

TRABALHO TEÓRICO SOBRE GERÊNCIA DE CONFIGURAÇÃO

Mateus Carvalho Gonçalves

Turma 10A

Nº de Matrícula 201810245

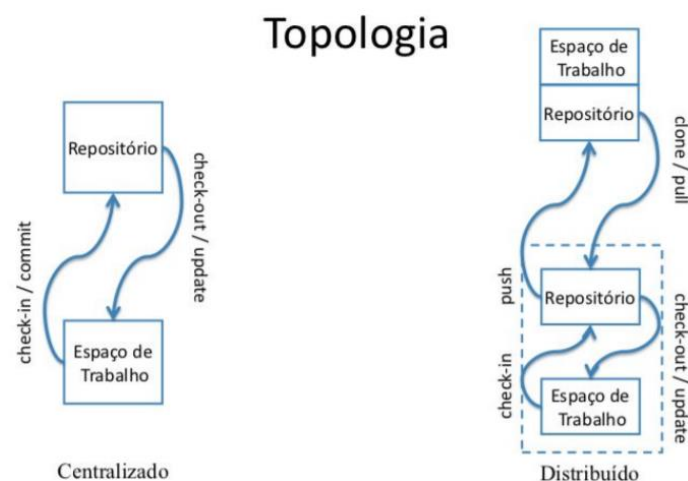
1 – Controle de modificações é a análise de riscos e impactos de mudanças que devem ser feitas em itens de configuração para avaliar a viabilidade e gerenciar o que deve ser feito e quando. O controle de versões é o gerenciamento do desenvolvimento das mudanças analisadas no processo anterior, fazendo decisões de branches e merges por exemplo (codelines). O controle de configuração é um controle de versão do sistema, ou seja, a escolha de versões estáveis de itens e ferramentas auxiliares para lançar uma versão estável do sistema (baselines).

2 – Item de configuração é cada parte única do sistema que é colocado sob controle de configuração, ou seja, que pode ser trabalhado para se criar versões e juntado novamente ao sistema.

3 – O item derivado é um item de configuração que pode ser obtido a partir de outro, chamado item fonte.

4 – Granularidade pode ser entendida como o quão atômico é o item de configuração. Ela é mais fina quanto menos itens derivados tiver e mais grossa caso contrário.

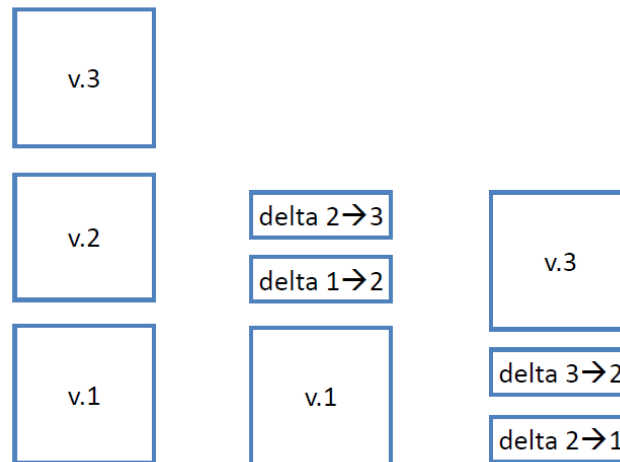
5 – A topologia do controle de versão pode ser centralizada ou distribuída. No primeiro caso não há branches no desenvolvimento, tornando-o linear. Já a topologia distribuída possui vários branches e espaços de trabalho, descentralizando o código e dando mais flexibilidade de gerenciamento de desenvolvedores.



Existem 3 tipos de armazenamento: Completo, Forward e Reverse.

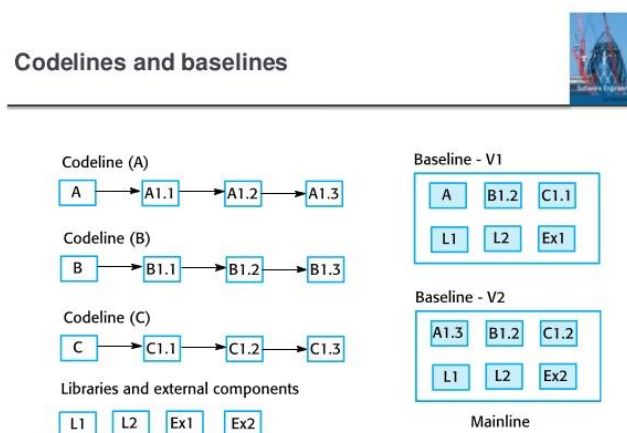
- O Completo salva no repositório todos os arquivos a cada versão modificada. Tem como vantagem não necessitar fazer merges para se obter uma versão específica, porém o gasto de armazenamento é muito grande e em projetos maiores se torna inviável.

- Forward salva a primeira versão e a partir da segunda salva os deltas, ou seja, apenas as mudanças feitas nos arquivos. Isso poupa espaço de armazenamento mas para se chegar à versão mais recente necessita-se fazer todos os merges com os deltas para alcançá-la.
- Já o Reverse salva apenas a versão mais recente e guarda os deltas para outras versões. Isso é uma vantagem em relação ao forward pois é mais provável precisar de versões mais recentes do que de versões mais antigas, e para isso são feitos menos processos.



6 – Existem 3 tipos de colaboração: pessimista, otimista e misto. O pessimista não permite que mais de um funcionário faça check-out em um arquivo e modifique-o, isso evita que sejam necessárias escolhas de mudanças depois. Já o otimista permite que isso seja feito, mas caso haja incompatibilidades serão necessários ajustes para o merge. A colaboração mista alerta o usuário que já houve um check-out naquele arquivo e deixa a cargo do usuário continuar com a ação ou não.

7 – Codeline é a linha de versões de um item de configuração, baseado em seu controle de modificações. Baseline é a junção de versões estáveis de itens de configuração e ferramentas auxiliares para criar uma versão estável do sistema. E mainline é a linha de evolução das baselines, ou seja, o conjunto delas. Release é uma versão de sistema lançada para o usuário.



8 – O merging é feito quando dois desenvolvedores clonam o mesmo item de configuração do repositório (branch), fazem mudanças nele e tentam reenviar ao repositório. Quando isso acontece, as diferentes mudanças por cada usuário desestabilizam o item e é necessário fazer escolhas de mudanças. O ideal é evitar o uso de mergings.

9 – Versão é uma “etapa” de desenvolvimento de um item, e configuração diz respeito a um sistema, ou seja, é a junção de versões dos itens necessários para criar um sistema estável.

10 – Ramo ou branch é a subdivisão de desenvolvimento de um item de configuração, ou seja, um item terá versões construídas em paralelo. Mainline é o conjunto de releases de um software.

11 –

TortoiseCVS:

Site: <http://www.tortoisecvs.org/>

Licença: GNU GPLv2

Preço: grátis

Mercurial SCM:

Site: <https://www.mercurial-scm.org/>

Licença: GNU GPL v2+

Preço: grátis

Git:

Site: <https://git-scm.com/>

Licença: GNU GPLv2

Preço: grátis

12 –

Freshservice:

Site: <https://freshservice.com>

Preço: Pacotes de US\$19 até US\$99

Gensuite:

Site: <https://www.gensuite.com/>

Preço sob consulta

Intelligent Service Management:

Site: <https://serviceaide.com/intelligent-service-management/>

Preço sob consulta

13 –

Bugzilla:

Site: bugzilla.org

Licença pública Mozilla

Preço: grátis

Git:

Site: <https://git-scm.com/>

Licença: GNU GPLv2

Preço: grátis

Jira:

Site:

https://www.atlassian.com/br/software/jira?=&aceid=&adposition=1t1&adgroup=68402436860&campaign=1728702166&creative=337246038703&device=c&keyword=jira&matchtype=e&network=g&placement=&ds_kids=p41901120498&ds_e=GOOGLE&ds_eid=700000001550060&ds_e1=GOOGLE&gclid=Cj0KCQjwn_LrBRD4ARIsAFEQFKszOAHFVKvBn489RmEuwbooEAdmXKk5MpENkItE1Kqp1KLbNgPOH3MaAko6EALw_wcB&gclidsrc=aw.ds

Licença comercial

Preço: pacotes de em média US\$20/mês