# Centro Universitário de Anápolis - UniEVANGÉLICA

Av. Universitária Km 3,5 Cx. Postal 122 e 901

CEP: 75 070-290 Anápolis/GO

Fones: (062) 3310 – 6658 e 3310 – 6690

# Acadêmicos:

**Felipe Ferreira;**

**Marcus Daniel**

**Pedro Henrique;**

**Raif André.**

**Gerência de Configuração de Software**

# AULA 2 ATIVIDADES COMPLEMENTARES

1. **Para você, o que é gerência de configuração de software?**

É um conjunto de atividades que servem de apoio as mudanças relacionadas ao desenvolvimento de software, mantendo a integridade e a estabilidade durante a evolução do projeto.

# O gerenciamento de configurações é o gerenciamento de um sistema de software em constante evolução. Neste contexto, quais problemas são ocasionados com a não adoção das políticas e processos efetivos da gerência de configuração de software?

Perca de versões anteriores de arquivos do projeto; impossibilidade de ter diferentes versões do sistema rodando ao mesmo tempo; códigos que são sobrescritos por acidente e são perdidos; dificuldade de manter registro de alterações e autores.

# Cite, pelo menos, quatro benefícios do uso da gerência de configuração de software.

* + Proporcionar estabilidade e confiabilidade
  + Aumentar a visibilidade do sistema
  + Identificar e corrigir problemas com maior rapidez
  + Apoiar a tomada de decisões

# Estabeleça e descreva uma solução para o problema de “alteração de um mesmo arquivo por dois desenvolvedores”. Indique as vantagens e desvantagens da solução.

Para tal situação e desejável a utilização do “*conflito(****merge***)” Assim a minha branch remota volta para o commit desejado e quando eu clonar o projeto vou começar a partir desse commit.

# Por que a Primeira Lei da Engenharia de Sistemas é verdadeira? Forneça exemplos específicos para, pelo menos, duas das quatro razões fundamentais para as alterações.

Porque Independente de onde se esteja no ciclo de vida de um sistema, o sistema vai se modificar e o desejo de modificá-lo vai persistir ao longo de todo o ciclo de vida.

Ex. 1: Novas condições de negócio e/ou mercado, modificam regras de negócio. Uma lei federal que mude a forma de pagamento de fatura de cartão de crédito, faz com que empresas de cartões de crédito tenham que modificar seus sistemas para se adequarem à nova regra.

Ex. 2: Novas necessidades do cliente exigem modificações de funcionalidades. O negócio de um cliente assistido pelo sistema, muda a forma de oferecer o produto a seus consumidores finais, o que obriga a modificação de funcionalidades do sistema.

# Os principais processos de gerenciamento de configurações estão interessados no gerenciamento de mudanças, gerenciamento de versões, na construção de sistema e no gerenciamento de releases. Descreva de maneira exemplificada as etapas do processo da gerência de configuração de software.

Controle e Acompanhamento de Mudanças: Mudanças aparecem durante todo o desenvolvimento e devem ser registradas, avaliadas e agrupadas de acordo com sua prioridade. Com base nessas informações, é possível planejar melhor o escopo, prazo e o custo de cada iteração. Em seguida, à medida que o desenvolvimento acontece, pode-se acompanhar o estado da solicitação da mudança até sua implementação e até o lançamento de uma versão em produção.

Registro da Evolução do Projeto: Cada vez que uma solicitação de mudança é implementada, acontece um incremento na evolução do projeto que deve ser registrado no histórico. Este incremento na evolução corresponde a uma configuração. O controle de versão é a parte principal da GCS. É o elo comum entre o controle de mudança e a integração do projeto.

Verificação da Integridade do Sistema: O objetivo da integração é verificar se a construção do sistema a partir dos itens registrados em uma configuração é bem sucedida. Em termos práticos, a integração é feita através de scripts que automatizam a construção, testes e e também a coleta de métricas de qualidade. As ferramentas de integração contínua acompanham o controle de versão e disparam os scripts cada vez que uma nova configuração é registrada.

# Faça uma síntese sobre o artigo científico lido: “Gerência de configuração de software”.

Modificações em artefatos existentes de um projeto podem consumir até 75% do custo total do ciclo de vida. A Gerência de Configuração de Software surgiu da necessidade de controlar essas modificações. Essa possui métodos e ferramentas para maximizar a produtividade e minimizar os erros cometidos na evolução do projeto.

A codificação do projeto por vários desenvolvedores podem causar conflitos, por esses estarem, ao mesmo tempo, modificando um mesmo arquivo. O último desenvolvedor a mexer no arquivo pode sobrescrever os códigos escritos por outros desenvolvedores, e aqueles seriam perdidos. Manter o projeto em um repositório resolve esse problema, pois, cada desenvolvedor copia os arquivos do repositório e os edita em sua estação de trabalho própria; depois os devolve ao repositório. Dessa forma, o repositório sempre tem uma versão funcionando do projeto, enquanto cada desenvolvedor adiciona a sua contribuição sem destruir a contribuição de seus companheiros.

Durante o desenvolvimento do projeto, os artefatos enviados ao repositório são guardados versionados; ao final de cada iteração, tem-se um conjunto de artefatos formalmente aprovados; esses artefatos servem de base para a próxima iteração, e seu conjunto é denominado “baseline”. Quando uma “baseline” é entregue ao cliente, esta é denominada “release” – esta é a diferença sutil entre “baseline” e “release”. A identificação de “baselines” e “releases” no repositório é feita por “tags”.

# Estudo de caso (planejamento dos itens de configuração):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome do projeto** | **Item de configuração (IC)** | **Tipo de (IC)** | **Versão do (IC)** | **Nome completo do (IC)** |
| **Academic** | **Desenvolvimento Software** | **Artefato** | **1.0** | **José Francisco** |
| **Academic** | **Documento especificação de requisitos** | **Artefato** | **1.1** | **Newton Rodrigues** |
| **Academic** | **Caso de uso** | **Artefato** | **1.2** | **Melquisedeque** |
| **Academic** | **Diagrama de classe.** | **Artefato** | **1.1** | **Jônatas Gabriel** |