TRABALHO TEÓRICO SOBRE GESTÃO DE CONFIGURAÇÃO

Aluno: André Victor Moura Teixeira Sousa

Matrícula: 201710160

1. Controle de configuração é o processo de garantia de que as versões de sistemas e componentes sejam registradas e mantidas para que as mudanças sejam gerenciadas e todas as versões de componentes sejam identificadas e armazenadas por todo o tempo de vida do sistema. É o desenvolvimento e aplicação de padrões e procedimentos para gerenciar um produto de sistema em desenvolvimento.
Controle de versão é um sistema que registra alterações em um arquivo ou conjunto de arquivos ao longo do tempo para que possa lembrar versões específicas mais tarde.
Controle de modificações armazenam todas as informações geradas durante o andamento das solicitações de modificações e relatam essas informações aos participantes interessados e autorizados.

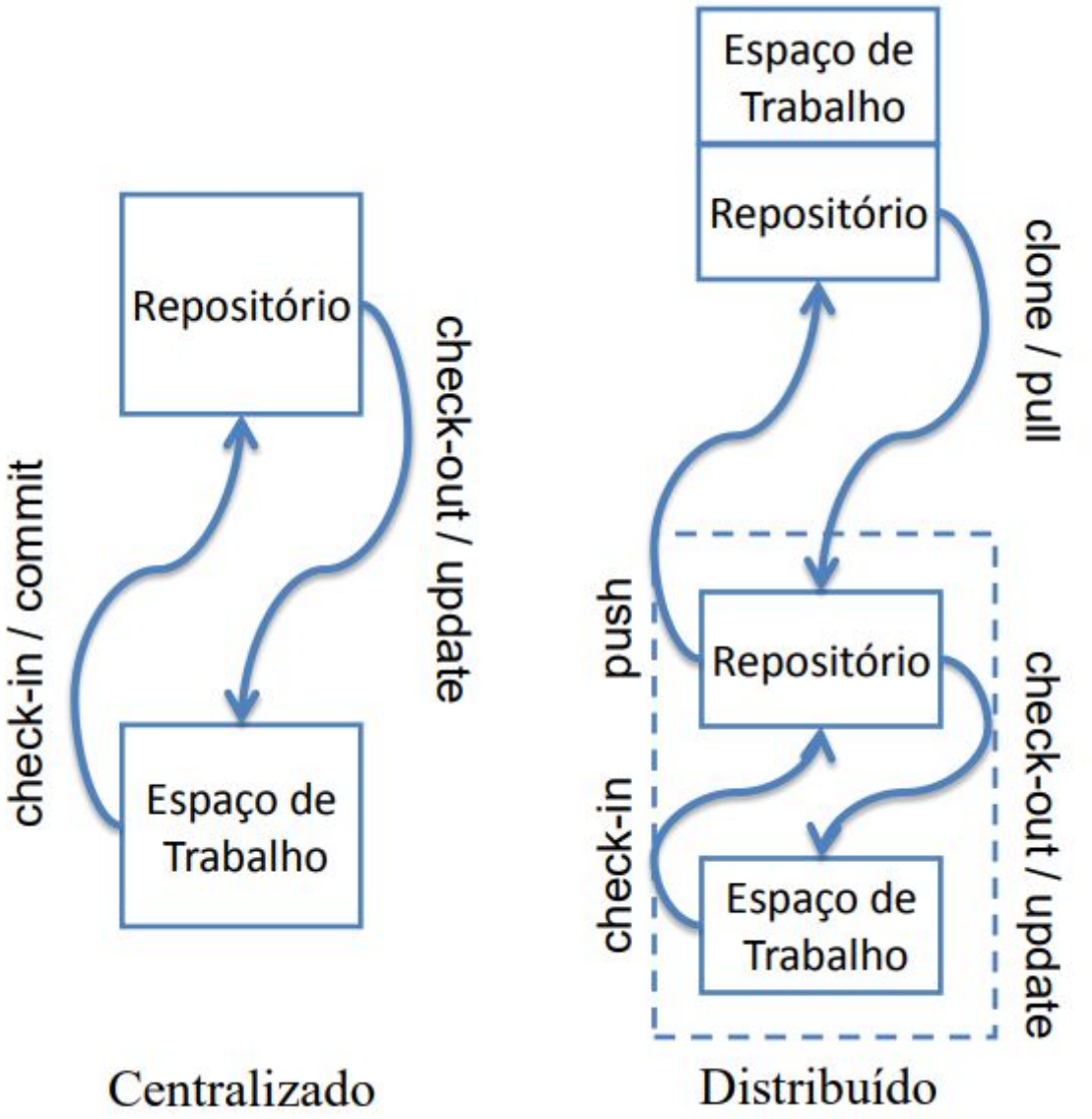
2. Qualquer coisa associada a um projeto de software (projeto, código, documentos, etc) que tenha sido colocado sob controle de configuração.

3. Derivado - Item de configuração que pode ser obtido a partir de outro item de configuração.
Fonte - Os itens de configuração que compõem o código-fonte.

4. Quando falamos de menor granularidade, ou granularidade fina, significa maior detalhamento (menor sumarização) dos dados. Maior granularidade, ou granularidade grossa, significa menor detalhamento (maior sumarização). Assim podemos notar que a granularidade e o detalhamento são inversamente proporcionais.

5. Centralizado e distribuído.

Centralizado: trabalha apenas com um servidor central e diversas áreas de trabalho, baseados na arquitetura cliente-servidor. Por ser centralizado, as áreas de trabalho precisam primeiro passar pelo servidor para poderem comunicar-se.
Distribuído: cada área de trabalho tem seu próprio “servidor”, ou seja, as operações de check-in e check-out são feitas na própria máquina. Porém diferentemente do centralizado, as áreas de trabalho podem comunicar-se entre si, recomenda-se usar um servidor como centro do envio dos arquivos para centralizar o fluxo e evitar ramificações do projeto e a perda do controle sobre o mesmo, geralmente o sistema te dá essa opção, oferecendo um servidor remoto para hospedar o projeto. A comunicação entre o servidor principal e as áreas de trabalho funciona com outras duas operações, para atualizar e mesclar o projeto, chamadas de pull e push.



O completo armazena cada versão no repositório demandando muito espaço. Permite rápida recuperação dos ICs.
A política de armazenamento baseado em forward armazena a primeira versão integralmente no repositório, e seus deltas em seguida, até a versão mais atual. É útil quando o desenvolvimento depende da recuperação constante de versões iniciais do sistema.
O reverse armazena os deltas e a partir deles são geradas as versões integrais.

6. Pessimista: somente um desenvolvedor pode modificar o IC em um dado momento, tem custo zero de junção de trabalho e ausência de paralelismo no desenvolvimento.
Otimista: vários desenvolvedores podem modificar um mesmo IC ao mesmo tempo. Tem um alto custo de junção de trabalho no caso de ICs complexos e permite o paralelismo no desenvolvimento.
Misto: permite que qualquer desenvolvedor saiba quem mais está modificando o IC. Tem um bom custo/benefício entre controle otimista e pessimista.

7. Codeline é um conjunto de versões de um componente de software e outros itens de configuração dos quais esse componente depende.
Mainline trata-se de uma sequência de baselines que representam diferentes versões de um sistema.
Baseline é uma coleção de versões de componentes que compõem um sistema. Elas são controladas, o que significa que as versões dos componentes que constituem o sistema não podem ser alteradas. Um tipo de baseline são os de qualidade: são métricas identificando as qualidades que se espera obter com os produtos ou resultados do projeto.
Release é uma versão de um sistema que foi liberada para os clientes para uso.

8. Trata-se da criação de uma nova versão de um componente de software, fundindo versões separadas em diferentes codelines.

9. Não. Uma versão é uma instância de um item de configuração que difere, de alguma forma, de outras instâncias deste item. Elas sempre tem um identificador único.
Configurações são diferentes formas de juntar partes do software para montar um software final. Pode ser vista como um IC composto de outros ICs. Pode ser formada por exemplo com a versão mais nova de um componente e a mais antiga dele. Podemos ter vários tipos de configurações como de sistema, processo, código fonte, etc.

10. Branches são “ramos” que isolam o desenvolvimento da linha principal mantendo o projeto original intacto sem que o mesmo seja afetado com novos erros que possam acontecer durante o desenvolvimento.
Trunk é o principal tronco de desenvolvimento, seguindo desde o início do projeto até o presente. É nele que o projeto deve se basear sempre. Normalmente é gerido por um desenvolvedor e recebe merges após aprovação de alguém que é responsável pelo projeto. Não faz sentido existir mais do que um trunk.

11. Git – Licença: GNU GPL v2. Preço: Gratuito. Site: https://git-scm.com/
Revision control system(RCS) – Licença: GNU GLP. Preço: Gratuito.
Site:https://www.gnu.org/software/rcs/
Concurrent Version System (CVS) – Licença: GNU GPL v2. Preço: Gratuito.
Site:https://savannah.nongnu.org/projects/cvs/
12. Trac – Licença BSD. Preço: Gratuito. Site: https://trac.edgewall.org/
Sharepoint – Licença: Proprietária. Preço: R$24~R$95.
Site:https://www.microsoft.com/pt-br/microsoft-365/sharepoint/collaboration
Redmine – Licença: GNU GPL v2. Preço: Gratuito. Site: https://www.redmine.org/
13. Subversion – Licença: Código aberto distribuído sob Licença Apache. Preço: Gratuito.
Site: https://subversion.apache.org/
Mercurial – Licença: GPL v2. Preço: Gratuito. Site: https://www.mercurial-scm.org/
Tortoisesvn – Licença: GNU GPL. Preço: Gratuito. Site: https://tortoisesvn.net/