Times de plataforma:
o estado-da-arte e da prática para acelerar a entrega contínua

Leonardo Leite
Serpro / USP

Leonardo Leite

Fabio Kon

Paulo Meirelles

Institute of
Mathematics and
Statistics (IME)

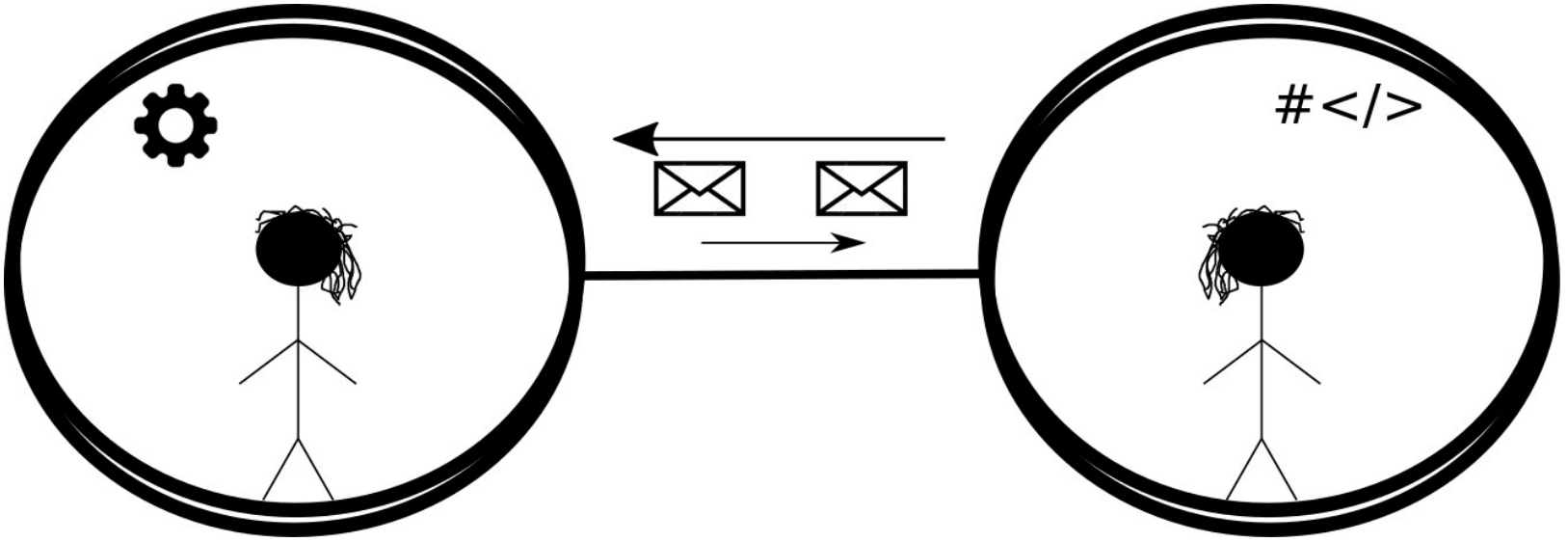
University of São Paulo
(USP)

Federal University of São Paulo
(Unifesp)

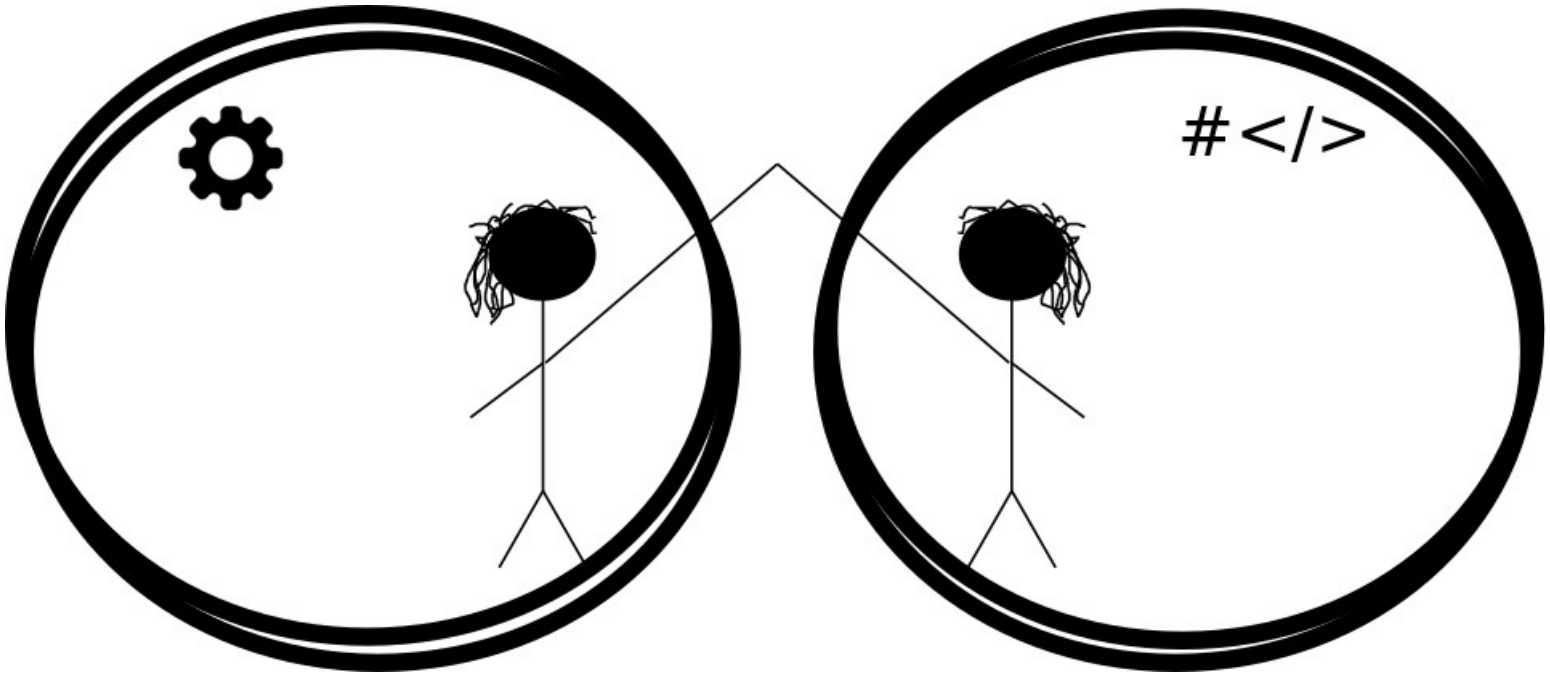

SERPRO
Soluções digitais que conectam
governo e sociedade

DevOps (?)

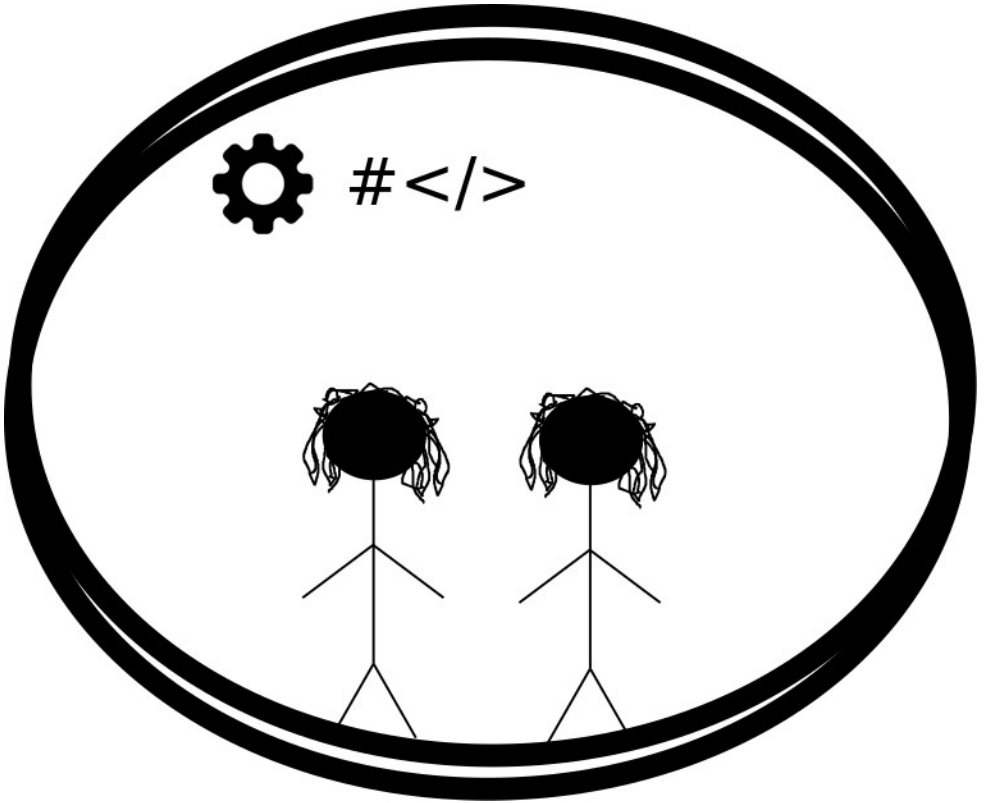
Departamentos em silos



Departamentos colaborando (DevOps clássico)



Times cross-funcionais



Questão de pesquisa

46 entrevistas

Entrevistas

- Papéis
  - 19 desenvolvedores
  - 7 gerentes de desenvolvimento
  - 3 gerentes de infraestrutura
  - 2 engenheiros de infraestrutura
  - 5 consultores externos
  - 1 gerente executivo
  - 1 membro de time de apoio
  - 1 designer

Entrevistas

- Países
  - 22 Brasil
  - 6 EUA
  - 4 Times distribuídos globalmente
  - 2 Alemanha
  - 2 Portugal
  - 1 França
  - 1 Canadá
  - 1 Itália

Entrevistas

- Gêneros
  - 29 Homens
  - 10 Mulheres

Entrevistas

- Anos desde a graduação
  - 4 com menos de 5
  - 5 entre 5 e 10
  - 11 com mais de 10

Entrevistas

- Tipos de organização
  - 34 privadas (visando lucro)
  - 3 governamentais
  - 1 privada (não visando lucro)
  - 1 organização internacional

Entrevistas

- Tamanhos das organizações
  - 4 com menos de 200 empregados
  - 9 entre 200 e 1000 empregados
  - 7 com mais de 1000 empregados

Entrevistas

- 5 unicórnios
- 3 tech giants

Entrevistas

- Domínios (setores) das organizações
  - IoT, finanças, defesa, administração pública, justiça, imóveis, mapas, educação, Internet, big data, pesquisa, seguro, nuvem, jogos, comércio eletrônico, telecomunicações, moda, relações internacionais, mobilidade, automação de escritório, consultoria de software, gerenciamento de estoque, automação veicular, gestão de equipe e suporte para desenvolvimento de software

Perguntas das entrevistas

- Responsabilidades

Perguntas das entrevistas

- Responsabilidades
  - Implantação

Perguntas das entrevistas

- Responsabilidades
  - Implantação
  - Construção de novos ambientes

Perguntas das entrevistas

- Responsabilidades
  - Implantação
  - Construção de novos ambientes
  - Requisitos não-funcionais

Perguntas das entrevistas

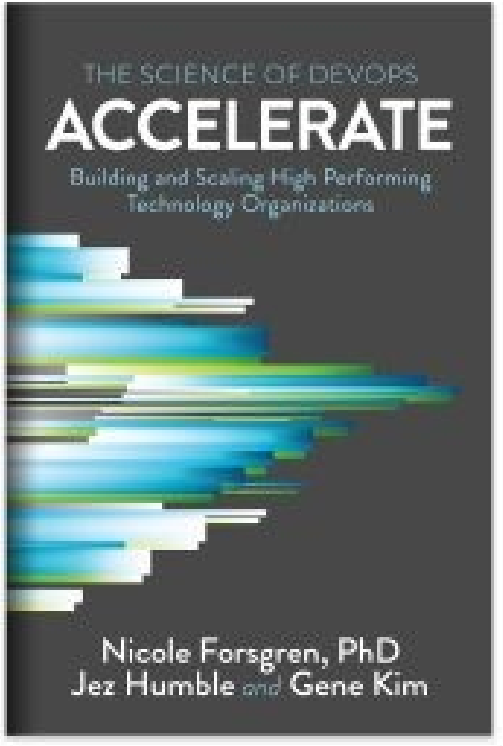
- Responsabilidades
  - Implantação
  - Construção de novos ambientes
  - Requisitos não-funcionais
  - Configuração e acompanhamento de monitoração

Perguntas das entrevistas

- Responsabilidades
  - Implantação
  - Construção de novos ambientes
  - Requisitos não-funcionais
  - Configuração e acompanhamento de monitoração
  - Tratamento de incidentes (principalmente após horário comercial)

Perguntas – desempenho de entrega

- Frequência de entrega
- Tempo do commit à produção
- Tempo médio de reparo



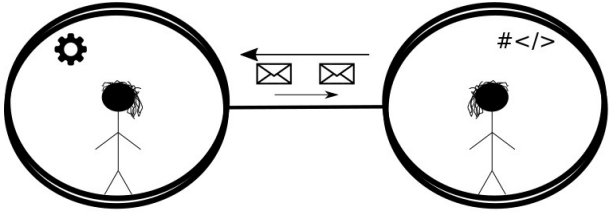
Desempenho de entrega – classificação

- Alto desempenho

  vs
- Sem alto desempenho

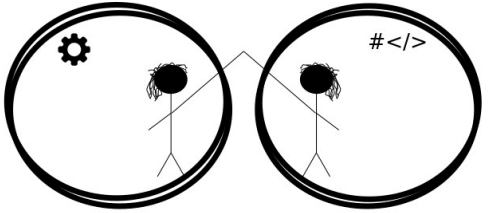
Encontramos...

1. Departamentos em silos



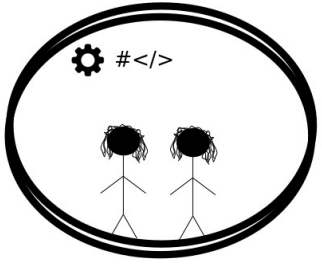
7

2. DevOps clássico

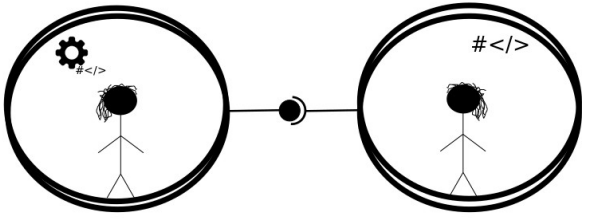


10

3. Times cross-funcionais

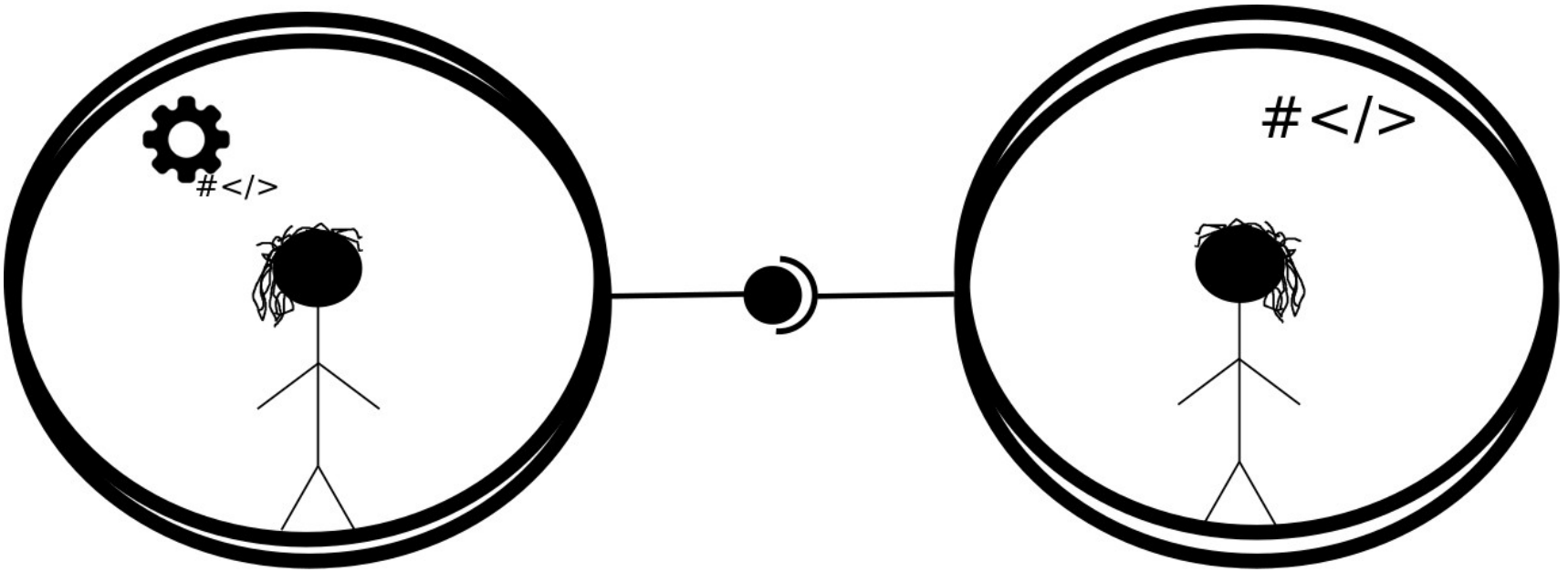


8



4

Times de plataforma



Times de plataforma são equipes de infraestrutura que fornecem serviços de infraestrutura altamente automatizados para empoderar os desenvolvedores para a implantação e operação de aplicações.

Exemplo ilustrativo – Estaleiro

Desempenho de entrega

Table 3: Organizational structures and delivery performance observed in our interviews.

Organizational structure	Delivery performance	Number of interviews
Siloed departments	Not-high	7
Classical DevOps	High	3
Classical DevOps	Not-high	7
Cross-functional	High	2
Cross-functional	Not-high	6
Platform team	High	4
Siloed departments *to* Classical DevOps	Not-high	4
Siloed departments *to* Cross-functional	High	1
Siloed departments *to* Platform team	High	1
Siloed departments *to* Platform team	Not-high	1
Classical DevOps *to* Cross-functional	High	1
Classical DevOps *to* Platform team	Not-high	1
Cross-functional *to* Platform team	Not-high	1

Todos com times de plataforma consolidados: alto desempenho de entrega

Desempenho de entrega

Table 3: Organizational structures and delivery performance observed in our interviews.

Organizational structure	Delivery performance	Number of interviews
Siloed departments	Not-high	7
Classical DevOps	High	3
Classical DevOps	Not-high	7
Cross-functional	High	2
Cross-functional	Not-high	6
Platform team	High	4
Siloed departments *to* Classical DevOps	Not-high	4
Siloed departments *to* Cross-functional	High	1
Siloed departments *to* Platform team	High	1
Siloed departments *to* Platform team	Not-high	1
Classical DevOps *to* Cross-functional	High	1
Classical DevOps *to* Platform team	Not-high	1
Cross-functional *to* Platform team	Not-high	1

Todos com times de plataforma consolidados: alto desempenho de entrega

Nenhuma outra estrutura com essa propriedade.

Desempenho de entrega

Table 3: Organizational structures and delivery performance observed in our interviews.

Organizational structure	Delivery performance	Number of interviews
Siloed departments	Not-high	7
~~Classical DevOps~~	~~High~~	~~3~~
Classical DevOps	Not-high	7
Cross-functional	High	2
Cross-functional	Not-high	6
Platform team	High	4
Siloed departments *to* Classical DevOps	Not-high	4
Siloed departments *to* Cross-functional	High	1
Siloed departments *to* Platform team	High	1
Siloed departments *to* Platform team	Not-high	1
Classical DevOps *to* Cross-functional	High	1
Classical DevOps *to* Platform team	Not-high	1
Cross-functional *to* Platform team	Not-high	1

Todos com departamentos em silos:

Sem alto desempenho de entrega

Desempenho de entrega

Table 3: Organizational structures and delivery performance observed in our interviews.

Organizational structure	Delivery performance	Number of interviews
Siloed departments	Not-high	7
Classical DevOps	High	3
Classical DevOps	Not-high	7
Cross-functional	High	2
Cross-functional	Not-high	6
Platform team	High	4
Siloed departments *to* Classical DevOps	Not-high	4
Siloed departments *to* Cross-functional	High	1
Siloed departments *to* Platform team	High	1
Siloed departments *to* Platform team	Not-high	1
Classical DevOps *to* Cross-functional	High	1
Classical DevOps *to* Platform team	Not-high	1
Cross-functional *to* Platform team	Not-high	1

Todos com departamentos em silos:

Sem alto desempenho de entrega

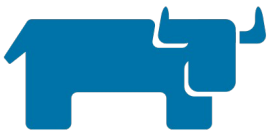
Exceto um:

transitando para time de plataforma

Ter uma equipe de plataforma é uma forma promissora de se alcançar um alto desempenho de entrega.

Tipos de plataforma

Plataforma de código aberto implantada internamente



RANCHER

kubernetes

Data center

Plataforma privada customizada

Platform

Data center

Plataforma de fachada para nuvem pública

Platform

Plataforma de fachada para nuvem pública

- Pra quê a plataforma?

Plataforma de fachada para nuvem pública

- Pra quê a plataforma?
- Centenas de serviços da nuvem

Plataforma de fachada para nuvem pública

- Pra quê a plataforma?
- Centenas de serviços da nuvem
- Padronização corporativa

Tipo de plataforma – Estaleiro

- ~~Plataforma de fachada para nuvem pública~~

Tipo de plataforma – Estaleiro

- ~~Plataforma de fachada para nuvem pública~~
- ~~Plataforma de código aberto implantada internamente~~

Estaleiro criado no fim de 2016.

Tipo de plataforma – Estaleiro

- ~~Plataforma de fachada para nuvem pública~~
- ~~Plataforma de código aberto implantada internamente~~
- Plataforma privada customizada

Projeto Estaleiro

SERPRO

Eu não sei nada de

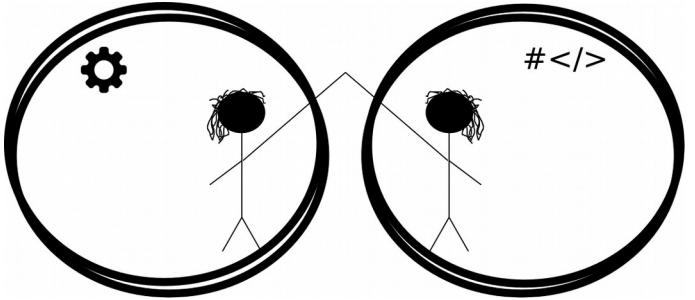
Kubernets!

Propriedades de organizações

com times de plataforma

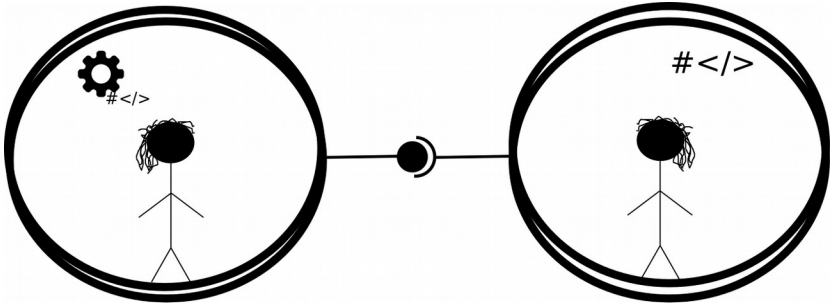
Desenvolvedores operam seus serviços!

Tratamento de incidentes



DevOps clássico

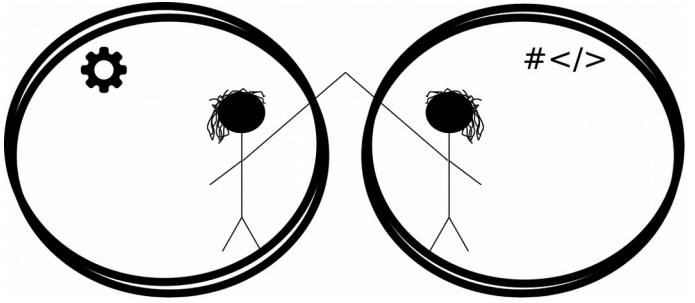
VS



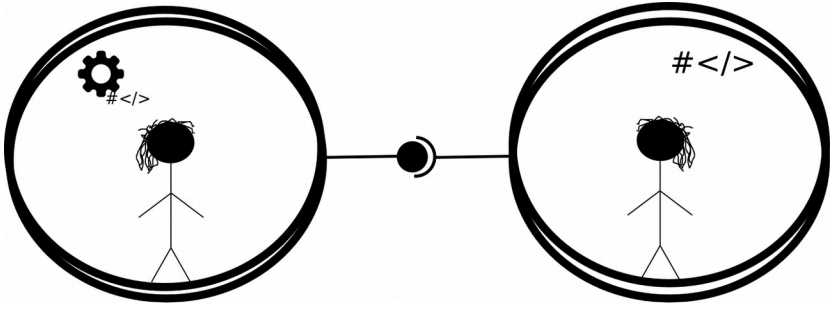
Times de plataforma

Tratamento de incidentes

VS



DevOps clássico



Times de plataforma

RNFs tratados pela plataforma

RNFs tratados pela plataforma

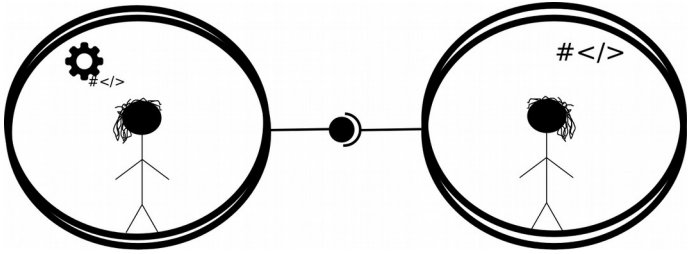
- Balanceamento de carga
- Auto scaling
- Throttling
- Monitoração

RNFs tratados pela plataforma

- Balanceamento de carga
- Auto scaling
- Throttling
- Monitoração

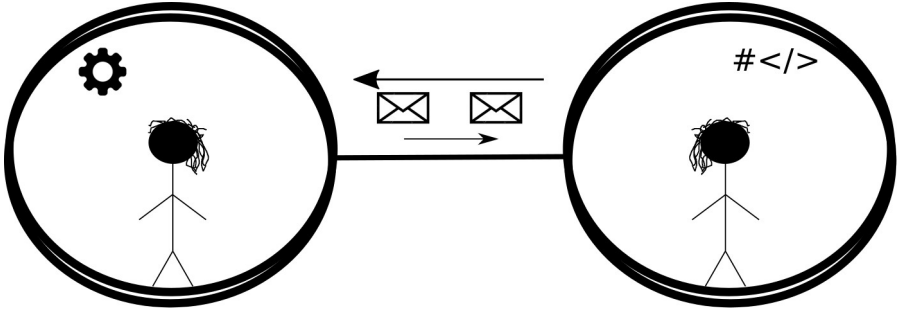
Suporte e colaboração para RNF

Suporte e colaboração

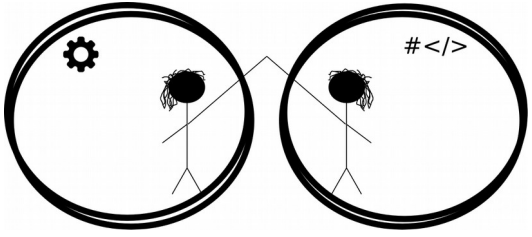


Times de plataforma

VS

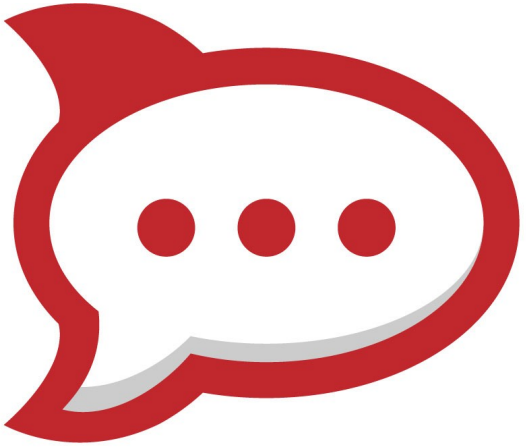


Departamentos em silos



DevOps clássico

#estaleiro



ROCKET.CHAT

As equipes de produto se tornam desacopladas do pessoal de infraestrutura.

Comunicação acontece para...

Comunicação acontece para...

- Investigações para resolução de incidentes

Comunicação acontece para...

- Investigações para resolução de incidentes
- Consultoria para desenvolvedores

Comunicação acontece para...

- Investigações para resolução de incidentes
- Consultoria para desenvolvedores
- Demandas dos desenvolvedores

Comunicação acontece para...

- Investigações para resolução de incidentes
- Consultoria para desenvolvedores
- Demandas dos desenvolvedores

Investigações – Estaleiro

Investigações – Estaleiro

- Histórico de métricas no Grafana não persistido

Investigações - Estaleiro

- Histórico de métricas no Grafana não persistido
- Troca de certificados

Consultorias - Estaleiro

Consultorias – Estaleiro

- Histórico do Grafana
  - "*É só clicar aqui e ali que aparece*", João Morais

Consultorias - Estaleiro

- Troca de certificados
  - "*O que deve estar ocorrendo no cenário de vocês é reuso de conexão, algo que vem muito forte com http2. Quando você reusa conexão, você não faz handshake tls, reusa a extensão sni da conexão antiga, e reusa certificados que por ventura haviam sido enviados. Para o caso do haproxy baixamos o protocolo para http1.1 de modo que esse reuso de conexão seja feito apenas em casos mais específicos, tal como keep alive e websocket.*", João Morais

Demandas - Estaleiro

Demandas - atendidas

- Retorno 429 para conexões excedidas (antes dava 503)

Demandas - atendidas

- Retorno 429 para conexões excedidas (antes dava 503)
- Download do truststore

Demandas - ainda não atendidas

- Especificar mais parâmetros de deploy as code

Demandas - ainda não atendidas

- Especificar mais parâmetros de deploy as code
- Histórico de alterações de variáveis de ambiente (quem alterou quando alterou) acessível ao desenvolvimento

Demandas - recusadas

- Adicionar range de IPs ao whitelist default

Iniciativas Estaleiro

- Queremos te ouvir...

Iniciativas Estaleiro

- Queremos te ouvir...
- Pesquisa de opinião de satisfação

Iniciativas Estaleiro

- Queremos te ouvir...
- Pesquisa de opinião de satisfação
- Pesquisa de opinião sobre próximas funcionalidades

Iniciativas Estaleiro

- Queremos te ouvir...
- Pesquisa de opinião de satisfação
- Pesquisa de opinião sobre próximas funcionalidades
- Comunicados
  - Ex: atualizações de segurança

Conflitos - Estaleiro

Conflitos – Estaleiro

- Instalação de módulo no sistema operacional (do contêiner)

Conflitos – Estaleiro

- Instalação de módulo no sistema operacional (do contêiner)
- Conflito a priori, não a posteriori

Conflitos - Estaleiro

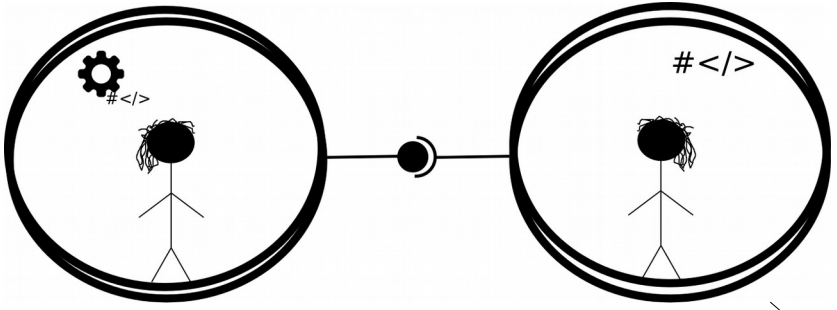
- Instalação de módulo no sistema operacional (do contêiner)
- Conflito a priori, não a posteriori
- "*Conflitos sempre existirão, o que importa é a qualidade do conflito*"

A equipe de infraestrutura não é mais solicitada para tarefas operacionais.

Tickets administrativos - Estaleiro

- Criação de sistemas
- Requisitar acessos (console, monitoração, logs)
- Preenchimento de variáveis sigilosas
- Gerenciamento de quota
- Regras de acesso para fora do Estaleiro

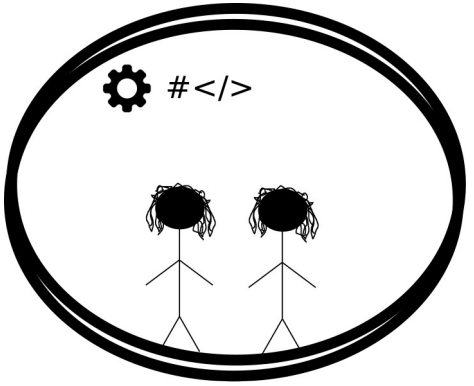
Em geral, as equipes de produto não precisam possuir especialistas em infraestrutura.



Times de plataforma

Não se espera que sejam experts em infra

VS



Times cross-funcionais

Deveriam ter especialistas em infra

Na DGTR1...

Na DGTR1...

Os especialistas em infraestrutura possuem habilidades de codificação.

Os especialistas em infraestrutura possuem habilidades de codificação.

Principalmente para plataformas privadas customizadas.

Os especialistas em infraestrutura possuem habilidades de codificação.

"João Morais desenvolveu o componente de roteamento do Estaleiro (Ingress HAProxy- "a Kubernetes ingress controller") que é amplamente usado no mercado hoje em dia (o código foi aberto) e é o único que atende às necessidades de throughput do Serpro."

Katz

https://github.com/jcmoraisjr/haproxy-ingress

Para saber mais sobre times de plataforma...

Platform Teams: An Organizational Structure for Continuous Delivery

Leonardo Leite, Fabio Kon
University of São Paulo, Brazil
leofl@ime.usp.br,kon@ime.usp.br

Gustavo Pinto
Federal University of Pará, Brazil
gpinto@ufpa.br

Paulo Meirelles
Federal University of São Paulo, Brazil
paulo.meirelles@unifesp.br

ABSTRACT

Software-producing organizations are seeking to release faster and more efficiently new versions of their products to their customers to remain competitive in the fierce software market. Continuous delivery practices arise as a potential solution since every commit to the repository could result in a production-candidate version of a product, accelerating time to market, and improving customer satisfaction. In this work, we employed Grounded Theory to investigate how organizations pursuing continuous delivery should organize their development and operations teams. We collected data from 27 IT professionals. After a careful analysis, we started the elaboration of a taxonomy with four patterns of organizational structures: (1) siloed departments, (2) classical DevOps, (3) cross-functional teams, and (4) platform teams. We observed that the platform team structure is the most distinctive classification of our taxonomy, and it has promising results regarding delivery performance. Some relevant aspects we found out about platform teams include: infrastructure specialists need coding skills; product teams have to operate their business services; and much of the non-functional concerns are handled by the platform, alleviating product teams.

CCS CONCEPTS

• **Software and its engineering → Software development process management**; **Programming teams**; *Software post-development issues.*

KEYWORDS

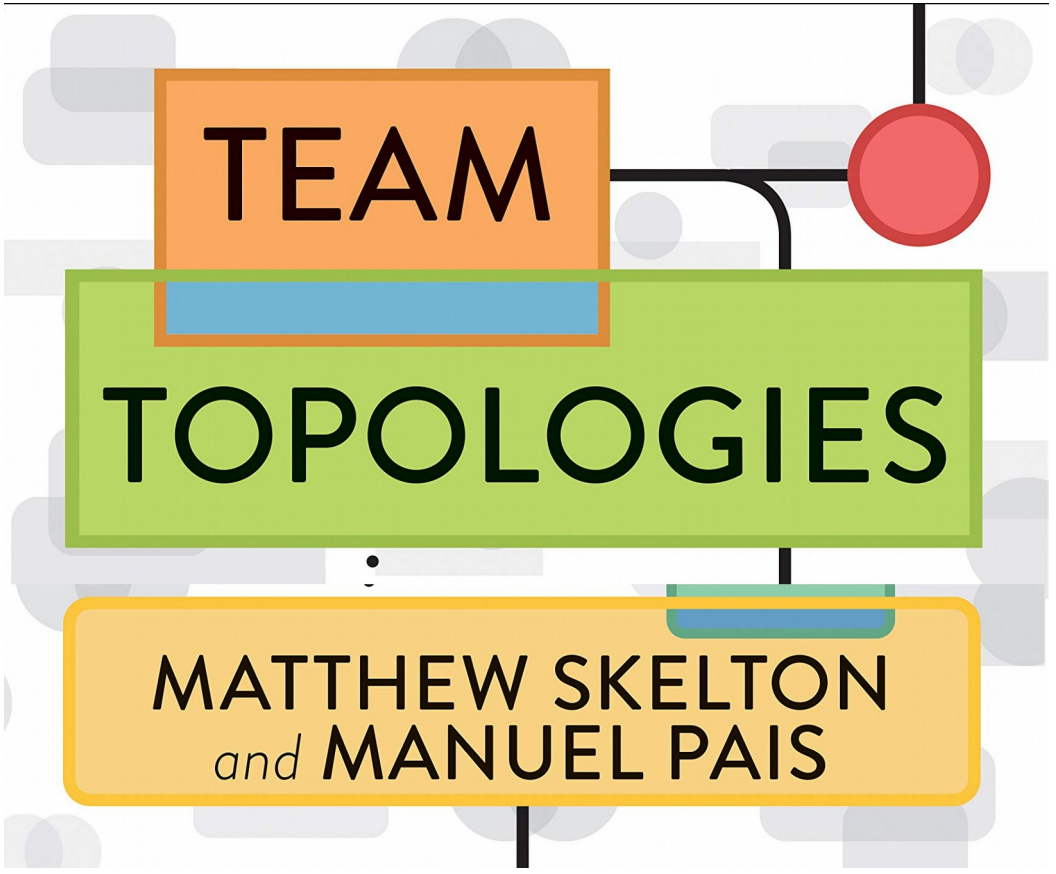
Continuous Delivery, Release Process, DevOps, Software Teams

their benefits, such as accelerated time to market [3]. The automation that continuous delivery introduces creates a profound impact on various aspects of the software engineering practice (e.g., development, testing, deployment) [23]; it also impacts the organizational structure [3] since release activities involve many divisions of a company (e.g., developers, operations, and business).

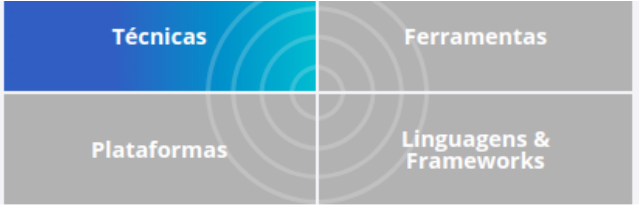
Therefore, organizations moving toward continuous delivery have not only to upgrade their software tooling arsenal but also find ways to better shape and integrate their IT teams. Such integration can occur according to different patterns that we call *organizational structures*. However, there is no substantial literature tackling how organizations should structure their teams to excel in the context of continuous delivery. This lack of research is particularly unfortunate due to at least two crucial reasons: *(i)* organizations wishing to adopt continuous delivery can be disoriented regarding how to design their human resources structure toward this goal; *(ii)* once a structure is chosen, the organization might be unaware of the consequences of this choice. The existing literature presents some classifications for organizational structures [15, 18, 25]. Although some of these works are empirical studies, the presented options of structures are an arbitrary start point for them, lacking empirical elaboration.

To mitigate this gap, our research efforts employ empirical methods to answer our main research question *"which organizational structures are software-producing organizations adopting for managing IT technical teams in a continuous delivery context?"* In this short paper, containing preliminary results, we partially answer our research question by presenting the most distinctive of these structures.

http://ccsl.ime.usp.br/devops/2020-03-17/platform-teams.html


TEAM
TOPOLOGIES
MATTHEW SKELTON
and MANUEL PAIS

TECHNOLOGY RADAR 2020



As informações desta página não estão completamente disponíveis no seu idioma de escolha. Esperamos disponibiliza-las integralmente em outros idiomas em breve. Para ter acesso às informações no idioma de sua preferência, faça o download do PDF **aqui**.

Técnicas

Times de produto de engenharia de plataforma

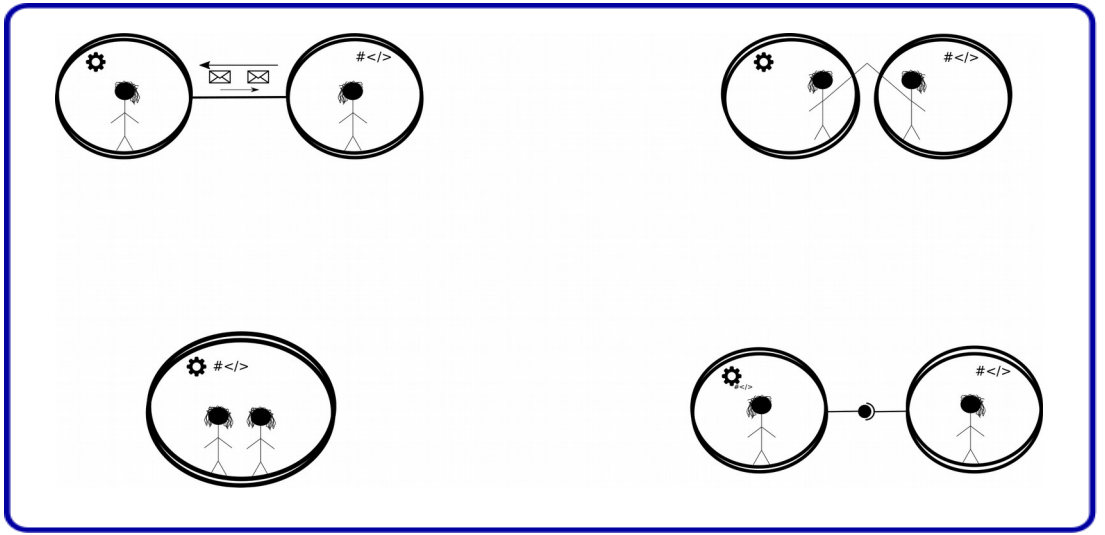
MAY 2020

EXPERIMENTE

A adoção de nuvem e DevOps, embora aumente a produtividade dos times que agora podem se mover mais rapidamente com dependência reduzida de infraestrutura e times de operações centralizados, também restringiu os times que não possuem as habilidades necessárias para autogerenciar uma stack completa de aplicativos e operações. Algumas organizações enfrentaram esse desafio criando **times de produto de engenharia de plataforma**. Esses times mantêm uma plataforma interna que permite aos times de entrega implantar e operar sistemas com prazo de entrega reduzido e complexidade de stack. A ênfase aqui está nas ferramentas de autoatendimento e suporte orientadas à API, com os times de entrega ainda responsáveis por dar suporte ao que implementam na plataforma. As organizações que consideram estabelecer um time de plataforma devem ter muito cuidado para não criar acidentalmente um **time separado de DevOps**, nem devem simplesmente renomear sua **estrutura existente de hospedagem e operações** como uma plataforma. Se você está se perguntando qual é a melhor configuração para os times de plataforma, usamos os conceitos de **topologias de time** para dividir os times de plataforma em nossos projetos em times de enablement, times de "plataforma em uma

ccsl.ime.usp.br/devops

As equipes de plataforma são equipes de infraestrutura que fornecem serviços de infraestrutura altamente automatizados para capacitar os desenvolvedores para a implantação de aplicativos.



ime.usp.br/~leofl

@leonardofl