



Princípios e Práticas em Arquitetura de Software

Capítulo 6. Estratégias de Gestão de Configuração e Versionamento

Prof. Paulo Nascimento



Aula 6.1. Motivação e Ferramentas



Nesta aula

- ❑ Motivação.
- ❑ Ferramentas – SVN.

Por que nós precisamos de controle de versão?

(Obs. Version Control = Revision Control = Source Controle)

Motivação

- Backup e Restore
 - Arquivos são salvos a medida em que são editados.
- Sincronização
 - Os arquivos podem ser compartilhados entre várias pessoas e elas ainda estarem com a ultima versão disponível.
- Undo (Desfazer)
 - Em caso de “bagunça”, é possível retornar para a última versão estável a qualquer momento.

Motivação

- Rastrear Mudanças (Track Changes)
 - É possível identificar o motivo das alterações em cada arquivo.
- Sandboxing
 - Em caso de grandes mudanças, é possível fazer à parte e de forma controlada.
- Branch e Merge
 - É possível isolar os arquivos, fazer alterações, testar e, somente após isso, reintegrar à versão estável.

Ferramentas



CVS

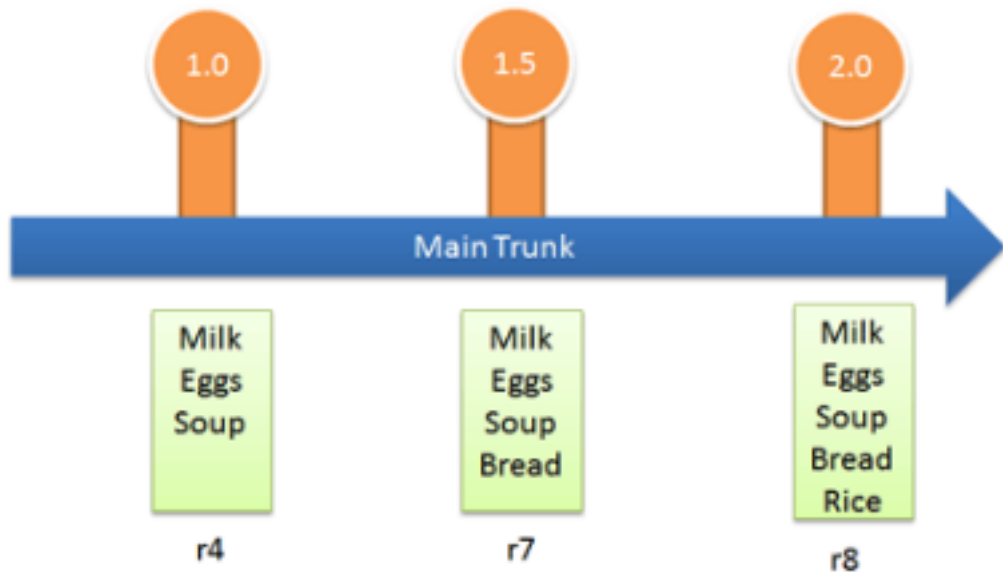
- Desde 80s, popular entre desenvolvedores e liberada sob a licença GNU.
- PRÓS:
 - É utilizado há muitos anos e considerado uma tecnologia madura.
- CONTRAS
 - Mover/Renomear arquivos não é considerado alteração de versão.
 - Não suporta operações atômicas.
 - Operações de branch são custosas (tempo).



Conclusão

✓ BENEFÍCIOS

- ✓ Backup e Restore;
- ✓ Sincronização;
- ✓ Undo (Desfazer);
- ✓ Rastrear Mudanças;
- ✓ Sandboxing;
- ✓ Branch e Merge.



Próxima aula

☐ Ferramentas

- ☐ SVN.

- ☐ GIT.

- ☐ Mercurial.



Aula 6.2. Ferramentas



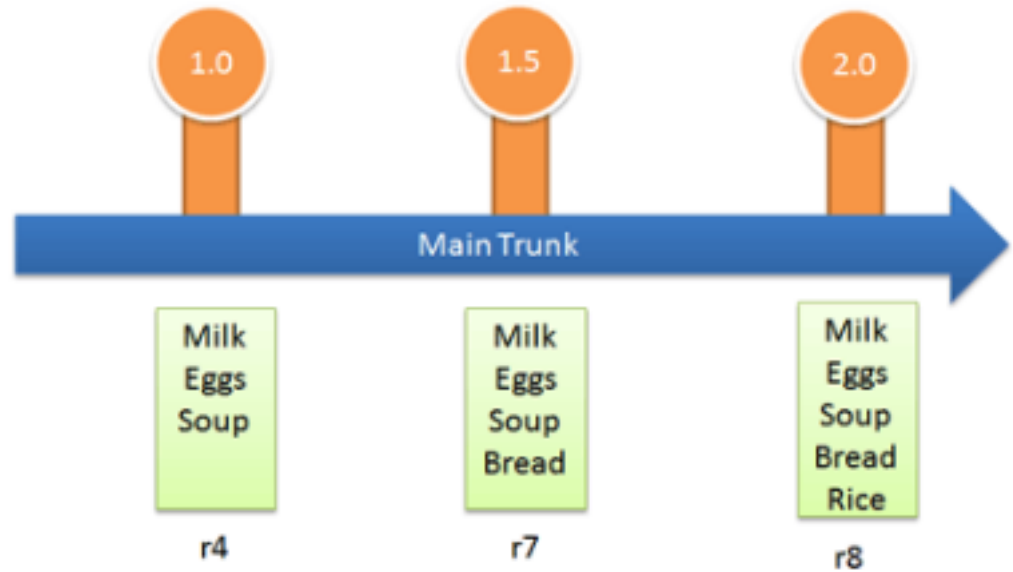
Nesta aula

- ❑ Ferramenta – SVN.
- ❑ Ferramenta – Git.
- ❑ Ferramenta – Mercurial.

Revisão

❑ BENEFÍCIOS

- ❑ Backup e Restore;
- ❑ Sincronização;
- ❑ Undo (Desfazer);
- ❑ Rastrear Mudanças;
- ❑ Sandboxing;
- ❑ Branch e Merge.



- Foi criado como alternativa ao CVS (Bugs e Compatibilidade).
- Free e open source, mas distribuído sob Apache License.
- PRÓS:
 - Inclui operações atômicas.
 - Operação de branch menos custosas.
 - Alta variedade de plugins para IDEs.
- CONTRAS:
 - Ainda contém bugs relacionados às alterações de nomes.
 - Poucos comandos de gestão de repositório.
 - Velocidade mais lenta.

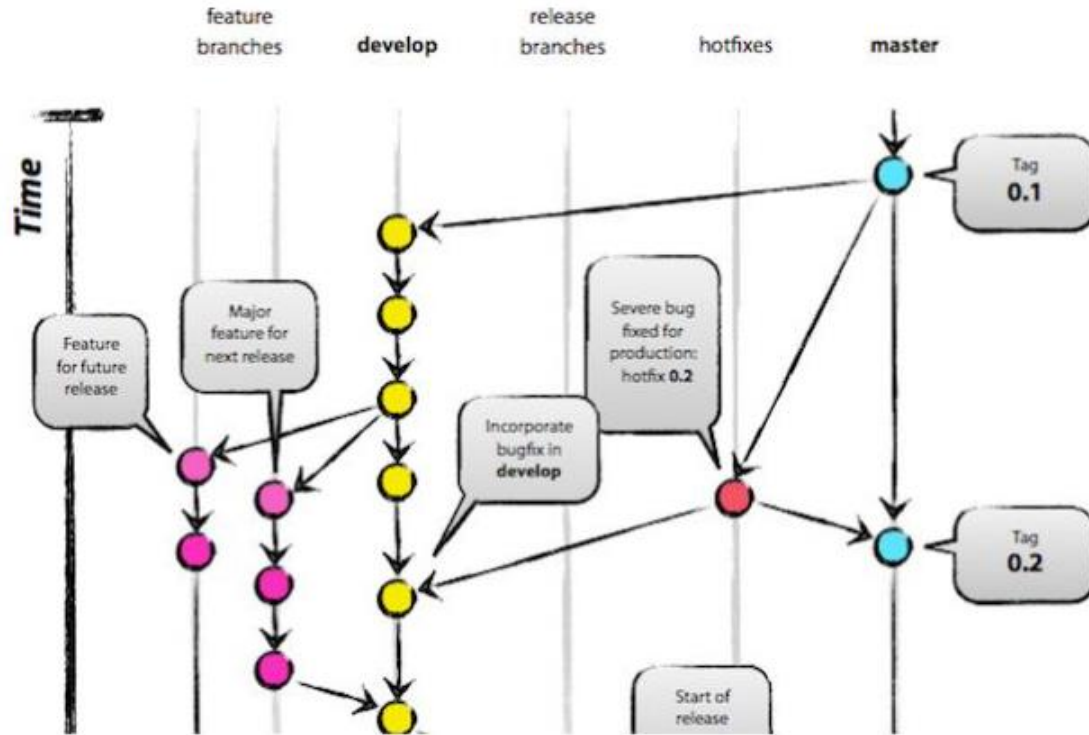


- Inicialmente desenvolvido por Linus Torvalds.
- Difere fortemente do CVS e SVN.
- Ideia original é ser rápido e permitir o controle distribuído de código.
- PRÓS:
 - Operações mais rápidas.
 - Operações de branch mais baratas.
 - Histórico disponível off-line.
- CONTRAS:
 - Curva de aprendizado.
 - Suporte Windows limitado comparado ao Linux.



- Também é um sistema distribuído.
- “Competidor do GIT”.
- Utilizado pelo OpeeOffice.org e implementado em Python.
- PRÓS:
 - Mais fácil de aprender do que o GIT.
 - Boa documentação.
 - Modelo distribuído.
- CONTRA:
 - Sem merge de dois parents.
 - Mais baseado em extensões (not scripts).

Conclusão



Próxima aula

❑ Requisitos Arquiteturais.