimento da Norma ISO/IEC 12207 [MCT02].

Conhecimento da Norma NBR ISO/IEC 12207 - Processos de Ciclo de Vida de Software -					
Categorias	Total	Micro	Pequena	Média	Grande
Conhece e usa sistematicamente	3,9	1,4	3,8	2,6	9,1
Conhece e começa a usar	8,3	6,1	6,8	7,7	14,8
Conhece, mas não usa	55,1	48,6	51,1	64,1	67,0
Não conhece	32,7	43,9	38,3	25,6	9,1

Tabela 7- Conhecimento da Norma ISO 9000 [MCT02].

Conhecimento das Normas ISO 9000 - Gestão da Qualidade					
Categorias	Total	Micro	Pequena	Média	Grande
Conhece e usa sistematicamente	19,4	3,4	16,3	34,1	44,0
Conhece e começa a usar	14,8	14,1	18,5	12,2	12,1
Conhece, mas não usa	52,4	62,4	50,4	43,9	41,8
Não conhece	13,4	20,1	14,8	9,8	2,2

Tabela 8- Conhecimento do Modelo CMM [MCT02].

Conhecimento do modelo CMM - Capability Maturity Model					
Categorias	Total	Micro	Pequena	Média	Grande
Conhece e usa sistematicamente	3,9	0,7	2,9	2,5	11,4
Conhece e começa a usar	17,1	3,4	20,4	30,0	29,5
Conhece, mas não usa	53,7	62,2	48,9	47,5	48,9
Não conhece	25,3	33,8	27,7	20,0	10,2

Tabela 9- Conhecimento do Projeto SPICE [MCT02].

Conhecimento do projeto SPICE - Software Process Improvement and Capability dEtermination (Technical Report ISO/IEC TR 15504)					
Categorias	Total	Micro	Pequena	Média	Grande
Conhece e usa sistematicamente	1,0	2,0	-	-	1,1
Conhece e começa a usar	3,2	1,4	3,0	5,1	5,7
Conhece, mas não usa	56,7	49,0	50,4	59,0	77,3
Não conhece	39,1	47,6	46,6	35,9	15,9

Prevê-se que na primeira década dos anos 2000, após ajustarem seus processos para a produção de software de qualidade dentro de prazos e orçamentos confiáveis, as organizações serão pressionadas por seus concorrentes a reduzir substancialmente os prazos para a entrega de produtos. Organizações que sejam capazes de integrar, harmonizar e acelerar seus processos de desenvolvimento e manutenção de software terão a primazia do mercado [MACHADO01].

Diante deste fato, podemos afirmar que por falta de utilização das normas ou modelos de qualidade de software, produzimos softwares de qualidade contestável e participando efetivamente da “Crise do Software” [PRESSMAN02].

7.1 Produto de Software

Quando entregamos a um cliente um pacote bem delimitado e identificado, podemos dizer que entregamos um produto [SPINOLA98].

A definição para produto de software segundo a norma IEEE-STD-610 [IEEE90] é:

“O conjunto completo, ou qualquer dos itens individuais do conjunto, de programas de computador, procedimentos, e documentação associada e dados designados para liberação para um cliente ou usuário final” [PAULK95].

7.2 Processo de Software

O conceito de processo de software se baseia no conceito generalizado de processo, que pode ser definido como uma seqüência de estados de um sistema que se transforma [SPINOLA98].

O SEI (*Software Engineering Institute*), da *Carnegie Melon University* propõe o seguinte [SEI]:

“Um processo é uma seqüência de passos realizados para um dado propósito. Colocado de maneira mais simples, processo é aquilo que você faz. Processo é aquilo que as pessoas fazem, usando procedimentos, métodos, ferramentas, e equipamentos, para transformar matéria prima (entradas) em produto (saída) que tenha valor para o cliente”. [PAULK95]

“O processo de software pode ser definido como um conjunto de atividades, métodos, práticas, e transformações que as pessoas empregam para desenvolver e manter software e os produtos associados (ex. planos de projeto, documentos de projeto (design), código, casos de teste, e manual do usuário)”. [PAULK95]

8. Gestão de Configuração de Software

Segundo Babich [BABICH86] gestão de configuração de software é:

“A arte de coordenar o desenvolvimento de software para minimizar a confusão é chamada de gestão de configuração. A gestão de configuração é a arte de identificar, organizar e controlar modificações de software que está sendo construído por uma equipe de programação. O objetivo é maximizar a produtividade pela minimização dos erros.”

Segundo Sommerville [SOMMERVILLE03] o gerenciamento de configuração (*configuration management – CM*) é o desenvolvimento e aplicação de padrões e procedimentos para gerenciar um produto de sistema em desenvolvimento. É necessário gerenciar os sistemas em desenvolvimento porque, à medida que eles se desenvolvem, são criadas muitas versões diferentes de software. Essas versões incorporam propostas de mudanças, correções de defeitos e adaptações para diferentes hardwares e sistemas operacionais. É possível que haja várias versões em desenvolvimento e em uso ao mesmo tempo. É necessário manter o controle das mudanças que foram implementadas e de como essas mudanças foram incluídas no software.

O processo de gerência de configuração é um processo de aplicação de procedimentos administrativos e técnicos, por todo o ciclo de vida de software, destinado a: identificar e definir os itens de software em um sistema, e estabelecer suas linhas básicas (baseline); controlar as modificações e liberações dos itens; registrar e apresentar a situação dos itens e dos pedidos de modificação; garantir a completeza, a consistência e a correção dos itens; e controlar o armazenamento, a manipulação e a distribuição dos itens.[ISO12207: 97]

Segundo Pressman [PRESSMAN02] a gestão de configuração de software (*software configuration management*, SCM) é uma atividade guarda-chuva que é aplicada ao longo de todo o processo de software. Como modificações podem ocorrer em qualquer época, as atividades de SCM são desenvolvidas para identificar as modificações, controlar modificações, garantir que as modificações sejam adequadamente implementadas e relatar as modificações a outros interessados.

As mudanças descontroladas em software geralmente levam ao caos e/ou crise de software.[REZENDE99]

Segundo Pádua [PADUA03] na engenharia de software, os artefatos que devem ser controlados incluem a documentação, os modelos e o próprio código (fontes e executável). Mesmo para projetos de porte muito pequeno, um mínimo de controle de versões é necessário para evitar o desperdício de trabalho. Com o controle de versões conservam-se versões antigas dos artefatos, que contém material que pode vir a ser novamente aproveitado, mas evita-se que versões mais antigas venham de forma inadvertida, a tomar o lugar de versões mais novas. Esse tipo de erro é comum mesmo em projetos individuais; e essa espécie de desperdício é facilmente evitada.

8.1 Gestão de Configuração de Software no CMMI

O CMMI – (*Capability Maturity Model Integrated*) foi desenvolvido pelo SEI (*Software Engineering Institute*), ligado à CMU (*Carnegie Mellon University*), em Pittsburgh, nos Estados Unidos. O desenvolvimento desse modelo foi financiado pelo DoD, Departamento de Defesa Americano, com o objetivo de se estabelecer um padrão de qualidade para software desenvolvido para as forças armadas.[PESSOA&SPINOLA]

O CMMI foi concebido para o desenvolvimento de grandes projetos militares e, para a sua aplicação em projetos menores e em outras áreas, é necessário um trabalho cuidadoso de interpretação e adequação à realidade da organização.[FIORINI98],[CMMI02]

No modelo CMMI foram estabelecidos níveis referentes à maturidade que a organização possui para desenvolver software: Inicial (1), gerenciado (2), definido (3), gerenciado Quantitativamente (4) e otimizado (5). [CMMI02]

Cada nível de maturidade está dividido em áreas-chave de processo chamadas de PA (*Process Area*), que estabelecem grandes temas a serem abordados, somando 18 áreas-chave.

No nível 2 do CMMI, os métodos de gerenciamento de software são documentados e acompanhados. Políticas organizacionais orientam os projetos estabelecendo processos de gerenciamento. Práticas bem-sucedidas podem ser repetidas em novos projetos. No nível 2 existe um sistema de gerenciamento implantado, que garante o cumprimento de custos e prazos em projetos similares já desenvolvidos anteriormente.[PESSOA&SPINOLA]

Entre as PA's do nível 2 do CMMI está presente o Gerenciamento da Configuração de Software. O propósito do Gerenciamento de Configuração é para estabelecer e manter a integridade de produtos usando a identificação de configuração, controle de configuração, condição de configuração, e auditorias de configuração.[CMMI02]

A área do processo de Gerenciamento de Configuração envolve : [CMMI02]

- ❖ Identificar a configuração de produtos selecionados que compõem as *baselines* em pontos dados no tempo;
- ❖ Controlar mudanças para artigos de configuração;
- ❖ Construir ou prover especificações dos produtos de software;
- ❖ Manter a integridade das *baselines*;
- ❖ Prover condição e dados precisos de configuração atuais para desenvolvedores, usuários finais, e clientes;

8.2 Gestão de Configuração na norma ISO 12207

A norma internacional NBR ISO/IEC 12207 – Tecnologia da Informação – Processos de Ciclo de Vida de Software [ISO12207: 97] é usada como referência em muitos países, inclusive no Brasil, para alcançar esse diferencial competitivo.

Ela tem por objetivo auxiliar os envolvidos na produção de software a definir seus papéis, por meio de processos bem definidos, e assim proporcionar às organizações que a utilizam um melhor entendimento das atividades a serem executadas nas operações que envolvem, de alguma forma, o software.

A arquitetura descrita na norma [MACHADO01] utiliza uma terminologia bem definida e é composta de: processos, atividades e tarefas para aquisição, fornecimento, desenvolvimento, operação, e manutenção do software.

Está estruturada em 3 processos de Ciclo de Vida: [ISO12207: 97]

- ❖ Processos Fundamentais
- ❖ Processos de Apoio
- ❖ Processos Organizacionais

A gestão de configuração de software está presente nos processos de apoio de ciclo de vida. Os processos de apoio auxiliam e contribuem para o sucesso e a qualidade do projeto de software.

Um processo de apoio é empregado e executado quando necessário para documentação, gerência de configuração, garantia da qualidade, processo de verificação, processo de validação, revisão conjunta, auditoria e resolução de problemas [MACHADO01].

A norma ISO 12207, descreve que o processo de Gerência de Configuração consiste em:

- ❖ Implementação do processo;
- ❖ Identificação da configuração;
- ❖ Controle da configuração
- ❖ Relato da situação da configuração
- ❖ Avaliação da configuração
- ❖ Gerência de Liberação e distribuição

8.3 Gestão de Configuração de Software na ISO 9000-3

A família ISO 9000 é composta de uma série de normas, e reconhece que existem 4 diferentes categorias genéricas de produtos e publicou diretrizes para implementação de sistemas da qualidade para cada uma destas categorias: Hardware: ISO 9004-1; Serviços: ISO 9004-2; Materiais Processados: ISO 9004-3; e Software: ISO 9000-3; [XAVIER01]

Devido as dificuldades específicas de interpretação de como implantar os requisitos da ISO 9001 ou 9002 em software, é fundamental o uso da ISO 9000-3 para auxiliar a implantação do sistema de gestão da qualidade. Esta dificuldade está relacionada com a terminologia usada na ISO 9001, muito voltada para hardware; usando a ISO 9000-3 esta barreira é eliminada. [XAVIER01]

As normas da série ISO 9000 foram expandidas com a edição das normas internacionais de série 10.000, que complementam estas e fornecem diretrizes para planos de qualidade, auditorias internas, qualificação de auditores, comprovação metrológica e manuais de qualidade. [ISO9000]

Nessa expansão foi criada uma norma específica para o Gerenciamento da Configuração. É a ISO 10007:95 - Gestão da qualidade. Diretrizes para gestão da configuração.

A norma ISO 10.007 fornece diretrizes para gestão da configuração. Esta é uma disciplina da gestão que é aplicada sobre o ciclo de vida de um produto, a fim de fornecer visibilidade e controle de suas funcionalidades e características físicas. As atividades descritas são uma forma de satisfazer certos requisitos encontrados em outras normas da série ISO 9000.

Na norma ISO 9000-3, o gerenciamento de configuração está inscrito no requisito Identificação e Rastreabilidade e consiste em: [ISO9000]

- ❖ Quando apropriado, a organização deve identificar o produto por meios adequados ao longo da realização do produto. A organização deve identificar a situação do produto no que se refere aos requisitos de monitoramento e de medição.
- ❖ Quando a rastreabilidade é um requisito, a organização deve controlar e registrar a identificação única do produto.

8.4 Gestão de Configuração de Software no projeto SPICE

A ISO/IEC 15504, SPICE (*Software Process Improvement and Capability dEtermination*), presta-se à realização de avaliações de processos de software com dois objetivos: a melhoria dos processos e a determinação da capacidade de processos de uma organização. [SALVIANO01]

Se o objetivo for à melhoria dos processos, a organização pode realizar a avaliação gerando um perfil dos processos que serão usados para a elaboração de um plano de melhorias. A organização deve definir os objetivos e o contexto, bem como escolher o modelo e o método para a avaliação e definir os objetivos de melhoria. [SALVIANO01]

No segundo caso, a empresa tem o objetivo de avaliar um fornecedor em potencial, obtendo o seu perfil de capacidade. Para isso ela deve definir os objetivos e o contexto da avaliação, os modelos e métodos de avaliação e os requisitos esperados. O perfil de capacidade permite ao contratante estimar o risco associado à contratação daquele fornecedor em potencial para auxiliar na tomada de decisão de contrata-lo ou não. [SALVIANO01]

A norma ISO 15504, foi baseada nos principais modelos existentes como CMMI, TRILLIUM e BOOTSTRAP, tendo entre objetivos básicos realizar uma harmonização destes diversos modelos existentes e prover um *framework* para estes modelos e outros que possam ser criados. [SALVIANO01]

Este modelo de referência descreve um conjunto de processos e boas práticas da Engenharia de Software consideradas universais e fundamentais. Quarenta processos e componentes de processos são descritos e organizados em cinco categorias de processo: Cliente-fornecedor, Engenharia, Suporte, Gerência e Organização. Estes processos são um superconjunto dos processos definidos na ISO / IEC 12207.

A gerência de Configuração está também presente no SPICE, na categoria SUPORTE denominado processo de Gerência de Configuração.

9. CONCLUSÃO

Todas as normas e modelos de qualidade para software têm por objetivo buscar organização e melhoria contínua no processo de desenvolvimento de software. A implementação da gestão de configuração de software está totalmente ligada a essas normas e modelos. Diante o quadro atual de baixa adoção das normas e modelos aqui no Brasil, segundo o MCT, citado neste trabalho, temos aqui identificado que a “Crise do Software” se justifica pela falta de metodologias e processos de controle e gerenciamento de configuração no desenvolvimento de software.

A situação ainda piora quando tratamos de pequenas e médias empresas, onde normalmente o desenvolvedor é sobrecarregado de tarefas e com prazos e custos extremamente inviáveis, a não implementação da gestão de configurações, compromete diretamente a qualidade do software.

Cabe ainda citar que a gestão de configuração de software está:

- Adequação aos princípios e objetivos da Engenharia de Software.
- Possibilita então ser fator crítico para o sucesso no desenvolvimento de software.
- Adaptabilidade na inexistência de alguns processos, dependendo do porte da empresa desenvolvedora de software.
- Com a sua adoção é possível contribuir para o fim da “Crise do Software”.
- Permite para micros, pequenas e médias empresas que possuem recursos humanos e financeiros limitados à possibilidade de competir com as empresas de maior porte.
- Com os processos de desenvolvimento de software controlados, documentados e gerenciados o pequeno desenvolvedor poderá assumir projetos de alta complexidade, aliados a técnica e criatividade, pois terá mais chance de sucesso.
- Melhor capacitado e provedor de metodologias que levam ao desenvolvimento de software com qualidade, o desenvolvedor poderá criar soluções que atendam as necessidades e os requisitos da empresa, contribuindo para criação de vantagens competitivas, sustentando as bases estratégicas das organizações.

10. REFERENCIAS BIBLIOGRÁFICAS

[BABICH86] BABICK, W.A., *Software Configuration Management*, Addison-Wesley, 1986

[CMMI02] CMMI, SEI, *Software Engineering Institute, Carnegie Mellon University*, <http://www.sei.cmu.edu>.

[FIORINI98] FIORINI, SOELI T., et al. *Engenharia de Software com CMM*, Rio de Janeiro, Ed. Brasport, 1998.

[IEEE90] IEEE STD. 610 12-1990, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE, Piscataway, NJ, 1997.

[ISO9000] NBR ISO/IEC 9001:2000, 9000-3 – *Software* - Rio de Janeiro, ABNT – Associação Brasileira de Normas Técnicas.

[ISO12207: 97] NBR ISO/IEC 12207:1997, *Tecnologia de Informação – Processos de Ciclo de Vida de Software*, Rio de Janeiro, ABNT – Associação Brasileira de Normas Técnicas.

- [LEE02] LEE, RICHARD C. e TEPFENHART, WILLIAM M., *UML e C++ - Guia de desenvolvimento orientado a objeto*, São Paulo, Ed. Makron Books, 2002.
- [MACHADO01] MACHADO, CRISTINA ÂNGELA FILIPAK in WEBER, KIVAL CHAVES, et al. *Qualidade e Produtividade em Software*, São Paulo, Ed. Makron Books, 2001.
- [MAFFEO92] MAFFEO, BRUNO, *Engenharia de Software e Especificação de Sistemas*, Rio de Janeiro, Ed. Campus, 1992.
- [MARTIN91] MARTIN, JAMES, *Engenharia da Informação*, Rio de Janeiro, Ed. Campus, 1991.
- [MCT02] MINISTÉRIO DA CIÊNCIA E TECNOLOGIA, Secretaria de Política de Informática, *Qualidade e Produtividade no Setor de Software Brasileiro*, Brasília, N.4, 2002.
- [PÁDUA03] FILHO, WILSON DE PÁDUA PAULA, *Engenharia de Software*, Rio de Janeiro, Ed. LTC, 2003.
- [PAULK95] PAULK, M.C. et al. *The Capatibility Maturity Model – Guidelines for improving the software process*, Addison Wesley, SEI series, 1995.
- [PESSOA&SPINOLA] PESSOA, MARCELO E SPINOLA, MAURO in WEBER, KIVAL CHAVES, et al. *Qualidade e Produtividade em Software*, São Paulo, Ed. Makron Books, 2001.
- [PRESSMAN95] PRESSMAN, ROGER S., *Engenharia de Software*, São Paulo, Ed. Makron Books, 1995.
- [PRESSMAN02] PRESSMAN, ROGER S., *Engenharia de Software*, Rio de Janeiro, Ed. McGraw-Hill, 2002.
- [REZENDE99] REZENDE, DENIS ALCIDES, *Engenharia de Software e Sistemas de Informações*, Rio de Janeiro, Ed. Brasport, 1999.
- [SALVIANO01] SALVIANO, CLENIO, in WEBER, KIVAL CHAVES, et al. *Qualidade e Produtividade em Software*, São Paulo, Ed. Makron Books, 2001.
- [SEI] SEI, *Software Engineering Institute, Carnegie Melon University*, <http://www.sei.cmu.edu>.
- [SOMMERVILLE03] SOMMERVILLE, IAN, *Engenharia de Software*, São Paulo, Ed. Pearson Education, 2003.
- [SPINOLA98] SPINOLA, MAURO DE MESQUITA, *Diretrizes para o desenvolvimento de software de sistemas embutidos*, Tese de Doutorado, USP - São Paulo, 1998.
- [XAVIER01] XAVIER, JORGE HERCULES, in WEBER, KIVAL CHAVES, et al. *Qualidade e Produtividade em Software*, São Paulo, Ed. Makron Books, 2001.