

01

Por que usar integração contínua ?

Transcrição

Imagine o seguinte caso, uma equipe de desenvolvedores que estão trabalhando em um projeto juntos, utilizam um servidor de versionamento do código-fonte, como o Git, Mercurial ou SVN. É comum, por engano ou alguma distração, que ao enviar um código para o Git algum desenvolvedor envie um código quebrado, que não compile. Os outros desenvolvedores, quando forem obter o código mais recente do Git, podem acabar obtendo o código quebrado, que havia sido enviado por engano pelo desenvolvedor descuidado. Isto atrapalha o fluxo de desenvolvimento da equipe, e poderia ser evitado caso alguma checagem rodasse automaticamente antes de integrar o código quebrado ao repositório de código central.

Outro exemplo bastante comum, é quando queremos pegar a última versão de projeto para buildar e colocar no servidor de produção. Quando não temos um servidor que executa nossos testes, o desenvolvedor responsável por colocar o código em produção só vai descobrir se o mesmo possui um bug ou não quando executar os testes em sua própria máquina. Imagine um cliente aguardando receber um software funcional, mas na véspera do lançamento, quando deveria ser enviado para produção que os bugs são descobertos. Tanto o cliente quanto o desenvolvedor responsável por esta tarefa ficarão furiosos pela falta de visibilidade do processo de entrega de software.

O que estes dois exemplos possuem de comum? Em ambos, o código defeituoso, seja por conta de erro de compilação ou pelos testes não passarem foram parar no servidor de versionamento. A descoberta dos erros foi de forma **tardia**, os erros ou falhas de um código problemático apareceram bem depois do que quando foram escritos.

A integração contínua vem para ajudar a **antecipar os problemas**, ela, continuamente, verifica o servidor de versionamento, checando a estabilidade do código: vendo se ele compila, se os testes executam com sucesso, se é possível empacotar o código para ser colocado em um servidor de produção. Essas verificações são feitas através de um software que fica no meio do caminho, entre o código comitado e o servidor de versionamento. Ele pode ser configurado, que é o que vamos fazer neste curso, para fazer verificações de acordo com a necessidade de cada equipe de desenvolvimento. Estes softwares são chamados de **servidores de integração contínua**.