



# DevOps

## Aula 2 - conceitos e *continuous integration*

---

Eidil Santos

[eidil.santos@faculdadeimpacta.com.br](mailto:eidil.santos@faculdadeimpacta.com.br)

# DevOps

---

- What is DevOps:
  - DevOps is about people, process and tools for the benefit of business agility;

# DevOps

---

- What is DevOps:
  - two contential areas: Dev X Ops;
    - one wants to push changes;
    - other wants to maintain the environments;
  - balance between velocity and quality;
  - application delivery as a supply chain or pipeline (in a industrial way);

# DevOps

---

- What is DevOps:
  - IDEA:
    - user stories;
    - work items;
    - etc;
  - CODE:
    - programming;
  - BUILD:
    - unit test cases;
    - packaging;

# DevOps

---

- What is DevOps:
  - DEPLOY:
    - baremetal;
    - cloud;
  - MANAGE:
    - environments;
  - LEARN:
    - continuous improvement;
      - speed;
      - quality;

# DevOps

---

- What is DevOps:
  - USE CASES:
    - optimize core:
      - its about monolithic legacy systems, retrofitting automation, optimize what is possible;
    - unlock legacy:
      - culture;
      - ways of working;
      - Dev and Ops together;
    - unleashing digital;
      - maturation (12-factor, kubernetes, docker);
      - continuous delivery (from ideation to continuous improvement);

# DevOps

---

**DevOps  
e  
continuous  
integration**

# DevOps

---

Comunicação

Colaboração/Compartilhamento

Automação

Monitoramento



# DevOps

	1970s–1980s	1990s	2000s–Present
<b>Era</b>	Mainframes	Client/Server	Commoditization and Cloud
<b>Representative technology of era</b>	COBOL, DB2 on MVS, etc.	C++, Oracle, Solaris, etc.	Java, MySQL, Red Hat, Ruby on Rails, PHP, etc.
<b>Cycle time</b>	1–5 years	3–12 months	2–12 weeks
<b>Cost</b>	\$1M–\$100M	\$100k–\$10M	\$10k–\$1M
<b>At risk</b>	The whole company	A product line or division	A product feature
<b>Cost of failure</b>	Bankruptcy, sell the company, massive layoffs	Revenue miss, CIO's job	Negligible

Fonte: Adrian Cockcroft, “*Velocity and Volume*”, apresentação de 2013.

# DevOps

---

- Mudanças técnicas;
  - Mudanças arquiteturais;
  - Mudanças culturais;
- 
- Sendo isso resultado de mudanças ‘filosóficas’ no processo de desenvolvimento:
    - Agile;
    - Lean;

# DevOps

---

- Áreas comumente envolvidas:
  - Gestão de produtos (business);
  - Desenvolvimento;
  - QA;
  - Operações (infraestrutura, banco de dados, redes, gestão de mudanças, etc);
  - Segurança da informação;

# DevOps

---

While the foundation of DevOps can be seen as being derived from Lean, the Theory of Constraints, and the Toyota Kata movement, many also view DevOps as the logical continuation of the Agile software journey that began in 2001.

# Continuous Integration

---

- Objetivos;
  - desenvolver;
  - testar;
  - liberar código;

# Continuous Integration

- Objetivos;
  - desenvolver;
  - testar;
  - liberar código;

Mais rapidamente

Com maior frequência

# Continuous Integration

---

- Ciclos curtos;
  - desenvolvimento;
  - validação;
  - qualidade;

# Continuous Integration

---

- Condições:
  - através de um processo repetível;
  - o mais autônomo possível;
  - passos bem definidos;



# Continuous Integration

## WATERFALL



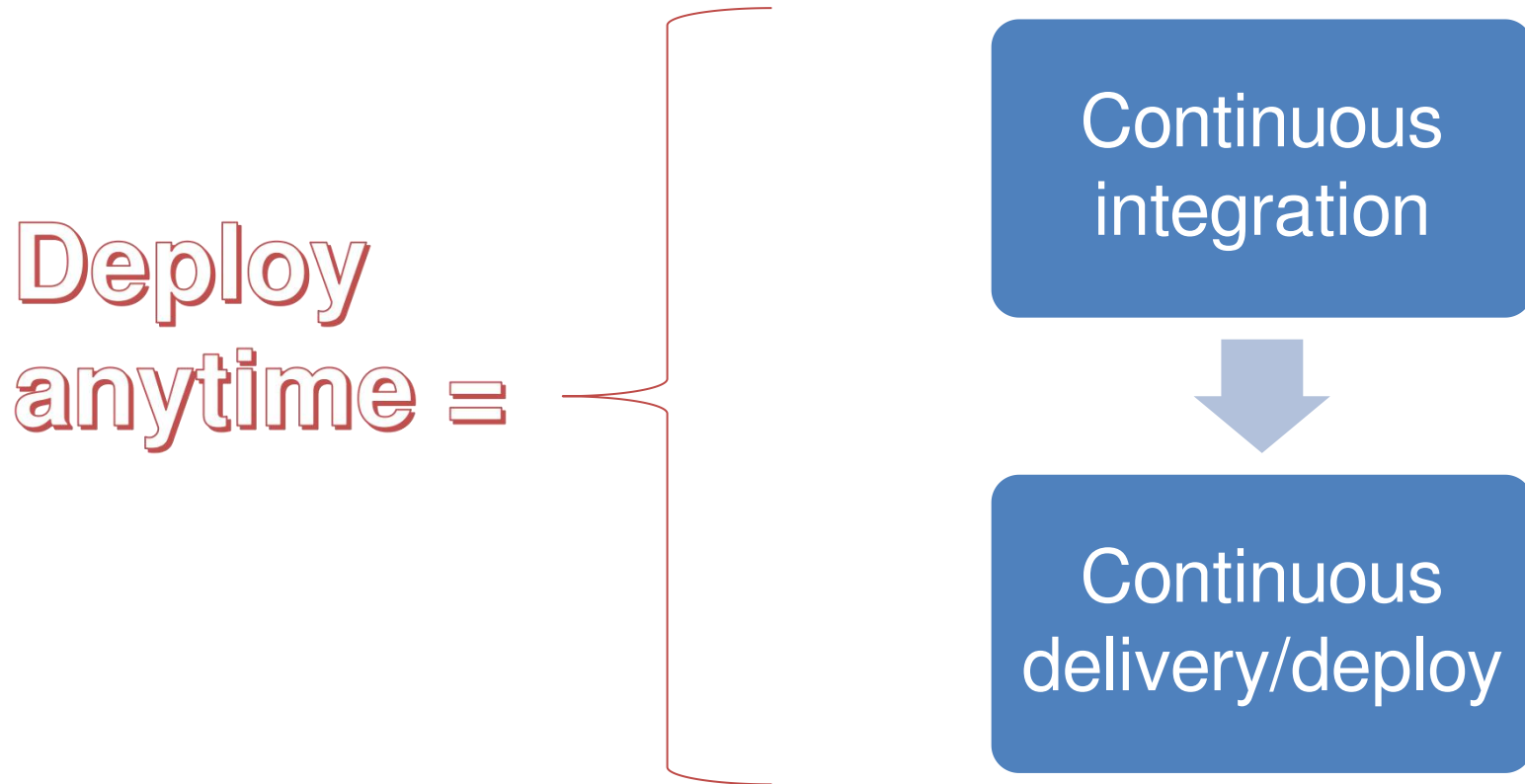
## AGILE



## DEVOPS



# Continuous Integration



# Continuous Integration

Deploy  
anytime =

- Desenvolvimento;
- Avaliação de qualidade:
  - revisão de código;
  - qualidade de código;
  - testes automatizados;
  - testes manuais;
- liberação de snapshots;
- liberação de releases;

# Continuous Integration

---

Continuous  
integration



- conhecer os processos;
- conhecer as pessoas;
- conhecer as tecnologias;
- buscar agilizar os processos;
- envolver as equipes;
- agilizar sem perder o controle;
- acelerar sem perder a qualidade;
- atender os clientes/business;

# Continuous Integration

Continuous  
integration

- conhecer os processos;
- conhecer as pessoas;
- conhecer as tecnologias;
- buscar agilizar os processos;
- envolver as equipes;
- agilizar sem perder o controle;
- acelerar sem perder a qualidade;
- atender os clientes/business;

**Equilíbrio entre  
velocidade e  
qualidade  
(eficiência)**

# Continuous Integration

---

- Case:
  - empresa do ramo financeiro:
    - 4 fábricas de software:
      - cada consultoria tem seu repositório;
      - 3 trabalham com Git e 1 com SVN;
    - 2 consultorias de testes:
      - consultoria A atua com gestão de testes e testes manuais;
      - consultoria B atua com automação de testes em uma das Squads;
    - time de infraestrutura ‘local’:
      - ambientes não-produção: on-premises;
      - ambiente produção: cloud;

# Continuous Integration

---

- Case:
  - empresa do ramo financeiro:
    - aprovações:
      - após ciclo de testes, reunião para definir subida para homologação;
      - após aprovação em homologação, comitê de mudanças;
    - processos:
      - cada consultoria atua de uma forma;
      - novas entregas chegam por diversas vias;
    - infraestrutura:
      - sem feedback pós mudança;
      - sem gestão de incidentes;

# Continuous Integration

---

- código:
  - problema -> códigos separados;
  - repositório centralizado:
    - gestão de código;
    - revisão de código;
    - repositório auditável;
  - merges simplificados;

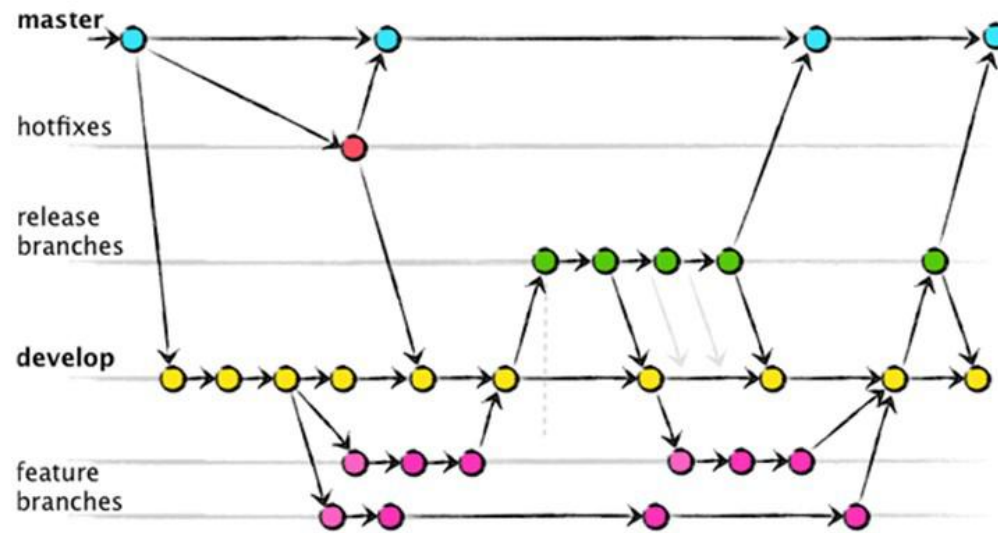


# Continuous Integration

---

- Workflow:
  - boas práticas;
  - código testável;
  - controle de histórico de alterações;
  - recuperar código em determinado ponto no tempo;
  - evitar perda de código;

# Continuous Integration



# Continuous Integration

---

- Aumentar a colaboração:
  - desenvolvimento;
  - operações;
  - qualidade;

# Continuous Integration

---

- Melhorar qualidade:
  - gestão do ciclo de testes;
  - gestão de defeitos;
  - automação;

# Continuous Integration

---

- Business:
  - atender ‘time-to-market’;
  - diminuir incidentes (ou melhorar a gestão disso);
  - feedback;

# DevOps

---

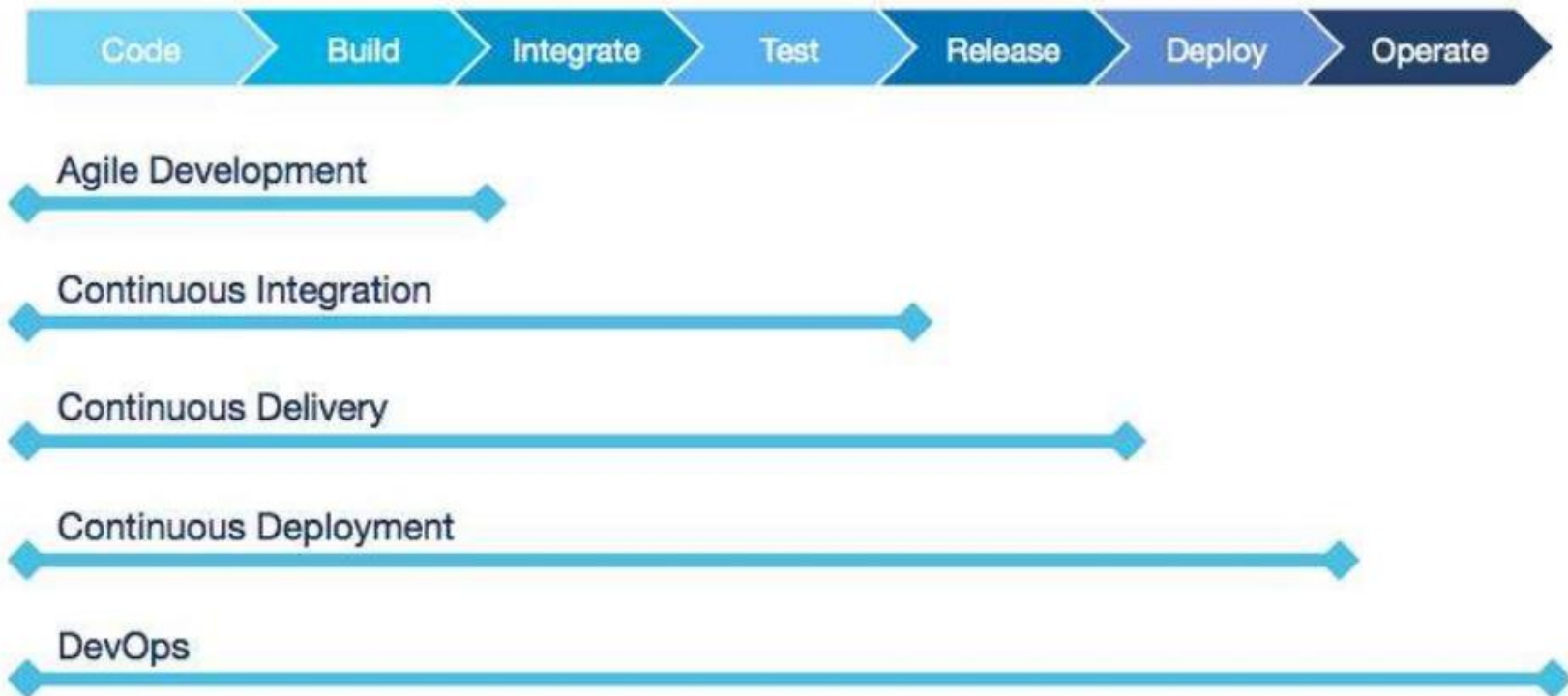
Comunicação

Colaboração/Compartilhamento

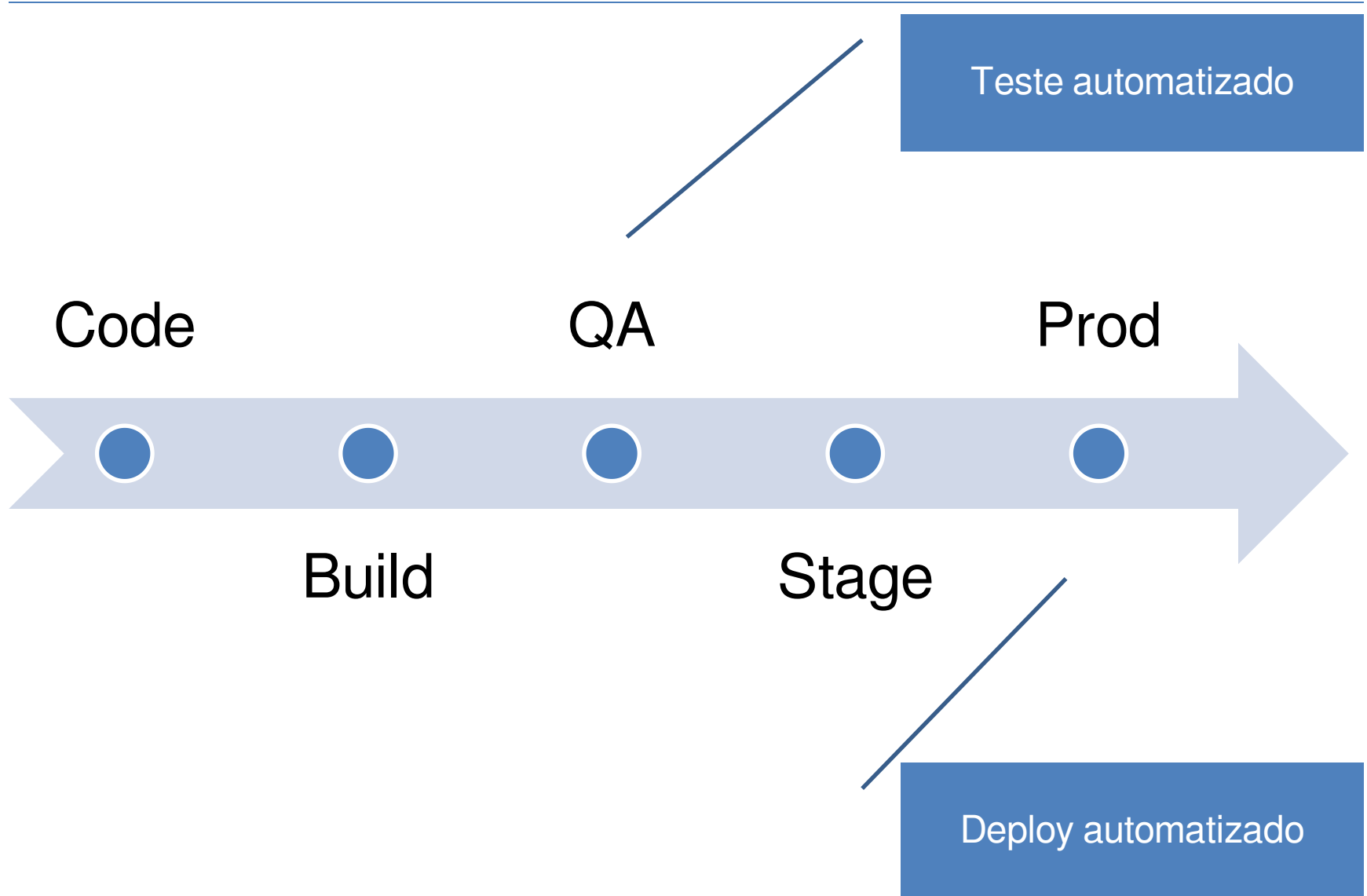
Automação

Monitoramento

# DevOps



# Continuous delivery





# Continuous delivery

---

Continua...