

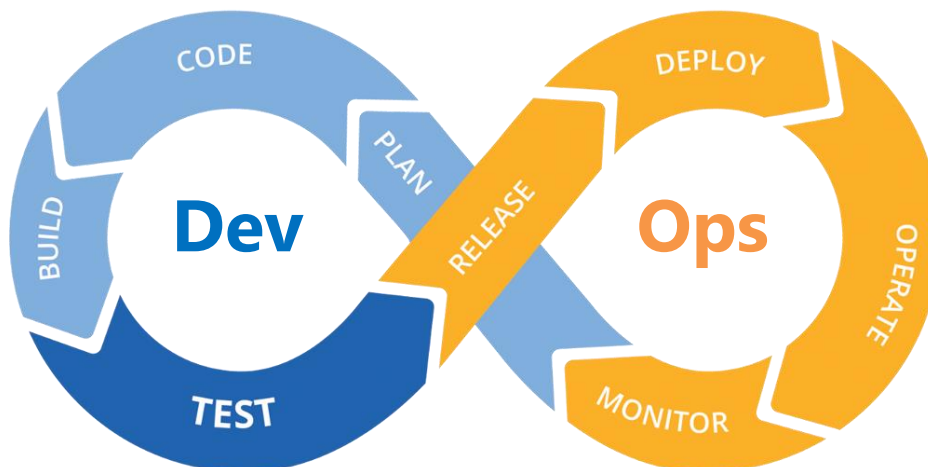
DevOps com GeneXus

Introdução



DevOps é uma metodologia que integra a criação de uma aplicação, com as operações posteriores ao desenvolvimento, como a colocação em produção, a manutenção e a evolução do software, em um ciclo sem fim. Vejamos como isso se aplica em GeneXus.

Definição de DevOps



DevOps é um conjunto de práticas ágeis que combina o desenvolvimento de software “Dev” e as operações de tecnologia da informação “Ops” para encurtar o ciclo de vida do desenvolvimento de sistemas e oferecer software de forma contínua.

O ciclo “Dev” (desenvolvimento) inclui o desenho, desenvolvimento, codificação, construção e teste da aplicação. O Ops (Operação) inclui a liberação, colocação em produção, operação e monitoramento da aplicação. Este monitoramento vai determinar futuras mudanças que devem ser feitas no software, como a manutenção por acertos de falhas ou a evolução para adicionar novas funcionalidades que permitam manter atual a aplicação ou dar-lhe um melhor uso.

A integração de todas as tarefas em um ciclo contínuo, permite às empresas que desenvolvem sistemas, aproveitar as oportunidades do mercado de forma mais rápida e reduzir o tempo para incluir respostas de clientes, melhorando a experiência dos mesmos com o software que utilizam.

Vejamos os principais problemas que se apresentam no ciclo de vida de uma aplicação.

Integração de mudanças



Salvo casos excepcionais, o desenvolvimento de software é uma tarefa de equipe. Um dos principais problemas que precisa ser resolvido é a integração das mudanças que cada desenvolvedor faz para a aplicação.

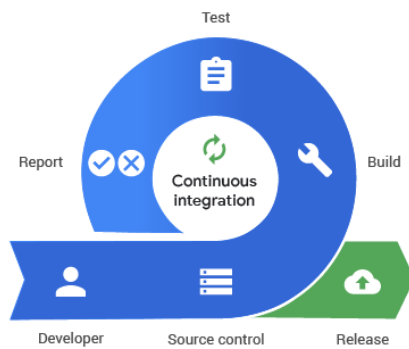
Cada desenvolvedor trabalha desenvolvendo uma funcionalidade, em um ciclo no qual codifica e testa até que cumpra seu objetivo.

Agora, o que acontece quando queremos integrar o trabalho desses desenvolvedores, de modo que as funcionalidades em que trabalharam sejam incluídas na próxima liberação da aplicação?

Ambos vão tentar subir suas mudanças para GeneXus Server, mas certamente haverá muitos conflitos e mais se estes desenvolvedores trabalharam de forma independente por um longo período de tempo. Talvez ambos tenham definido objetos com o mesmo nome e diferente propósito, ou talvez algum deles tenha modificado objetos existentes, removendo funcionalidades que outro desenvolvedor conta com elas.

Integração Contínua

- Integração frequente das alterações, pelo menos uma vez por dia
- Verificação da integração de forma automática



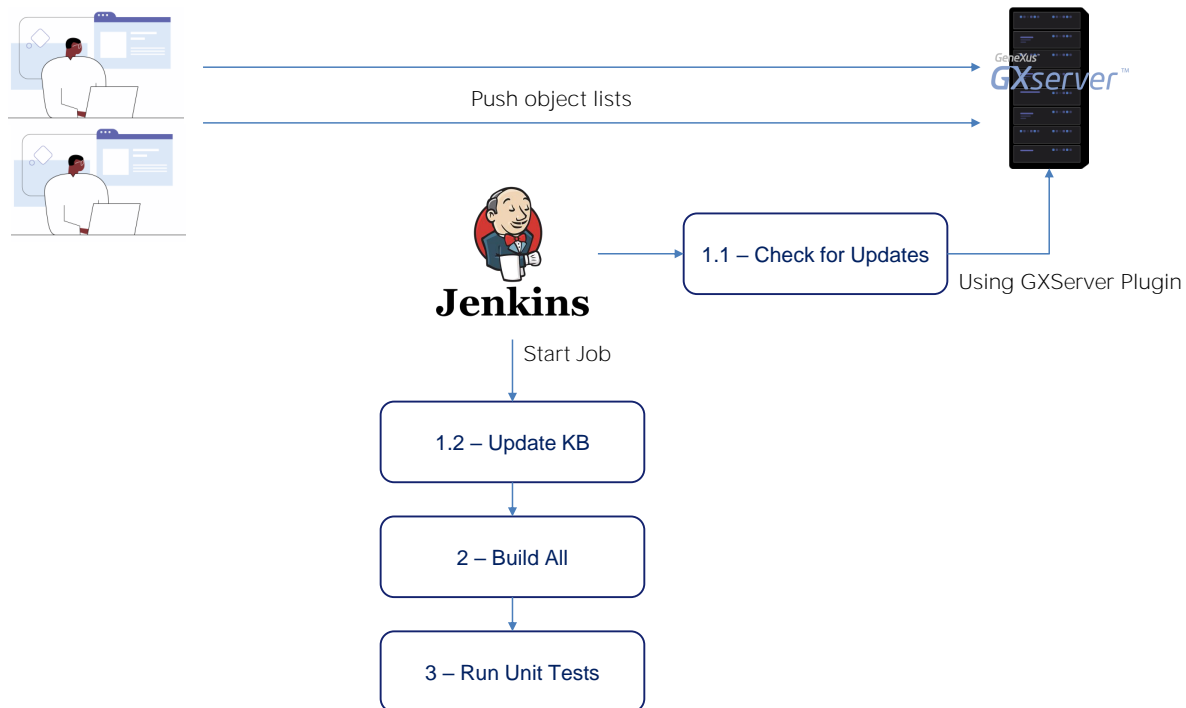
A ideia da integração contínua é que o trabalho se integre de forma frequente, se possível mais de uma vez por dia, de maneira que evite conflitos e, em casos que estes apareçam, sejam pequenos e fáceis de resolver.

Uma vez que se produz a integração, deve ser montada e testada a aplicação de forma automática para detectar os erros de integração o mais rapidamente possível. O teste automático também é útil para validar que não deixou de funcionar algo que já estava funcionando bem, o que se chama teste de regressão.

Ferramentas para a integração contínua



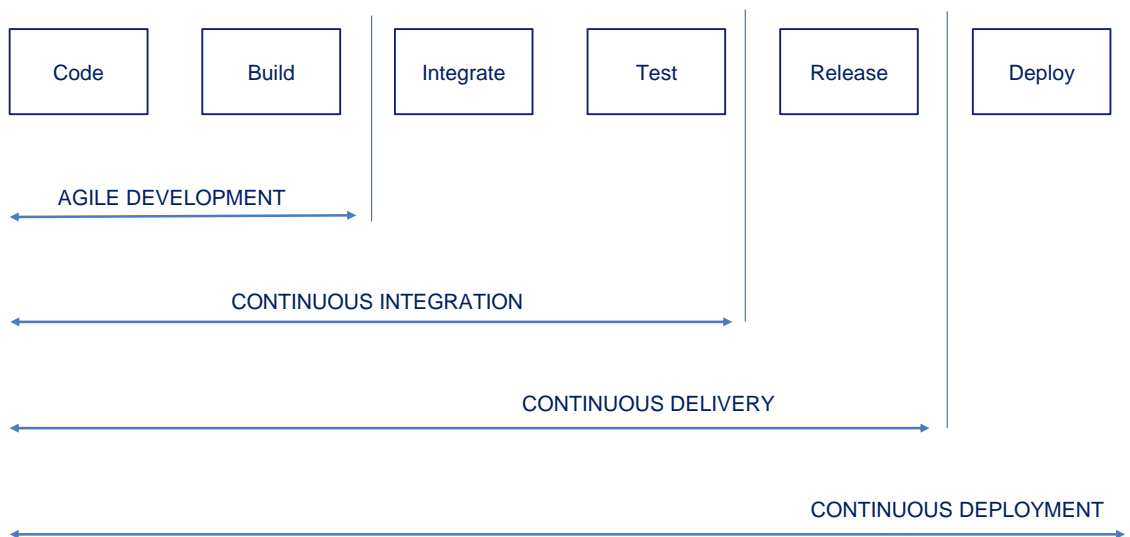
Para isso, vamos usar GeneXus Server e Jenkins que é um motor de integração contínua e é a ferramenta que vai orquestrar todo o processo.



Em cada commit que faça cada desenvolvedor, será realizado pelo menos o seguinte processo automático: Atualizar a KB, Build All e executar os testes.

Se tudo vai bem o build é bem-sucedido e é possível seguir adiante, caso contrário tem que parar o processo e resolver o conflito.

Etapas do ciclo de vida de uma aplicação



Estas são as etapas do ciclo de vida de uma aplicação.

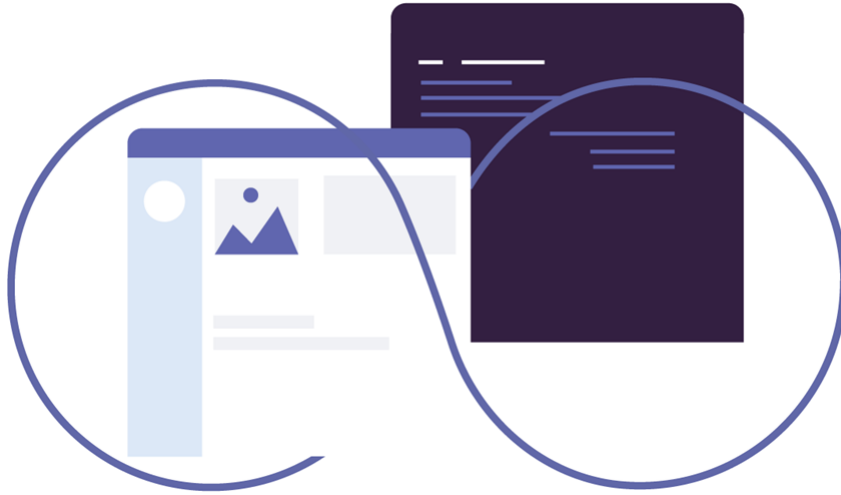
Se nosso processo é capaz de acelerar o processo de desenvolvimento e construção, temos uma metodologia de desenvolvimento ágil.

Quando adicionamos ao nosso processo a automatização da integração e validação, é que temos a Integração Contínua

Se a isso acrescentarmos a automação da montagem da release, de forma que, em seguida, publicar a produção se possa fazer com um clique, chegamos a Entrega Contínua

Quando nosso processo está tão acertado e automatizado que podemos chegar a colocar em produção de forma automática, é que temos Colocação em Produção Contínua.

Processo DevOps



E quando já incluímos em nosso processo a operação e monitoramento da aplicação em produção, de forma que se gere informação que retroalimenta o plano de quais funcionalidades convém desenvolver depois, é que chegamos a ter um processo DevOps.

Criação e monitoramento de processos de Integração Contínua

The screenshot shows the GeneXus IDE interface for Continuous Integration. The main workspace is divided into two sections:

Integration Pipelines

Status	Name	Version	Environment	Run	Last Run	Next Run
Failure	JUnitStable	JUnitStable	Java Environment	69	9/10/2020 17...	12/10/2020 10:31:51
Success	JUnitStableNew	JUnitStable	Java Environment	15	9/10/2020 17...	12/10/2020 17:26:48

Pipeline Activity

Run	Status	Run Date	Duration
15	Success	9/10/2020 17:26	00:06:05
14	Unstable	6/10/2020 20:19	00:01:01
13	Failure	6/10/2020 20:18	00:00:00
12	Failure	6/10/2020 19:38	00:00:00
11	Success	6/10/2020 18:11	00:00:48
10	Failure	6/10/2020 18:08	00:00:49

Como vimos, para alcançar a metodologia DevOps, devemos ter certeza de alcançar a Integração Contínua, isto é, após o desenvolvimento da aplicação, automatizar a integração de mudanças e a validação.

É possível criar e monitorar processos de integração contínua (pipelines) a partir do IDE de GeneXus e do console de GeneXus Server.

Uma vez que subimos a KB para GXserver, se vamos para Knowledge Manager / TeamDevelopment, temos a aba "Continuous Integration" na qual podemos criar pipelines, ver suas execuções, quais commits entraram em cada uma, o resultado e o log completo de cada execução.

Mais informações:

<https://wiki.genexus.com/commwiki/servlet/wiki?40706>

Para mais informações sobre o uso da metodologia DevOps em GeneXus, visite o seguinte link da Wiki:

<https://wiki.genexus.com/commwiki/servlet/wiki?40706>

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications