

# DevOps

João Carlos Testi Ferreira

Faculdade SENAI

Florianópolis, 2016

## Sumário

### 1 Um pouco de história

- Software e hardware

### 2 Ambientes

- Desenvolvimento e Produção
- Processo de desenvolvimento
- Ambientes segregados

### 3 Perfis

- Desenvolvimento
- Produção

### 4 A evolução

- Desenvolvimento

### 5 Produção

- Modelo de implantação

### 6 Resumo

- Histórico
- Equipes
- DevOps
- Referências usadas

## O início

No início da informática tudo era muito complexo. Quem era dessa área era visto como uma pessoa muito inteligente e deslocada da sociedade.



## Hardware

As coisas aconteciam no nível do hardware. Para programar era necessário conhecimento de eletrônica. Era preciso conhecer o funcionamento interno da máquina.



## Linguagem de programação

Para tornar mais viável o uso do computador foram criadas linguagens de programação.

No início em um nível muito baixo, ainda exigindo conhecimentos da estrutura do processador.



## Simplificando

Na medida que mais interessados surgem para o uso da informática ela precisa ser simplificada. Os problemas a serem resolvidos aumentam de complexidade – logo a solução deve ser simples o suficiente para suportar a complexidade do problema.

Handwritten mathematical derivation showing the simplification of an equation:

$$11x = 22$$
$$x = \frac{22}{11}$$
$$x = 2$$

## Características

No início não havia diferença entre o local que se desenvolvia e o local em que a aplicação estava disponível para o cliente, até porque estes ambientes eram muito caros para haver qualquer replicação. As coisas aconteciam "a quente".



## Riscos

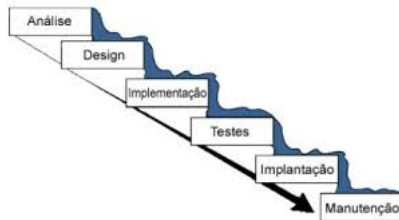
Na medida que o software passa a sustentar o negócio, essa forma de trabalho torna-se inviável. É necessário separar o local de desenvolvimento do local em que a aplicação está disponível para o cliente.





## Padronização

A padronização da forma de desenvolver, o reforço na definição de processos e o uso de modelos tradicionais de desenvolvimento cresceram com o objetivo de se obter segurança. Identificaram necessidades de ambientes controlados para o desenvolvimento, cada ambiente com um objetivo.

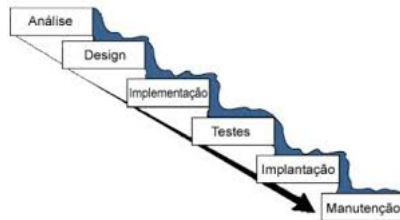


## Um ambiente para cada objetivo

### Objetivos

tão distintos nas etapas do processo forçam a criação de alguns ambientes. Tipicamente temos 4 ambientes:

- Desenvolvimento
- Testes
- Homologação
- Produção



## Desenvolvimento

O ambiente de desenvolvimento é sujeito a pouco controle. Ocorrem muitas instalações e desinstalações. Experiências são realizadas, a aplicação é submetida a situações extremas, força-se o erro para identificar o comportamento da aplicação.



## Teste

O ambiente de testes precisa ser mais controlado que o de desenvolvimento. Não podem ser instaladas ou desinstaladas coisas à revelia – ele deve ser parecido com o ambiente de produção. Ele deve permitir avaliações das mais diversas, inclusive de crash do sistema.



## Homologação

Quando quem vai testar é o cliente, deve se ter certeza de que o ambiente é idêntico ao de produção (normalmente com as mesmas características, porém reduzido). Normalmente não serão realizados testes mais críticos, o que se realiza são testes de uso.



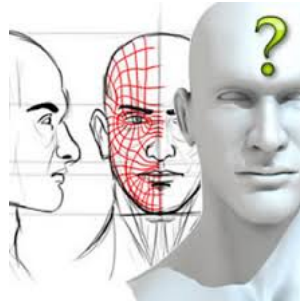
## Produção

Com o software sustentando o negócio, a produção passa a ser um ambiente crítico – não devem ocorrer erros em produção, este deve ser um ambiente estável, robusto e resiliente.



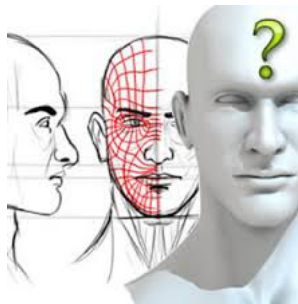
## Características de desenvolvedor

Os desenvolvedores precisam de habilidades para realizar suas atividades. Além das habilidades, características comportamentais são desejadas, conforme o tipo de atividade realizada.



## Características de desenvolvedor

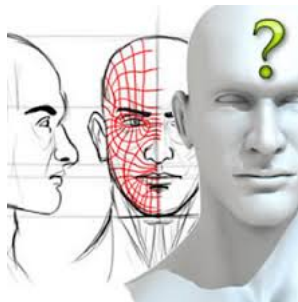
A característica de um codificador é diferente da característica de um testador. Um analista de requisito, por exemplo, deve ser uma pessoa que tenha como característica perguntar mais e inferir menos.





## Características de desenvolvedor

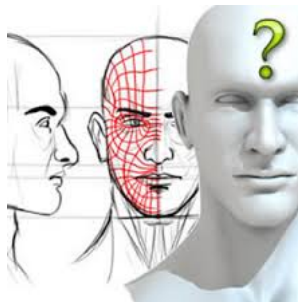
Como regra geral, podemos dizer que o pessoal de desenvolvimento é do tipo que gosta de experimentar, fazer coisas novas ou diferentes, realizar atividades que exijam conhecimento e habilidade para criação.



## Característica da equipe de produção

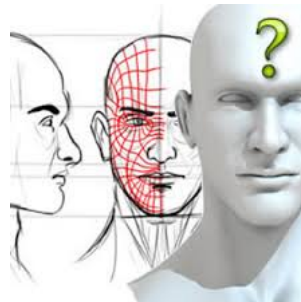
Além das habilidades necessárias para uma pessoa atuar em ambiente produtivo são esperadas algumas características comportamentais.

Como exemplo, uma pessoa de produção tende a ser alguém que gosta de estabilidade e segurança.



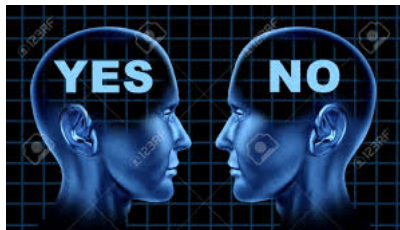
## Característica da equipe de produção

Pessoas de produção não são "simpáticas" a mudanças, pois cada mudança implica em instabilidade do ambiente e, para eles, o ambiente deve ser o mais estável possível. Quanto menos muda, menos instabilidades temos.



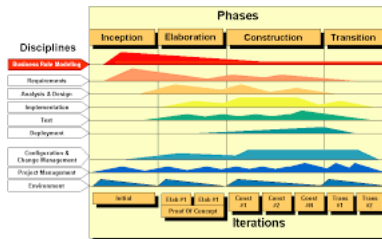
## Perfis antagonicos

A característica oposta destes perfis é outro fator que complica o processo de criação e evolução do software. Ainda mais complicado porque, como temos perfis conflitantes, as habilidades acabam por ser exclusivas de um ou outro perfil.



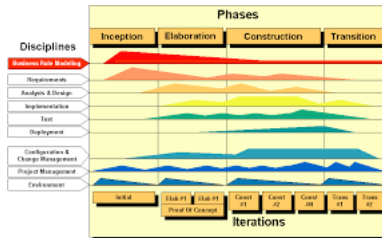
## Processos tradicionais

Para melhorar a qualidade do software, gerando a segurança necessária para o negócio e o retorno desejado para o dinheiro investido, processos tradicionais com grande carga de controle passaram a ser recomendados e cresceram em uso e importância no mercado.



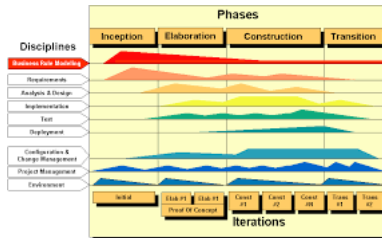
## Processos tradicionais

Houve a busca por certificação em processos (CMMI, MPS) que conferiam à organização que ela seguia um processo bem definido na execução de suas atividades. A escolha dos clientes, em muitos casos, levava em consideração tais certificações.



## Processos tradicionais

Se sob o aspecto de risco a busca por processos bem definidos foi uma alternativa, observou-se a dificuldade de se ter o software no tempo esperado. Para que o tempo fosse atendido, o processo era atropelado e a razão para o processo definido se perdia.



## Métodos ágeis

Em resposta a grande dificuldade de atendimento das necessidades dos clientes, vários metodologistas romperam com a "segurança" do processo tradicional e partiram para uma abordagem mais agressiva – foco no atendimento das necessidades do cliente.





## Métodos ágeis

Para evitar conflitos entre as partes envolvidas no desenvolvimento, passamos a tratar todos como **uma equipe**. Elimina-se o cliente versus equipe, codificador versus testador e assim por diante.



## Métodos ágeis

Além da redução dos conflitos, temos uma comunicação mais próxima, muito menos sujeita a mal-entendidos. A proximidade fornece mais comprometimento à equipe, todos percebem o que cada um faz, como colaboram e como produzem o que se espera ao final do projeto.



## Produção tradicional

O modelo de produção tradicional causa forte impacto na implantação de novas aplicações ou suas evoluções. Até por conta da característica do pessoal de produção, a implantação é um gargalo que pode atrasar ou até inviabilizar a entrada em produção das aplicações.



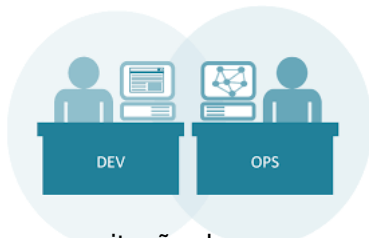
## Produção tradicional

Outro ponto a considerar é a complexidade do software e de sua implantação de forma distribuída, que dificulta ou até impede a ação da equipe de produção de realizar essa atividade sem algum tipo de assistência. Aumenta a necessidade da equipe de produção ter alguma noção de desenvolvimento. Da mesma forma, a equipe de desenvolvimento precisa conhecer como a aplicação será mantida em produção.



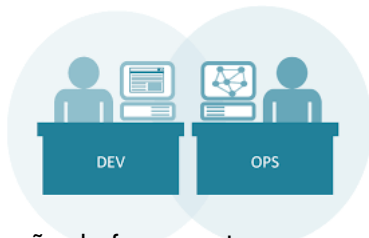
## DevOps

A aproximação do desenvolvimento com a produção torna-se essencial para o atendimento das necessidades dos clientes. Essa aproximação implica em disseminação de conhecimento, em capacitação de pessoas, em flexibilidade de perfis.



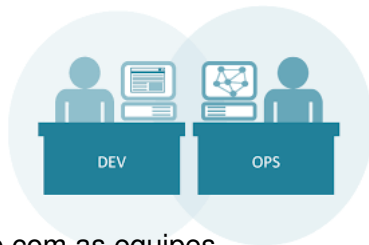
## DevOps

Para tornar o controle mais efetivo é necessário a inclusão de ferramentas que possam oferecer velocidade no processo e segurança de forma simultânea. Muitos acreditam que isso é DevOps (a simples inclusão de ferramentas nesse processo).



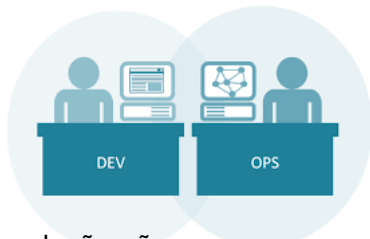
## DevOps

Outros acreditam que DevOps é um método de trabalho em que há uma etapa comum entre produção e desenvolvimento. Na verdade, mais do que qualquer coisa, DevOps é uma cultura, é algo para ser construído com as equipes.



## DevOps

Peça chave para o DevOps é a Gerência de Configuração. Sem um efetivo controle dos itens de configuração e automatização dos processos de evolução de tais itens, as atividades de desenvolvimento e produção não conseguem uma aproximação adequada de forma segura.





## Ambientes de desenvolvimento

O desenvolvimento envolve pessoas de perfis distintos e as atividades exigem ambientes segregados para sua execução. Os perfis envolvem aspectos, habilidade e comportamento e os ambientes referem-se a segregação e similaridade entre eles.

## A evolução dos ambientes

Os ambientes acabaram evoluindo para quatro ou cinco tipos. Temos o ambiente de desenvolvimento, testes, homologação e produção, podendo ainda ter um ambiente de quarentena.

## Desenvolvimento e produção

Normalmente as equipes possuem comportamento conforme suas atividades, assim, equipes de desenvolvimento são mais criativas, gostam de mudanças e equipes de produção gostam de estabilidade. Há um conflito entre estes perfis.

## Aproximação das equipes

A constante evolução e aumento de complexidade do que se produz, formas de instalação e necessidades envolvidas acaba por forçar as equipes a terem mais conhecimentos que fogem de seu perfil. A necessidade de reduzir o tempo para implantação das aplicações força a aproximação de equipes distintas.

## DevOps e ferramentas

A aproximação das equipes não produz muito resultado se os interesses são opostos. Para aproximar estes interesses podemos usar ferramentas que facilitem processos de implantação e forneçam a segurança desejada pela produção.

## Cultura DevOps

Somente a aplicação de ferramentas não resolve o problema, é necessário que seja construída uma cultura de entendimento do processo que envolve essa aproximação. Essa cultura é produzida pela capacitação das pessoas nas áreas diversas e pela sensibilização da necessidade da realização dessas ações.

## Para saber mais ...

DAVIS, Jennifer; DANIELS, Katherine. **Effective devops**: building a culture of collaboration, affinity, and tooling at scale. Sebastopol: O'Reilly Media, Inc., 2015.

HÜTTERMANN, Michael. **Devops for developers**. New York: Apress, 2012.

SWARTOUT, Paul. **Continuous delivery and devops**: a quickstart guide. 2 ed. Birmingham - Mumbai: Packt Publishing Ltd, 2014.