

Integração contínua

Sumário

O que é

Ferramentas

Por que usar integração contínua?

Testes

Build

Servidor de Integração Contínua

Deploy

Integração Contínua

Surgiu como parte das práticas da metodologia ágil XP (Extreme Programming).

Tem como foco o desenvolvimento de software em ciclos menores, proporcionando melhor resposta a alterações e inclusão de novos requisitos.

Integração contínua trata-se de um conjunto de boas práticas que também podem ser adotadas em metodologias de desenvolvimento convencionais.

Integração Contínua

Tem como ideia principal a diminuição dos riscos através do monitoramento constante das alterações, mantendo a integração frequente do código.

Esta atividade passa a fazer parte do processo de desenvolvimento, sendo considerada como um procedimento normal e corriqueiro.

Integração Contínua

A mesma deve ocorrer com a maior frequência possível, idealmente a cada alteração efetuada(commit), pois em caso de problema, o conjunto de modificações em que a falha está inserida é menor, facilitando a identificação e correção do(s) erro(s).

Integração Contínua

Para que a integração seja efetiva, é necessário verificar o funcionamento do código após as modificações.

Para atender esta necessidade, o desenvolvimento em conjunto com a realização de testes é fundamental, de preferência com a adoção da metodologia TDD (Test Driven Development).

Integração Contínua

Como resultado da execução de uma integração bem sucedida, conseguimos atingir os seguintes benefícios:

Garantia que o código compila e que as funcionalidades testadas mantêm sua integridade e funcionamento esperados;

Uma potencial versão do software, que pode ser utilizada, por exemplo, pela equipe de QA para execução de testes.

Benefícios - Melhora a produtividade do desenvolvedor

A integração contínua ajuda sua equipe a ser mais produtiva ao liberar os desenvolvedores de tarefas manuais e encorajar comportamentos que ajudam a reduzir o número de erros e bugs implantados para os clientes.

Benefícios - Encontrar e investigar bugs mais rapidamente

Com testes mais frequentes, sua equipe pode descobrir e investigar bugs mais cedo, antes que no futuro os problemas cresçam demais.

Benefícios - Distribuir atualizações mais rapidamente

A integração contínua ajuda a sua equipe a distribuir atualizações para os clientes mais rapidamente e com maior frequência.

Developers



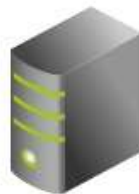
Code commits

**Source
Repository**



Triggers build

**Continuous
Integration
Server**



Compile

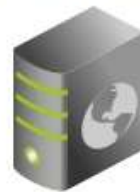
Run unit tests

Run integration
tests

Package

Build process

Web Server



Deploys
application

Ferramentas

Jenkins

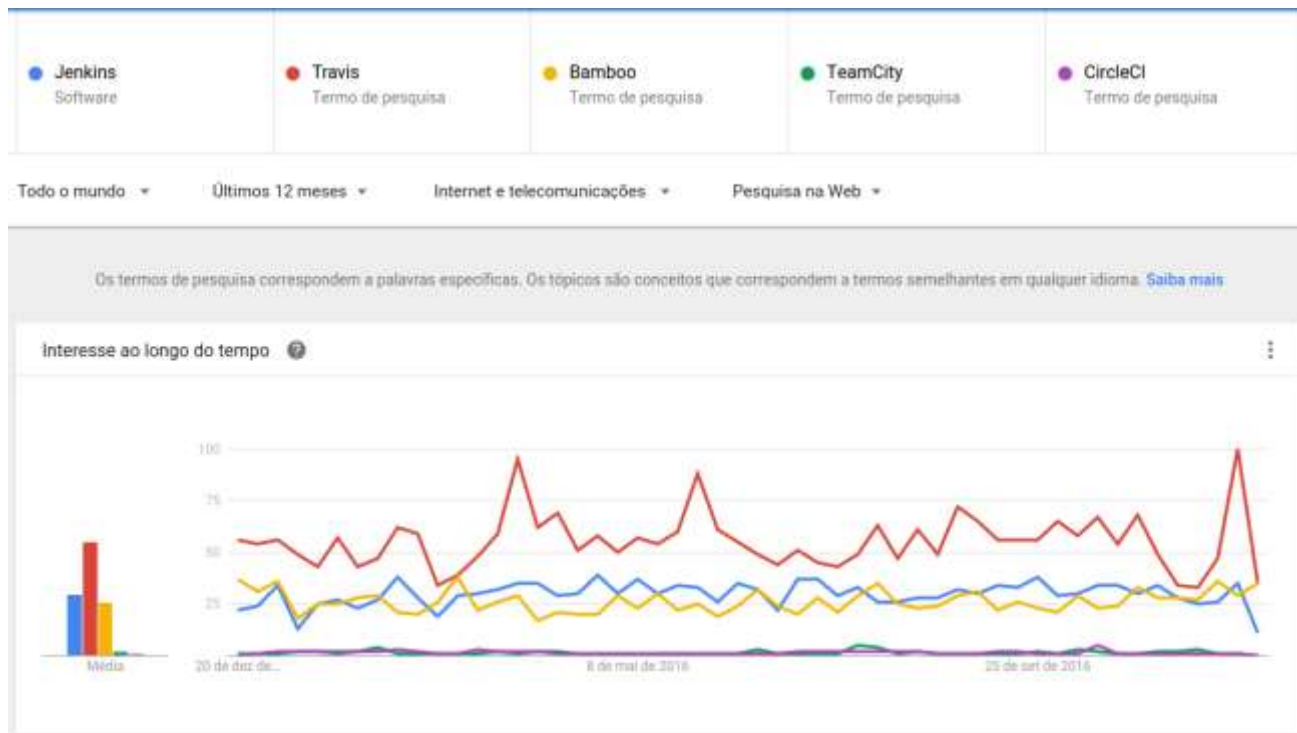
Travis CI

Bamboo

CircleCI

TeamCity

Ferramentas



Testes

Um bom modo para pegar erros mais rápida e eficientemente é incluir testes automáticos no processo de build. Testes não são perfeitos, mas eles podem pegar muitos bugs – o suficiente para ser útil.

Os testes necessitam estar aptos para rodarem com um simples comando e serem checados automaticamente. O resultado da suíte dos testes em execução deve indicar se estes falharam. Para uma build ser auto-testável a falha de um teste deve fazer com que a build falhe.

Build

Com a utilização de teste e o controle de versão centralizado, é necessária para uma boa integração, a automatização dos builds. O processo de build (compilar, preparar o executável, rodar os testes automatizados, etc.)

Não automatizar os builds, é um erro, pois diversos problemas ocorrem quando o processo é feito de forma manual, pois cada desenvolvedor acaba realizando o build de forma diferente, podendo criar problemas na integração.

Build

Um outro ponto importante, é que os builds devem ser auto testáveis, ou seja, durante os builds é necessário ter também os testes automatizados, como mencionados, para melhorar no processo de detecção de erros.

Build

A automatização do processo de build é possível por meio da criação de scripts, que executam cada uma dessas etapas da construção em sequência, reportando o resultado ao final da execução de cada uma. A automatização de build utilizando scripts traz benefícios para o processo de construção.

Build

- Facilidade na execução

O script de build geralmente é disparado por um ou poucos comandos, simplificando a execução do build.

Build

Padronização do build

O script de build é compartilhado por toda a equipe de desenvolvimento, sendo executado da mesma forma por cada integrante.

Build

Parametrização do build

O script de build pode ser elaborado para permitir a utilização de parâmetros durante a construção. Um exemplo é o fornecimento de parâmetros para que a versão construída esteja de acordo com os possíveis ambientes de execução da aplicação (ambiente de testes, ambiente de homologação ou ambiente de produção).

Build

Execução de todas as etapas

O script de build executa todas as etapas em sequência, garantindo que elas não sejam esquecidas ou suprimidas. O script de build até pode permitir a supressão de alguma etapa durante a construção, mas esse recurso deve ser usado apenas em casos específicos.

A facilidade no disparo do build é aproveitada pelo Servidor de Integração Contínua, para manter a integração e construção frequente do projeto.

Servidor de Integração Contínua

O Servidor de Integração Contínua é uma aplicação que tem como objetivo principal recuperar as alterações ocorridas no repositório e integrá-las, com a execução do script de build. Para que esta tarefa seja executada, o servidor deve permitir a configuração de planos de build, que armazenam as configurações necessárias para que a integração possa ser executada, como o endereço do repositório para monitoramento das modificações, a frequência da execução, o(s) comando(s) para o disparo do build e as configurações das notificações dos resultados.

Servidor de Integração Contínua

Uma funcionalidade essencial do Servidor de Integração Contínua é a configuração de mecanismos de notificação. Estes mecanismos divulgam os resultados da integração, sendo importantes para uma ação rápida por parte da equipe de desenvolvimento, em caso de falha na integração. Geralmente, o principal mecanismo de notificação utilizado é o disparo de e-mail, mas alguns mecanismos alternativos podem ser utilizados, como plugins para IDEs, plugins Desktop e o envio de mensagens de texto (SMS).

Deploy

Em termos gerais, deploy significa colocar em posição. Na prática, geralmente disponibilizar um sistema para uso, seja num ambiente de desenvolvimento, para testes ou em produção.

O termo deploy pode significar muitas coisas, dependendo do ambiente e da tecnologia. Um desenvolvedor pode fazer o deploy da aplicação web no seu servidor web local. Pode também fazer o deploy para produção, usado pelos usuários finais.

Deploy

Nível de automação

O deploy pode ser “manual”, parcialmente automatizado ou completamente automatizado.

Deploy

Uma das principais dores de cabeça que um cliente enfrenta, mesmo em empresas que adotam metodologias ágeis como XP e Scrum, está ligada ao produto ser colocado em produção e não funcionar: a famosa frase “mas no meu computador funcionou”.

Deploy

A solução está em criar um sistema de homologação.

Primeiro o projeto é compilado, depois os testes unitários são executados.

Na primeira fase, os testes executam as funcionalidades como o cliente faria e cobrem a maior quantidade de funções plausíveis de automatização. Na segunda fase, os desenvolvedores podem optar por deployar para um sistema de homologação, onde o cliente aprovará com os dados copiados de produção (alterações ligadas a sigilo de dados devem ser aplicadas).

Deploy

Por fim, quando o sistema está aprovado em homologação, com um clique o desenvolvedor efetua o deploy para produção.

Conclusão

Toda essa automatização tem um custo inicial de desenvolvimento que é compensado com a minimização de erros humanos no processo de deploy e na lentidão do mesmo.

A chance de funcionar em produção é maximizada uma vez que a mesma foi testada em homologação com uma cópia mais fidedigna do sistema de produção possível: menos reclamações e tempo perdido pelo cliente.

Referências

<http://www.devmedia.com.br/integracao-continua-da-teoria-a-pratica/28284>

<https://aws.amazon.com/pt/devops/continuous-integration/>

<http://www.devmedia.com.br/integracao-continua-da-teoria-a-pratica/28284>

<http://www.pedromendes.com.br/2009/03/04/traducao-do-artigo-sobre-integracao-continua/>

<http://blog.caelum.com.br/integracao-continua-deploys-e-aprovacoes-sem-dores-de-cabeca-para-o-cliente/>