

# DEVELOPER EXPERIENCE

Desbloqueando entregas ágeis

Olá pessoal

De onde vem a necessidade de entender  
experiência de desenvolvimento?

- Uma introdução ao conceito
- Um case de sucesso
- Outros exemplos surgindo no mercado

**DE ONDE VEM A NECESSIDADE DE  
ENTENDER A  
EXPERIÊNCIA DE DESENVOLVIMENTO?**

Primeiro precisamos de falar sobre  
Agilidade e Entrega Contínua

# AGILIDADE E LEAN

Construção colaborativa, evitando desperdício

# Processos para responder rápido a necessidades de negócio

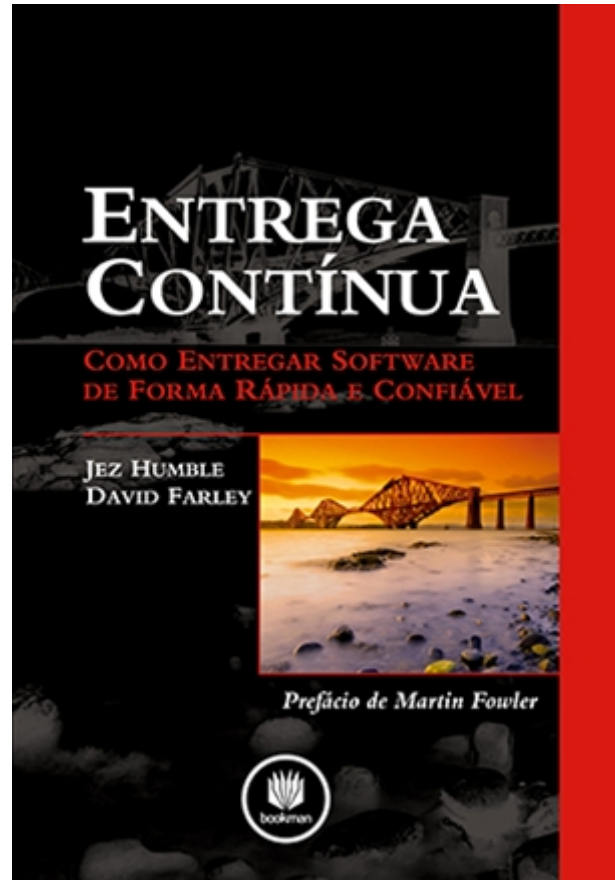


# ENTREGA CONTÍNUA

Reduzindo riscos por entregar frequentemente

Ter um processo de entrega confiável e bem definido,  
onde o produto passa por *build, test, deploy*

## ⇧ Entrega Continua - Jez Humble & David Farley



# LEAD TIME

Quanto tempo para fazer uma mudança e tê-la refletida em produção

**ENTREGA CONTÍNUA É PRÉ-REQUISITO  
PARA BOAS ENTREGAS**

# COMO CRESCER E MANTER UMA ENTREGA CONTÍNUA?

10 pessoas desenvolvendo

Manter o **contexto do negócio** dentro da equipe é simples



As ferramentas e processos são simples

**10 EQUIPES DE 10 PESSOAS  
DESENVOLVENDO**

A complexidade de negócio que esse grupo suporta  
cresce

As mudanças possuem **riscos e impacto maiores**

As ferramentas e processos precisam se adaptar ao crescimento da complexidade

# 10 PESSOAS X 100 PESSOAS

- Qual o risco de uma mudança no código?
- Qual o risco de uma mudança no banco de dados?
- Qual o risco de uma mudança na plataforma?

Falamos sobre a **jornada do usuário** para entender como é interagir com nosso sistema

E se...



E se pensarmos nas **pessoas que desenvolvem** os  
nossos sistemas como Usuários dos nossos **processos**  
**de entrega?**

# QUAL A EXPERIÊNCIA DE DESENVOLVIMENTO?

Qual a **experiência de uso** que nossas ferramentas e processo de entrega tem?

Qual o **custo e esforço** ao fazer:

- Uma mudança na lógica do código?
- Uma mudança no banco de dados?
- Uma mudança na infraestrutura?

Quais são os **comportamentos** que o nosso processo de entrega **incentivam** nas nossas equipes?

Quanto maior o esforço, menos o uso

**DEVELOPER EXPERIENCE BUSCA INCENTIVAR A  
ENTREGA CONTÍNUA**

Developer Experience é **otimizar** Entrega Contínua



É preciso **reduzir os custos** dos processos para  
**incentivar** Entrega Contínua

**QUAIS OS COMPORTAMENTOS QUE GOSTARÍAMOS DE  
INCENTIVAR NAS EQUIPES?**

# CASE

Incentivando **pequenas mudanças** em bancos de dados entre 25 equipes

## Cenário inicial

Uma mudança de banco de dados precisa:

- Ser revisado pelo DBA
- Não podemos permitir comandos que percam dados (Drop, deletes)
- Não podemos permitir comandos que geram locks nas tabelas
- Mudanças estruturais precisam de cuidado redobrado

## Comportamentos adotados

- Mudanças demoraram 5 dias para serem revisadas pelo DBA
- Conflitos nas prioridades atrasavam as revisões
- Equipes evitavam fazer mudanças no banco de dados
- Falta de confiança por não saberem o que é aceitável
- Falta de previsibilidade quando a mudança seria aplicada em produção
- Mudanças aconteciam no final da sprint, gerando dependência no código que seria entregue

Mudanças em banco eram evitadas e deixados para o  
último dia da sprint

Esse não é o comportamento que gostaríamos de incentivar

## Juntamos uma equipe para:

- Automatizamos a maioria das checagens nas mudanças
- Melhoramos a mensagem de feedback quando encontramos erros
- Reduzimos o ciclo de feedback permitindo executar as verificações antes mesmo de criar um ticket
- Nos casos que ainda não haviam verificações automáticas, enviamos para o DBA nos ajudar



Mudanças foram de 5 dias de revisão para 1 hora

*Lead Time* foi de de 40 horas para 1 hora

# **DEVELOPER EXPERIENCE EM OUTRAS EMPRESAS**

# **NUBANK: ENGINEERING PRODUCTIVITY**

## (EngProd)

**EM OUTROS LUGARES, COM OUTROS NOMES**

- ThoughtWorks sobre ↗ Developer Experience como diferenciador
- Heroku sobre ↗ Developer Experience como oferta de mercado
- Google sobre ↗ Engineering Productivity na qualidade dos sistemas
- Netflix sobre ↗ Integracao Continua por membros da equipe de Developer Productivity

# SE VOCÊ AINDA NÃO PRÁTICA ENTREGA CONTÍNUA

- Faça um exercício de ↗avaliação de entrega continua
- Crie um ↗Value Stream Map
- Descubra qual seu ↗Lead Time
- Adote mudanças no processo para torná-lo confiável e automatizável

Uma **boa experiência** de desenvolvimento incentiva **mudanças menores e mais frequentes**, com riscos menores para o seu negócio



Podemos conversar mais durante a conferência

Bruno Tavares -  @bltavares

# UM EXERCÍCIO PARA SUA EQUIPE

Para conhecer a **ergonomia** do seu processo de desenvolvimento

Quanto **tempo demora** para uma pessoa chegar no projeto e mudar algo em **produção**?

# O IDEAL:

## No mesmo dia

- Escolha uma mudança de baixo risco
  - Uma mudança em um texto é um ótima primeira tarefa
- Acompanhe a configuração do ferramental
- Mostre o caminho para produção
- Vejam a mudança em produção

**PARA QUEM ESTÁ CHEGANDO NA EQUIPE**

É o melhor momento para ser apresentado a um processo que **você usará todo os dias**

As próximas mudanças aconteceram em um  
ambiente já configurado

No fim do dia você sente que já **contribuiu no projeto**



# PARA QUEM ESTÁ ACOMPANHANDO O ONBOARDING

Você descobre onde **de uso no processo de entrega**

Você observa o que é preciso otimizar para  
reduzir o Lead Time

E já começa com um bom onboarding

Obrigado

Bruno Tavares - @bltavares