**Centro Universitário UniEvangélica**

**Anápolis 18 de fevereiro de 2018.**

**Acadêmico: Éber Lucas de França Oliveira**

**Professora: Renata Dutra Braga**

**Disciplina: Gerência de Configuração de Software**

**Período: 7° período**

**Fichamento do Artigo Gerência de Configuração de Software solicitado pela disciplina de Gerência de Configuração de Software.**

**Gerência de Configuração**

1. **Desenvolvendo software de forma eficiente e disciplinada**

**1.1** Atualmente os sistemas de software não param de evoluir, tanto em tamanho quanto em complexidade. Devido a esse crescimento é notável que a gerência de várias partes do processo também se desenvolveria.

**1.2** Prova de que a gerência é importante pois modificações em artefatos existentes, chega a consumir 75% do custo total do seu ciclo de vida.

**1.3** E aproximadamente, 20% de todo o esforço de manutenção é gasto em erros de implementação e os outros 80% são usados na adaptação do software em função de modificações em requisitos funcionais, regras de negócios e na reengenharia da aplicação.

**1.4** Gerência de Configuração de Software surgiu da necessidade de se controlar estas modificações, por meio de métodos e ferramentas e processos, com o intuito de maximizar a produtividade e diminuir os erros posteriores.

**1.5** O uso de processos de Gerência de Configuração é bastante importante para que se possa obter maior controle sobre os artefatos produzidos e modificados em todas etapas do desenvolvimento.

**1.6** O motivo da sua importância é geralmente associado às dificuldade qando a Gerência de Configuração não é utilizada no desenvolvimento de software.

**1.7** Pelo ponto de vista do desenvolvimento, a Gerência de Configuração de Software abrange três sistemas principais:

* Controle de modificações;
* Controle de versões e
* Controle de gerenciamento de construção.

**2. Terminologia**

**2.1** O sistema de controle de versões permite que os artefatos sob Gerência de Configuração se desenvolvam de forma distribuída, concorrente e disciplinada, evitando perdas ou sobreposições durante o desenvolvimento e a manutenção do artefato.

**2.2** O sistema de controle de modificações grava todas as informações geradas durante o desenvolvimento das solicitações de modificação e relata essas informações aos participantes e usuários autorizados.

**2.3** Existem inúmeras vantagens em utilizar Gerência de Configuração de Software:Algumas delas são:

* (1) ganho de produtividade e eficiência;
* (2) diminuição do retrabalho e dos erros;
* (3) aumento da disciplina no processo de desenvolvimento;
* (4) aumento da memória organizacional;
* (5) acesso às informações qualitativas e quantitativas referentes ao processo de desenvolvimento, como por exemplo, medida de esforço para efetuar uma alteração e freqüência de modificações por componente;
* (6) possibilidade de estabelecer uma trilha de auditoria indicando por que, quando e por quem um artefato foi alterado;
* (7) auxílio à gerência de projetos e
* (8) garantia de ambiente estável no qual o produto deve ser desenvolvido.

**2.4** O sistema de controle de versões permite que os artefatos sejam obtidos, por meio de uma operação conhecida como check-out, modificados e, depois, retornados ao repositório, através de uma operação conhecida como check-in.

**2.5** A cada operação de check-in realizada, a versão do item é incrementada de uma unidade. Quando o item é adicionado pela primeira vez no repositório, este item passa a ter a versão igual a 1. Para cada item de configuração armazenado, são anexadas informações como: datas da criação ou alteração, comentários e versões.

**2.6** As ferramentas de controle de versões normalmente suportam a definição de diferentes políticas de trabalho. Dentre essas políticas, podemos citar a política pessimista, que enfatiza o uso de check-out reservado, fazendo bloqueio e inibindo o paralelismo do desenvolvimento sobre o mesmo artefato.

**2.7** Outra política bastante utilizada é a otimista. Desta forma, se um artefato for alterado simultaneamente por dois desenvolvedores, a política assume que a quantidade de conflitos será naturalmente baixa e que será mais fácil tratar cada conflito individualmente, caso eles venham a ocorrer.

**2.8** A política otimista usa um mecanismo conhecido como junção (merge), que une as modificações efetuadas em paralelo sobre um mesmo artefato e produz uma nova versão deste artefato que contém a soma das modificações.

**2.9** A Gerência de Configuração de Software também permite que a implementação de novas funcionalidades por uma equipe seja realizada em paralelo, mas de forma isolada e independente das modificações de outros desenvolvedores.

**3. Estratégias de Organização**

**3.1** Algumas estratégias podem ser adotadas para melhorar a organização do trabalho dos desenvolvedores no projeto. De forma genérica, criar um ramo é um meio de organizar o trabalho, pois isola o desenvolvimento de outras modificações e possibilita que o trabalho seja executado sem que uma modificação específica cause impacto nas demais alterações do software.

**3.2** Estratégias de ramificação podem ser utilizadas, durante o desenvolvimento, em situações específicas. Algumas podem ser combinadas, outras são mutuamente exclusivas. Algumas estratégias de são:

* **[Manutenção caótica]** Estratégia onde não existe isolamento e, portanto, a impossibilidade de separar o que é manutenção corretiva e evolutiva. Na mainline ocorre a evolução e a correção do software. Não existem ramificações.
* **[Manutenção em série]** Estratégia de organização que separa as evoluções das correções no software. Esta estratégia pode ser usada quando uma release do produto será entregue para a homologação, que é a fase onde testes serão executados e os erros corrigidos. Neste caso, na mainline só ocorre a evolução do produto, enquanto o ramo fica destinado às correções. O ramo é temporário e as junções acontecem no sentido do ramo para a mainline.

**4. Gerência de Configuração e Desenvolvimento de Software**

**4.1** Por ser uma área fortemente relacionada ao controle, a Gerência de Configuração é referenciada em diversas normas, processos, procedimentos, políticas e padrões, como ISO 12207, CMMI e MPS.Br.

**4.2** Do ponto de vista gerencial, o processo de Gerência de Configuração de Software é dividido em cinco funções:

* Identificação da configuração;
* Controle da configuração;
* Acompanhamento da situação da configuração e,
* Auditoria da configuração e gerenciamento de entrega.

**5. Conclusão**

**5.1** O artigo nos nos mostra as diferentes visões em relação a Gerência de Configuração e que tem um papel importante no processo de desenvolvimento.

Na perspectiva gerencial, a Gerência de Configuração é dividida em cinco funções, que são: identificação da configuração, controle da configuração, contabilização da situação da configuração, auditoria da configuração e gerenciamento de liberação e entrega. Sob a perspectiva de desenvolvimento, a Gerência de Configuração é dividida em três sistemas principais: controle de modificações, controle de versões e gerenciamento de construção. As cinco funções podem ser implementadas pelos três sistemas descritos na perspectiva de desenvolvimento, acrescidos de procedimentos manuais quando necessário.

Este artigo também introduziu a Gerência de Configuração, informando os benefícios que esta área provê, além de conceitos e estratégias que podem ser utilizadas para desenvolver software de forma produtiva, eficiente e disciplinada com uso de ferramentas.