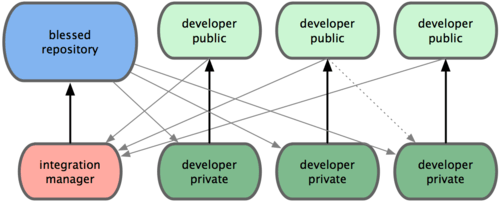
|  |  |
| --- | --- |
|  | **UNIVERSIDADE DO OESTE DE SANTA CATARINA**  **UNOESC – CHAPECÓ** |
| **Fluxos de trabalho** | |

|  |  |
| --- | --- |
| **DADOS DA DISCIPLINA:**       Disciplina: Git  Curso: Engenharia da Computação / Sistemas da Informação       Data: 23/10/17 Turno: Vespertino  Professor: Iskailer Rodrigues  Acadêmicos: Daniel Wenzel,  Jhonatan Oliveira,  Nicolas Maia,  Ayrton Lorenceti,  Lucas Medeiros   |  | | --- | |  | |
| **CONTEÚDOS:** Trabalho sobre os tipos de fluxos de trabalho em git. |

Fluxo de trabalho do Gerente de Integração

Quando iniciamos um projeto de software, e decidimos trabalhar com uma equipe de profissionais de TI, é comum a utilização do git para gerenciamento do projeto em si.

O fluxo de trabalho do gerente de integração se baseia em um gerente central que controla as informações que entram e que saem do projeto, possuindo acesso ao repositório principal (blessed repository), enquanto os demais desenvolvedores trabalham em cima de um clone do projeto privado até o momento em que é conveniente mesclar seu projeto com o principal.



Exemplo de etapas usadas pelo fluxo de trabalho gerente de integração (GitHub):

1. Gerente cria um repositório base para armazenamento do projeto principal;
2. Desenvolvedores realizam um fork do projeto para seu próprio repositório público;
3. Desenvolvedores criam um clone do projeto público para um repositório local (máquina de trabalho);
4. Desenvolvedores *criam novos arquivos/modificam arquivos* do projeto em sua máquina local;
5. Desenvolvedores realizam um pull do repositório local para seu próprio repositório público;
6. Desenvolvedores realizam um pull request de seu projeto público para dentro do repositório principal (gerente);
7. Gerente de integração avalia os arquivos do projeto e autoriza ou não o pull request;

* Caso o gerente aceite o pull request, o projeto do desenvolvedor é mesclado com o base;
* Caso o gerente negue o pull request, o projeto do desenvolvedor é ignorado (o gerente pode criar uma mensagem de resposta ao desenvolvedor, com o motivo no qual o pull foi negado);

Logo após, o processo volta à etapa 4 até a finalização do projeto.

Esse fluxo é comumente utilizado, como citado da ferramenta de repositório online GitHub, pois os desenvolvedores podem continuar trabalhando em cima do projeto local enquanto o gerente administra os pedidos de pull. E o gerente pode continuar seu trabalho enquanto os desenvolvedores progridem em seus projetos.

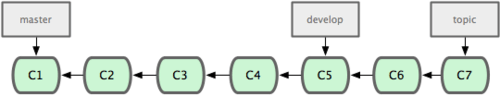
**Fluxos de Trabalho com Branches**

A criação e merge de branchs torna possível classificá-los em fluxos de trabalhos. A partir da análise destes fluxos, somos capazes de incorporá-los em ciclos de desenvolvimentos.

O git utiliza um merge de três vias. Fonte, que é o change set do qual se está executando o merge. Destino, o changeset para o qual se está executando o merge. E antecessor, o changeset (ou commit) que é o parente mais semelhante entre fonte e destino. Devido à está característica, é possível realizar o merge de um branch em outro várias vezes durante um período longo de maneira fácil.  Isso possibilita poder usar vários branches sempre abertos em diferentes estágios de um ciclo de desenvolvimento.

Este sistema de trabalho é adotado por muitos desenvolvedores Git. Eles utilizam os branches master somente para códigos já estáveis e aprovados. Em segundo plano, eles possuem outro branch para o desenvolvimento e realização de testes da estabilidade de códigos ainda não liberados. Para isso, ele puxa (pull) branches tópicos (branches de curta duração) quando já estiverem finalizados para passarem por uma avaliação definitiva de estabilidade. Quando os branches de desenvolvimento, comumente chamados de “develop”, chegam a um nível de estabilidade desejado, pode ser feito o merge com os branches master.

Neste tipo de sistema, os branches estáveis estão muito atrás na linha histórica de commits, e os branches de ponta, os que ainda estão sendo desenvolvidos, estão a frente no histórico.



Existem também projetos que, seguindo este mesmo princípio, utilizam vários branches “develop”, de desenvolvimento. Cada um com um nível de estabilidade específico. Quando um destes branches atinge o limite de estabilidade de seu branches, é feito então o merge com seu branch superior. Esse ciclo fica se repetindo até atingir o branch master.

AAS