

## Aula 4

### Gerência de Configuração e Evolução

Profª Adriana Costa

### Conversa Inicial

#### Controle de versionamento dos itens de configuração

- As mudanças são inerentes a qualquer software, pois as necessidades dos clientes mudam, as tecnologias evoluem e as tendências dos mercados precisam ser acompanhadas para que a competitividade cresça e resulte em ganhos financeiros melhores

#### Controle de versionamento dos itens de configuração

- Como as mudanças são inevitáveis e precisam ser gerenciadas adequadamente, é importante contar com uma ferramenta automatizada para controlar o versionamento da evolução dos itens de configuração, de forma a inserir mais organização e segurança na construção e manutenção dos softwares

- Esta aula está dividida em cinco temas principais, sendo eles:
  - Artefatos de projeto
  - Versionamento dos itens de configuração
  - Sistemas de controle de versão
  - Configuração de ambientes, baselines e release
  - Gerência de mudança de arquivos

### Artefatos de Projetos

### Artefatos de projeto

- A construção de um software envolve o entendimento do que precisa ser feito e a tradução das necessidades dos usuários em uma solução técnica. O entendimento sobre os requisitos precisa ser documentado, além de documentar a solução técnica projetada

### Artefatos de projeto

- Mas a documentação de um software não deve ser feita sem um motivo claro e plausível. Documentar apenas por documentar não agrega valor, muito pelo contrário, gera prejuízo financeiro e de tempo

### Artefatos de projeto

- Artefato é todo produto de trabalho gerado ao longo do desenvolvimento do projeto, tanto como apoio para a equipe que vai construir o software como para documentar o que está sendo construído

### Artefatos de projeto

Artefato	Item de Configuração	Justificativa
Plano de Projeto	Não	Artefato alterado constantemente, sem impacto para os demais artefatos do projeto
Planilha de Risco	Sim	Artefato que serve de insumo para o planejamento de custo, prazo e tarefas do projeto
Requisitos	Sim	Artefato base para todo o projeto. Todos os demais artefatos dependem da integridade dele
Modelo de Classe	Sim	Artefato base para a construção do software, alterações afetam diretamente a construção no código
Modelo de Banco de Dados	Sim	Artefato base para a construção do software, alterações afetam diretamente a construção no código
Código	Sim	Artefato base para o funcionamento do software. Precisa ser controlado para evitar defeitos e retrabalho
Casos de Teste	Não	Artefato que não impacta outros artefatos do software

Fonte: Costa, 2022

### Versionamento dos Itens de Configuração

### Versionamento dos itens de configuração

- Os artefatos críticos de um projeto são fortes candidatos a serem tratados como itens de configuração, controlados por meio do processo de gerência de configuração
- O MPS.Br indica que é preciso definir critérios para selecionar os itens de configuração

### Subprocessos da gerência de configuração

- Registro da evolução do projeto, também conhecido como controle de versão. É o subprocesso que controla cada solicitação de mudança que é solicitada e implementada, para registrar o histórico do incremento do software

### Subprocessos da gerência de configuração

- Controle e acompanhamento de mudanças, também conhecido como controle de mudança. É o subprocesso responsável por registrar, avaliar e organizar todas as mudanças em um software de acordo com sua prioridade e necessidade para o negócio

### Subprocessos da gerência de configuração

- Estabelecimento da integridade do sistema – é o subprocesso responsável pela integração contínua e tem como objetivo verificar, conforme o software vai sendo construído, se os itens de configuração estão corretos e estão se integrando de maneira adequada aos demais itens de configuração, para gerar um executável que atenda às necessidades dos usuários

### Integração contínua

- É a ação de integrar os componentes criados em um software continuamente para que o processo ocorra de maneira gradual, facilitando a organização do projeto, a identificação e a correção de defeitos

### Integração contínua

- É uma prática de desenvolvimento de software na qual os desenvolvedores, em espaços de tempo curtos e constantes, juntam suas alterações de código em um repositório central. A frequência de integração deve ser definida pela equipe de desenvolvimento, dependendo da criticidade, do tamanho e da complexidade do software

### Integração contínua

- Em termos práticos, a integração contínua é feita por meio de scripts que automatizam o processo de integração dos componentes que estão sendo construídos e disponibilizados para a fase de teste do ciclo de vida de desenvolvimento, por meio da criação de uma versão compacta do software

### **Integração contínua**

- As ferramentas utilizadas para realizar a integração contínua acompanham as ferramentas de controle de versão e disparam os scripts cada vez que uma nova configuração é registrada, ou em horários programados, dependendo da definição dada pela equipe de desenvolvimento

### **Integração contínua**

- É preciso ter disciplina para manter a integração dos componentes que vão sendo construídos de maneira contínua e em intervalos regulares

### **Sistemas de Controle de Versão**

### **Sistema de controle de versão**

- O controle de versão é uma forma de controlar os componentes de software, quando são criados e quando são alterados, mantendo o histórico de tudo que ocorre ao longo do tempo

### **Problemas que podem ocorrer**

- Sobrescrever o código por acidente
- Perder o controle de quais alterações já foram efetuadas, de quando foram feitas e de quem as fez
- Dificuldade para recuperar o código de uma versão anterior que está em produção
- Problemas para manter variações do sistema ou manutenção em paralelo

### **Funcionamento do controle de versão**

- O repositório armazena o histórico de evolução do projeto, registrando toda e qualquer alteração feita em cada um dos itens de configuração versionados. Ele deve ser único e centralizado para todo o projeto, com a equipe de desenvolvimento tendo acesso para manter o controle de versão contínuo, de acordo com a periodicidade acordada entre a equipe

### Funcionamento do controle de versão

- Para manter a organização e a integridade do código, o desenvolvedor usa uma área de trabalho que contém a cópia dos arquivos do projeto e que é monitorada a fim de se identificar as mudanças realizadas

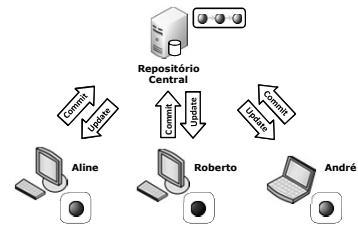
### Funcionamento do controle de versão

- Os comandos de commit (acesso) ou update (atualização) são executados pelo desenvolvedor, tanto para buscar um arquivo do repositório e alterá-lo na sua área de trabalho local quanto para salvar sua atualização no repositório
- A área de trabalho fica localizada, normalmente, na máquina de trabalho do desenvolvedor

### Controle de versão centralizado

- Segue a topologia em estrela, ou seja, é organizado apenas em um único repositório central e com várias cópias de trabalho, uma para cada desenvolvedor, em suas áreas de trabalho ou máquinas próprias

### Controle de versão centralizado

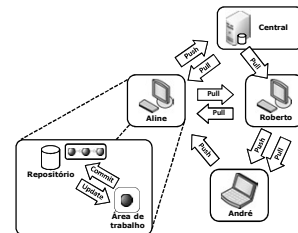


Fonte: <https://blog.pronus.io/posts/controle-de-versao/conceitos-basicos-de-controle-de-versao-de-software-centralizado-e-distribuido/>

### Controle de versão distribuído

- Organizado por meio de repositórios independentes, um para cada desenvolvedor, com área de trabalho acoplada. O relacionamento entre eles é feito por meio das operações de commit e update
- A comunicação entre os repositórios independentes e o repositório central ocorre por meio dos comandos de pull (puxar, ou ler) e push (empurrar, ou atualizar)

### Controle de versão distribuído



Fonte: <https://blog.pronus.io/posts/controle-de-versao/conceitos-basicos-de-controle-de-versao-de-software-centralizado-e-distribuido/>

## Configuração de Ambientes, Baselines e Release

### Baseline e release

- **Baseline** é a organização lógica de itens de configuração estáveis e aprovados. É um conceito utilizado para evoluir um conjunto de componentes ao longo das fases do desenvolvimento
- **Release** é uma entrega formal feita ao cliente, no final de uma iteração. É o conjunto de componentes integrados, testados e prontos para serem liberados para o cliente

### Branch ou ramificação

- No contexto da gerência de configuração, um ramo é uma linha diferente na evolução do software, que forma uma variação isolada e controlada do projeto. Ou seja, um ramo é uma outra linha de desenvolvimento do código, que, quando pronta, poderá ser integrada à linha principal do código do software

### Tipos de ramificações

- **Principal:** é o mais importante de um projeto, utilizado para armazenar e controlar o código principal e integrado do software
- **Dedicado:** usado para isolar as implementações complexas, demoradas ou incertas que não podem ser feitas diretamente no ramo principal e que geram riscos para a estabilidade do software

### Tipos de ramificações

- **Manutenção:** corresponde a uma versão lançada em produção que precisa ser corrigida, de forma pontual, sem impactar o conjunto das demais funcionalidades em desenvolvimento

## Gerência de Mudança de Arquivos

### **Boas práticas em mudança de arquivos**

- Identificar o requisitante da mudança
- Identificar o motivo para realizar a mudança
- Identificar qual será o retorno esperado após a implantação da mudança
- Identificar os riscos envolvidos na mudança

### **Boas práticas em mudança de arquivos**

- Identificar as necessidades de recursos para o desenvolvimento e a entrega da mudança
- Identificar quem é o responsável pela construção, pelos testes e pela implementação da mudança
- Identificar o impacto da mudança nos demais softwares e áreas da empresa