

Estabilidade do Branch

Antes de falarmos sobre as **estratégias em si**, vamos conhecer alguns conceitos. **Ao trabalhar com diversas branches**, ao atuar como **package builder** ou **deployer** temos que conhecer **quais branches estão prontas para serem enviadas para os diversos ambientes**. O código das branches podem ser um código estável, pronto para ser enviado para produção, ou um código em desenvolvimento, **onde o *developer* está efetivamente trabalhando em novas funcionalidades** ou uma branch com código para validação.

Stable Trunk (Trunk-Based Development)

Nesse tipo de desenvolvimento temos uma **branch master** com o código sempre estável. **Novas funcionalidades são criadas a partir de branches locais e temporárias e são combinadas de volta com a master assim que atingem um ponto estável.**

Unstable Trunk

Esse caso dá **mais flexibilidade** ao **developer** pois não é preciso **garantir que o código esteja estável no master**. Tanto novas funcionalidades e bugfixes são aplicadas no master. Para criar um novo release, **é criada uma nova branch** a partir do master e **caso seja necessário novos commits para estabilizar a branch**, só esses commits voltam para o master (usando cherry-pick), e não há uma nova combinação das branches.

Integration Pipelines

Aqui a **branch master contém o código estável** e mais atualizado possível. As **branches de funcionalidades são criadas a partir do master** e **vários *developers* trabalham em paralelo cada um em sua branch**. **Ao final as branches são combinadas numa nova branch de integração, que servirá como base de um novo release.** Assim que **validado o código da branch de integração pelos *testers***, a **branch de integração** é combinada com o **master**, podendo ser descartada essa branch logo em seguida.