



Laboratório de Sistemas Computacionais Complexos

2020/04/16 - AULA 11

<https://uclab.xyz/sistemas-complexos-aula11>

Leonardo Leite
leofl@ime.usp.br

Renato Cordeiro Ferreira
renatocf@ime.usp.br

Thatiane de Oliveira Rosa
thatiane@ime.usp.br

João Francisco Daniel
joaofran@ime.usp.br

Alfredo Goldman
gold@ime.usp.br

Agenda

Tema da aula:

DevOps

1. Definição DevOps
2. Cultura DevOps
3. Entrega contínua
4. Runtime and reliability
5. DevOps e microserviços
6. Ferramentas

Definição de DevOps

DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions, while guaranteeing their correctness and reliability

(LEITE *et al.*, 2019)

Definição de DevOps - CAMS

- Culture
- Automation
- Measurement
- Sharing

(HUMBLE and MOLESKY, 2011)

DevOps

DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions, while guaranteeing their correctness and reliability

DevOps

DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions, while guaranteeing their correctness and reliability

Entre Devs e Ops

Antes o ágil ficava só nos devs

DevOps

DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions, while guaranteeing their correctness and reliability

Cultura de colaboração!

Cultura (Westrum, 2014)

- **Pathological** (power-oriented)

Fear and threat

- **Bureaucratic** (rule-oriented)

Protect the department

- **Generative** (performance-oriented)

Focus on the mission (alinhamento)

Cultura (Westrum, 2014)

Pathological	Bureaucratic	Generative
Low cooperation	Modest cooperation	High cooperation
Messengers “shot”	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

Cultura e desempenho de entrega



(FORSGREN *et al.*, 2018)

The three ways of DevOps

- Mapping the value stream for global optimization, not local optimization
- Amplifying continuous feedback loops to support necessary corrections
- Improving daily work through a culture promoting frequent experimentation, risk-taking, learning from mistakes, and knowing that practice and repetition are prerequisites to mastery

(KIM *et al.*, 2016)

The three ways of DevOps

- Mapping the value stream for global optimization, not local optimization
- Amplifying continuous feedback loops to support necessary corrections
- Improving daily work through a culture promoting frequent experimentation, risk-taking, learning from mistakes, and knowing that practice and repetition are prerequisites to mastery

Muito mais sobre cultura, né?

CAMS

(KIM *et al.*, 2016)

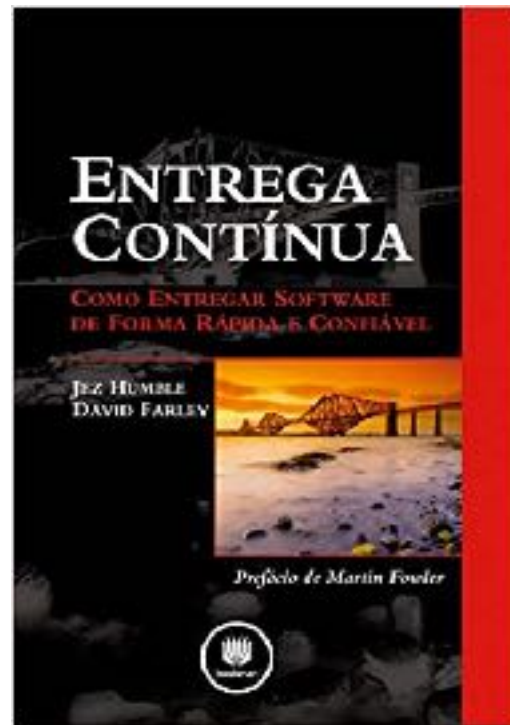
DevOps

DevOps is a collaborative and multidisciplinary effort within an organization **to automate continuous delivery of new software versions**, while guaranteeing their correctness and reliability

Entrega contínua

- Any software version committed to the repository must be a production-candidate version
- After passing through stages, such as compilation and automated tests, the software is sent to production by the press of a button

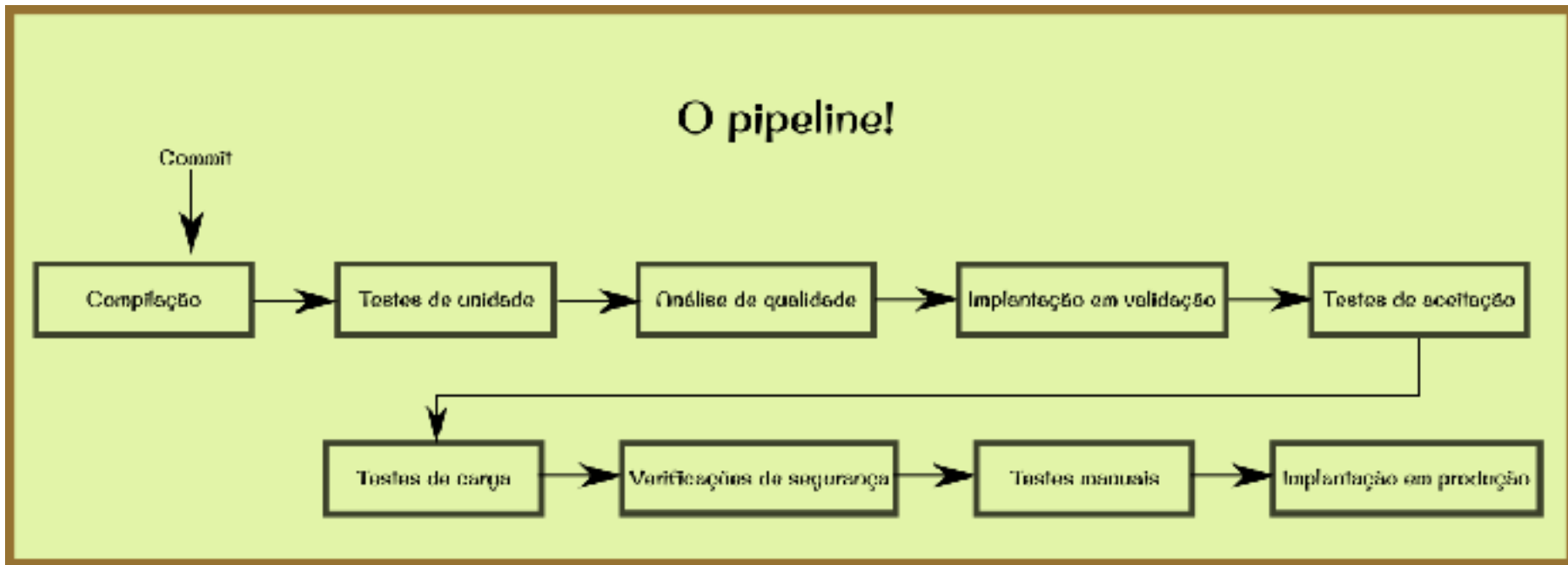
CAMS




Entrega contínua

O deploy se torna chato!

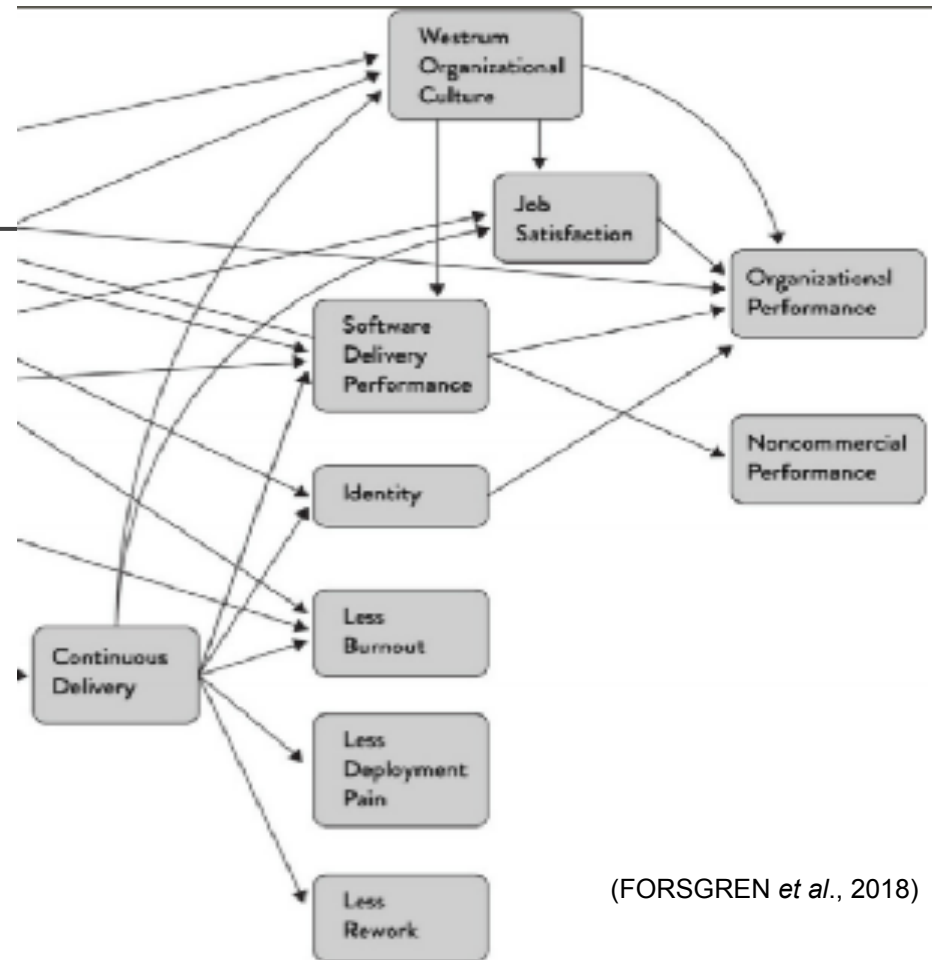
Entrega contínua



Entrega contínua

	#46520 by  latest	Ymaster  16df3408  1641872 Melhorias para ...		 00:09:14  3 days ago
	#46522 by  latest	Ymaster  246bb7ac  1641872 Melhorias para ...		 00:06:19  5 days ago
	#46517 by  latest	Ymaster  5ec9a87f  1641872 Melhorias para ...		 00:06:41  5 days ago

Entrega contínua



(FORSGREN *et al.*, 2018)

Delivery performance

- Frequência de implantação
- Tempo de entrega
- Tempo de recuperação
- Frequência de falhas

(FORSGREN *et al.*, 2018)

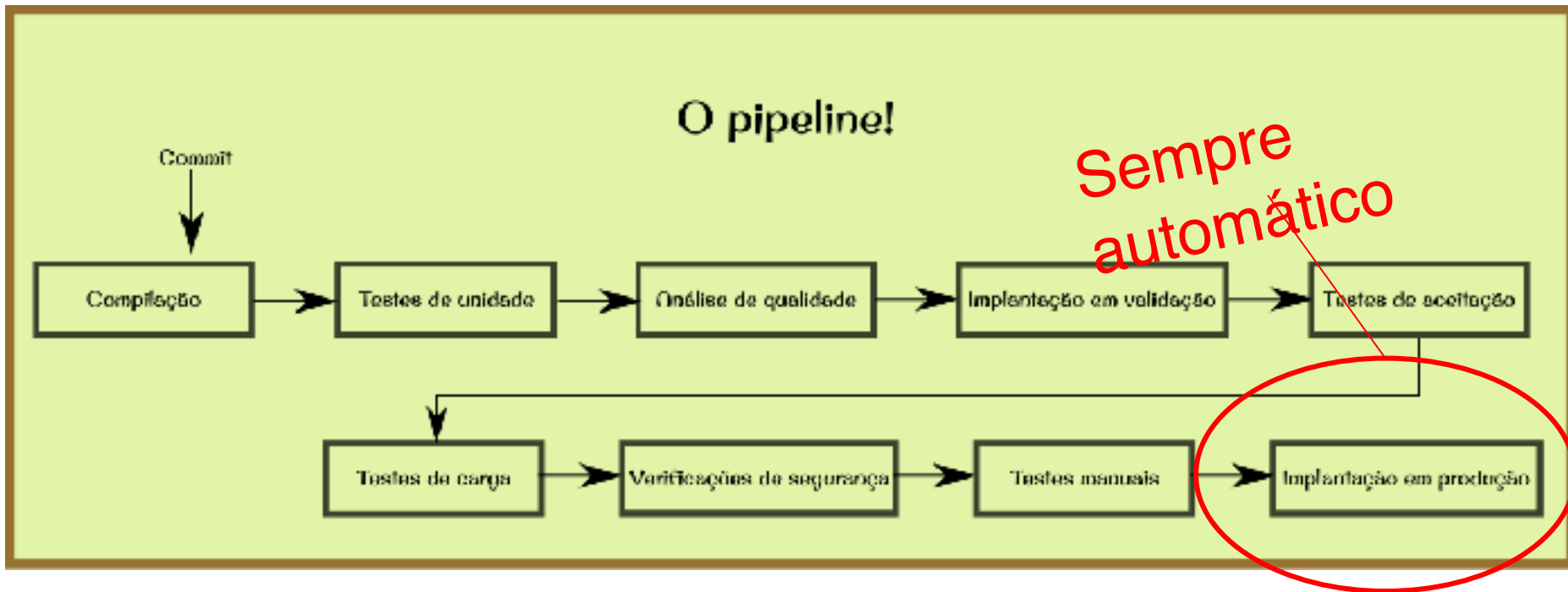
Delivery performance

- Frequência de implantação
- Tempo de entrega
- Tempo de recuperação
- Frequência de falhas

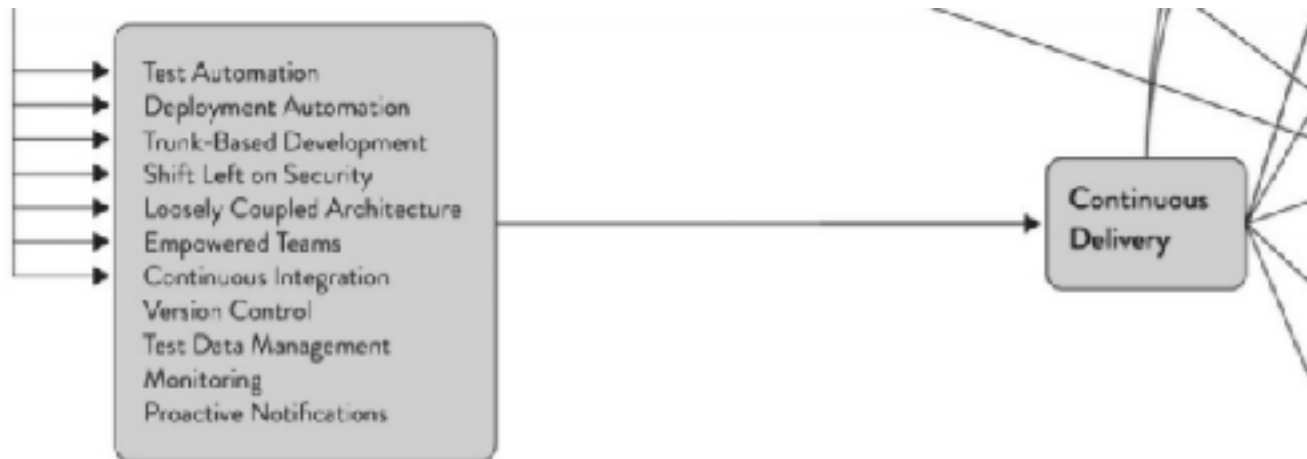
(FORSGREN *et al.*, 2018)

I.e.: o quão bem você consegue fazer entrega contínua

Implantação contínua



Práticas - Entrega Contínua



(FORSGREN *et al.*, 2018)

Integração contínua

- Integrate and test changes after no more than a couple of hours
- Team programming isn't a divide and conquer problem
- It's a divide, conquer, and integrate problem
- The integration step is unpredictable
- The longer you wait to integrate, the more it costs

(BECK and ANDRES, 2004)

O que seria não fazer integração contínua?



Integração contínua vs entrega contínua?



DevOps

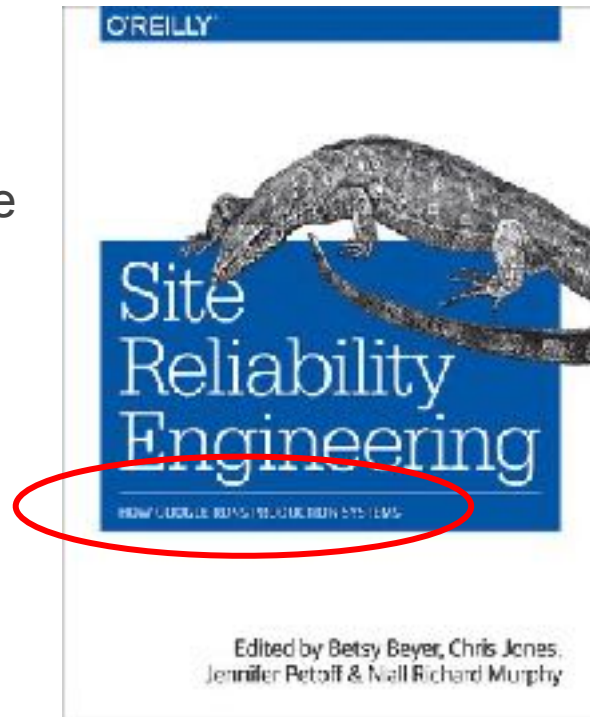
DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions, **while guaranteeing their correctness and reliability**

Como garantir correctness?



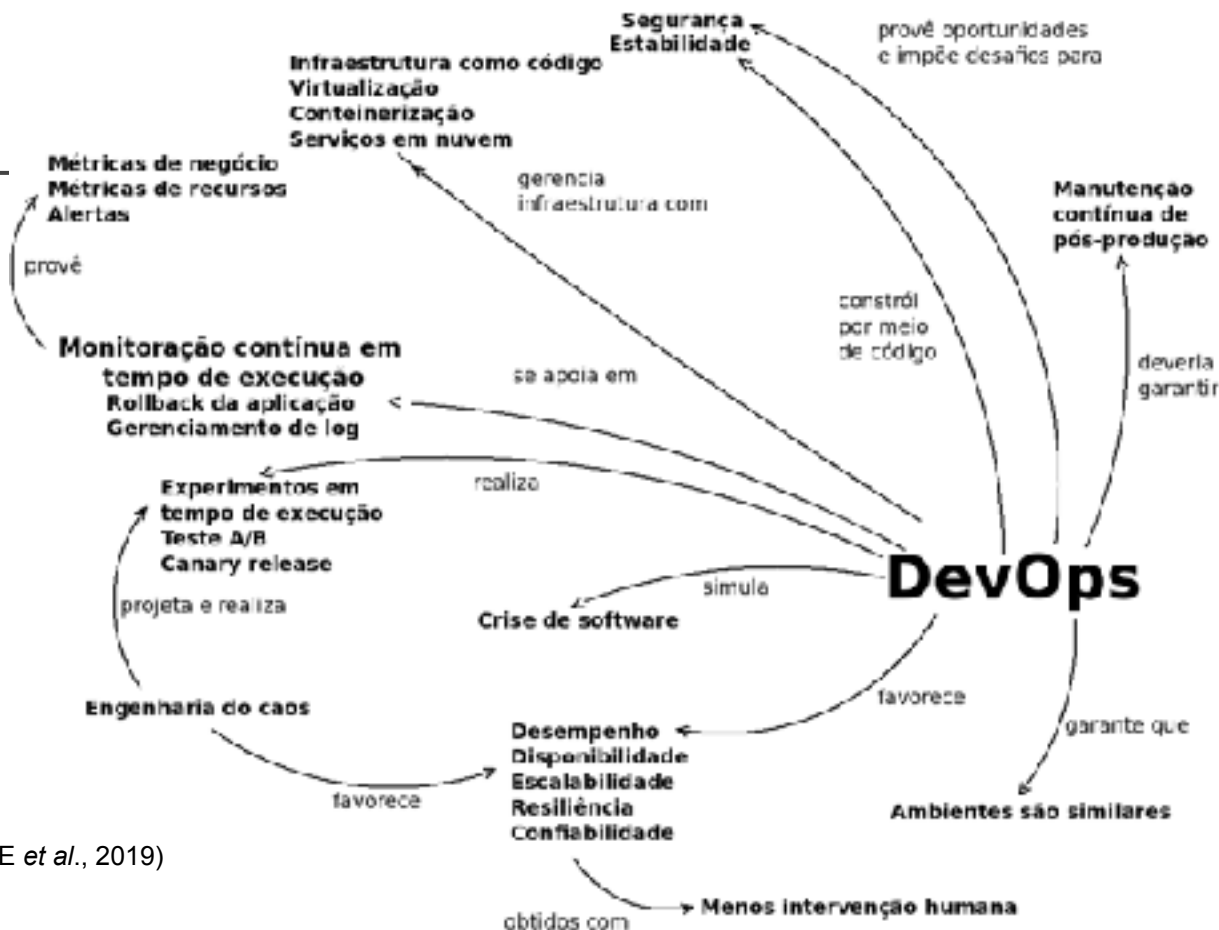
Reliability

- Reduce toil
- Infrastructure staff code to improve NFR
- Limit of 50% of operational work to the infrastructure staff
- Developers sustain the product for a period



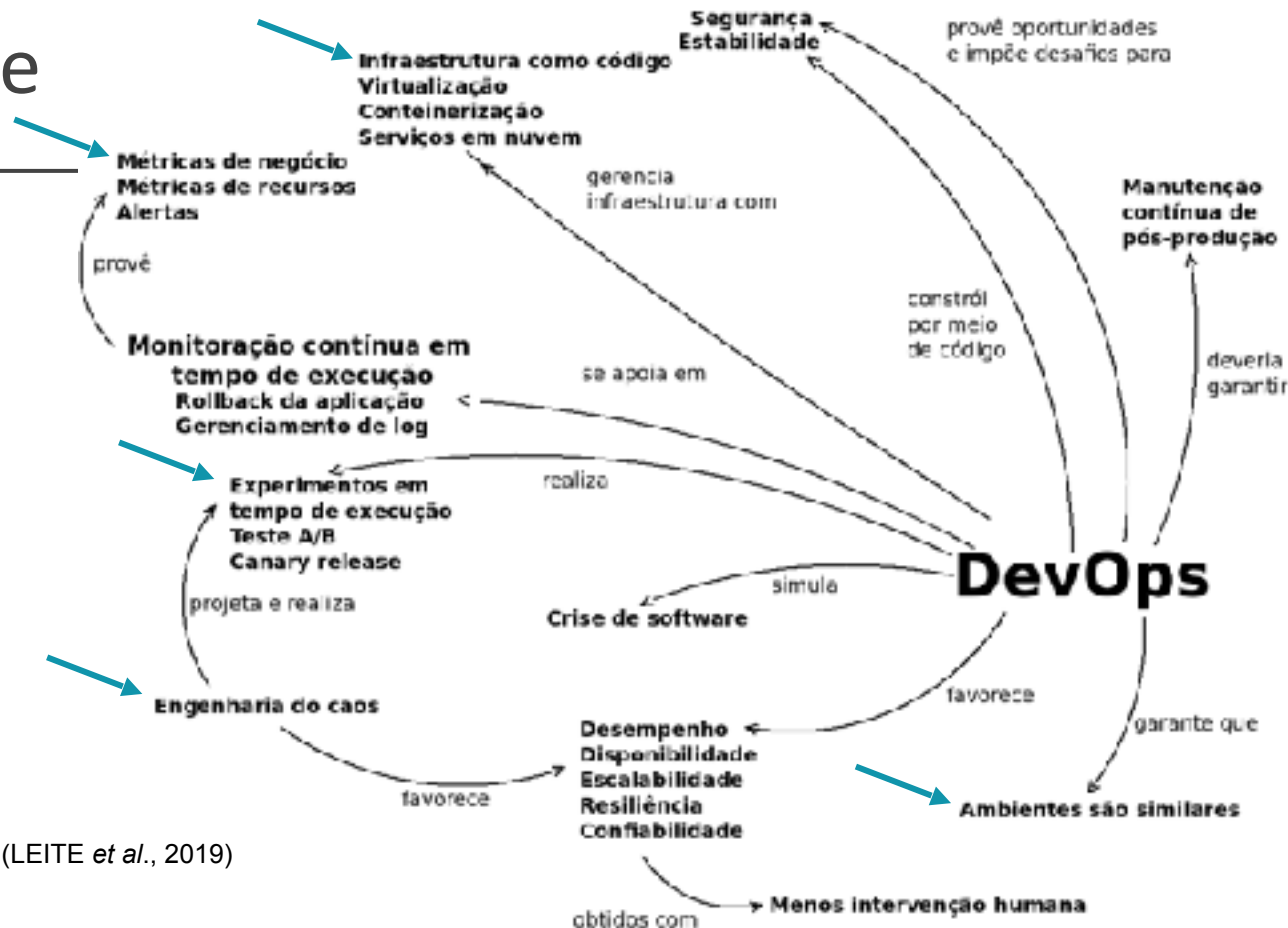
Práticas Runtime

CAMS



(LEITE *et al.*, 2019)

Práticas Runtime





<https://kahoot.it/>



Microserviços

- The microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API
- These services are built around business capabilities and independently deployable by fully automated deployment machinery

(LEWIS and FOWLER, 2014)



Por que microserviços precisam de DevOps?



Por que DevOps precisa de microsserviços?



Microserviços e DevOps

- Automação de deploy
- Entrega contínua com monolito é mais difícil
 - Build de 10 minutos
 - Log aggregator
 - Correlation ID's
 - Monitoração
 - Extra: automação da automação!

Ferramentas

[illegible]

Ferramentas

Tem que saber tudo isso para ser um DevOps?



PERIODIC TABLE OF DEVOPS TOOLS (v3)

The table is organized into groups based on tool categories, each represented by a different color:

- Build:** Jenkins, Maven, Gradle, etc.
- Configuration:** Ansible, Puppet, Chef, etc.
- Containerization:** Docker, Kubernetes, etc.
- CI/CD:** Jenkins, Travis CI, CircleCI, etc.
- Cloud:** AWS, Azure, Google Cloud, etc.
- Monitoring:** Prometheus, Grafana, etc.
- Security:** SonarQube, Snyk, etc.
- Collaboration:** Slack, Jira, etc.

Below the main table, there is a section for **XebiaLabs** with a link to [XebiaLabs DevOps Publication Guidelines](#).

Ferramentas

Table 4. Major DevOps tools related to actors and DevOps concepts

Category	Examples	Actors	Goals	Concepts
Knowledge sharing	Rocket Chat	Everyone	Human collaboration	Culture of collaboration
	GitLab wiki			Sharing knowledge
	Redmine			Breaking down silos
	Trello			Collaborate across departments
Source code management	Git	Dev / Ops	Human collaboration	Versioning
	SVN			Culture of collaboration
	CVS		Continuous delivery	Sharing knowledge
	ClearCase			Breaking down silos
Build process	Maven	Dev	Continuous delivery	Collaborate across departments
	Gradle			Release engineering
	Rake			Continuous delivery
	JUnit			Automation
	Sonar			Testing automation, Correctness
				Static analysis

(LEITE *et al.*, 2019)

Ferramentas

Continuous Integration	Jenkins GitLab CI Travis Nexus	Dev / Ops	Continuous delivery	Frequent and reliable release process Release engineering Continuous integration Deployment pipeline Continuous delivery, Automation Artifact management
Deployment automation	Chef, Puppet Docker Heroku Open Stack AWS Cloud Formation Rancher Flyway	Dev / Ops	Continuous delivery Reliability	Frequent and reliable release process Release engineering Configuration management Continuous delivery Infrastructure as code Virtualization, Containerization Cloud services, Automation
Monitoring & Logging	Nagios Zabbix Prometheus Logstash Graylog	Ops / Dev	Reliability	You built it, you run it After-hours support for Devs Continuous runtime monitoring Performance, Availability, Scalability Resilience, Reliability, Automation Metrics, Alerting, Experiments Log management, Security

(LEITE *et al.*, 2019)

Referências

Kent Beck, Cynthia Andres. **Extreme Programming Explained: Embrace Change**. 2 ed. Addison-Wesley Professional, 2004

Jez Humble, David Farley. **Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation**. 2010

Jez Humble and Joanne Molesky. **Why Enterprises Must Adopt Devops to Enable Continuous Delivery**. Cutter IT Journal. 2011

Referências

Ron Westrum. **The study of information flow:** A personal journey. Safety Science. 67, 58-63, 2014.

James Lewis and Martin Fowler. **Microservices**. 2014. <https://www.martinfowler.com/articles/microservices.html>.

Gene Kim, Jez Humble, Patrick Debois, John Willis, John Allspaw. **The DevOps Handbook:** How to Create World-Class Agility. 2016

Referências

Niall Richard Murphy, Betsy Beyer, Chris Jones, Jennifer Petoff. **Site Reliability Engineering: How Google Runs Production Systems**. 2016

Nicole Forsgren, Jez Humble, Gene Kim. **Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations**. 2018

Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. **A Survey of DevOps Concepts and Challenges**. ACM Computing Surveys, 52, 1-35, 2019.

Licença

Estes slides são concedidos sob uma Licença Creative Commons. Sob as seguintes condições:

Atribuição, Uso Não-Comercial e Compartilhamento pela mesma Licença.

Mais detalhes sobre essa licença em: creativecommons.org/licenses/by-nc-sa/3.0/

Leonardo Leite

<https://www.ime.usp.br/~leofl/>

<https://ccsl.ime.usp.br/devops/>

<https://twitter.com/leonardofl>

<https://www.linkedin.com/in/leonardo-alexandre-ferreira-leite-17194322>



Laboratório de Sistemas Computacionais Complexos

2020/04/16 - AULA 11

<https://uclab.xyz/sistemas-complexos-aula11>

Leonardo Leite
leofl@ime.usp.br

Renato Cordeiro Ferreira
renatocf@ime.usp.br

Thatiane de Oliveira Rosa
thatiane@ime.usp.br

João Francisco Daniel
joaofran@ime.usp.br

Alfredo Goldman
gold@ime.usp.br