



TÉCNICAS ÁGEIS DE PROGRAMAÇÃO

Daniel Wildt – Aula 01

Professores

DANIEL WILDT

Professor Convidado

Profissional de tecnologia preocupado com desenvolvimento de produtos e serviços com equipes focadas em aprendizado, melhoria contínua e autonomia. Mentora e produz conteúdo em vídeo, áudio e texto sobre: consciência de tempo, experiência de usuário, empreendedorismo e metodologias ágeis. Sócio e mentor na Wildtech, Blogger/YouTuber no danielwildt.com, sócio e diretor na uMov.me.

GUILHERME LACERDA

Professor Convidado

Graduado em Informática pela Universidade da Região da Campanha (2000). Mestre em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (2005). Atualmente, cursa Doutorado em Ciência da Computação na UFRGS, na área de Engenharia de Software. Consultor/Instrutor associado da Wildtech, trabalhando com coaching e mentoring nas áreas de Engenharia de Software, Gerência de Projetos e Produtos e Metodologias Ágeis (eXtreme Programming, SCRUM, Lean). Possui mais de 20 anos de experiência em desenvolvimento de software. Atuou por vários anos como analista/projetista/desenvolvedor de software. Possui as certificações de SCRUM Master (SCM) e SCRUM Professional (CSP) pela SCRUM Alliance. Membro do IASA (International Association of Software Architects). Fundador do Grupo de Usuários de Métodos Ágeis (GUMA), vinculado a SUCESU-RS. É docente de graduação (Ciência da Computação, Análise e Desenvolvimento de Sistemas e Sistemas de Informação, Gestão de TI - Unisinos) e pós-graduação (Engenharia de Software, Desenvolvimento de Aplicações Móveis - Unisinos e Desenvolvimento Full Stack - PUCRS).

Professores

MICHAEL DA COSTA MÓRA

Professor PUCRS

Graduado em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (UFRGS), mestre em Computação e doutor em Ciência da Computação pela mesma universidade. Professor-adjunto do Instituto de Informática. Tem experiência na área de ciência da computação com ênfase em inteligência artificial, atuando principalmente nos seguintes temas: inteligência artificial, aprendizagem de máquina, agentes inteligentes e sistemas multiagentes, engenharia de software e desenvolvimento de sistemas, ensino de programação e de ciência da computação.

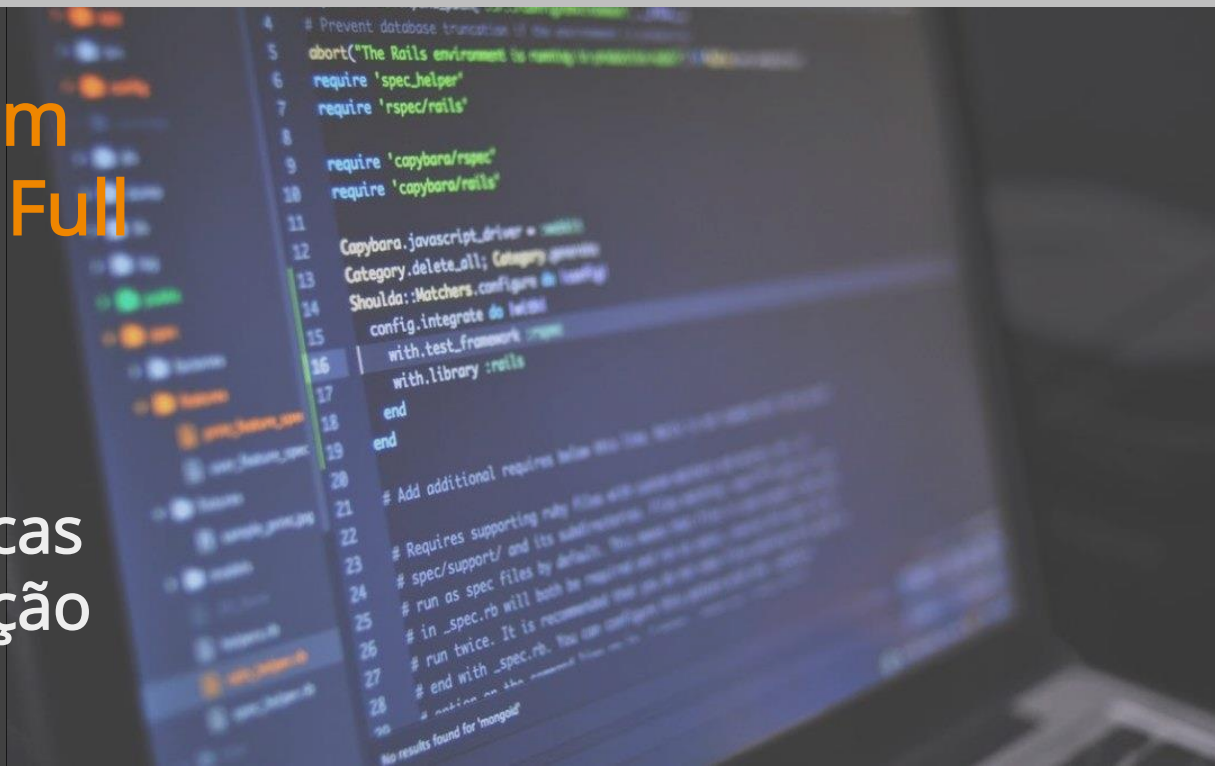
Ementa da disciplina

Fundamentos da agilidade: primórdios, manifesto ágil, princípios da agilidade. Panorama das metodologias ágeis. Extreme programming: características, valores, práticas, as práticas na prática. Test driven development (TDD): origens, codificar – testar – projetar, benefícios e armadilhas, variações, TDD na prática. Behaviour driven design (BDD): origens e princípios, BDD x TDD, benefícios e armadilhas, BDD na prática.



Especialização em Desenvolvimento Full Stack

Disciplina de Técnicas
Ágeis de Programação





Daniel Wildt

- Empresário (umov.me, wildtech, treinebasquete, mais.dev, desenlatar)
- Mentor, Dev, Baterista
- [site](#) | [twitter](#)



Guilherme Lacerda

- SW Engineering Lead (umbler)
- Mentor (wildtech)
- Professor Universitário (unisinos)
- [site](#) | [twitter](#)

Parte 1

Agenda - Introdução

Parte 1

1. Manifesto Ágil
2. Princípios e o dia a dia da equipe
3. Valores

*"A questão não é SE a qualidade técnica é importante, mas **COMO** a qualidade técnica será alcançada"*

Jim Highsmith

*Agile Project Management: creating innovative products,
Addison-Wesley Professional (2004)*

[ver detalhes do livro](#)

EXTREME PROGRAMMING BRASIL

Primeiro Congresso Brasileiro de Metodologias Ágeis de Software

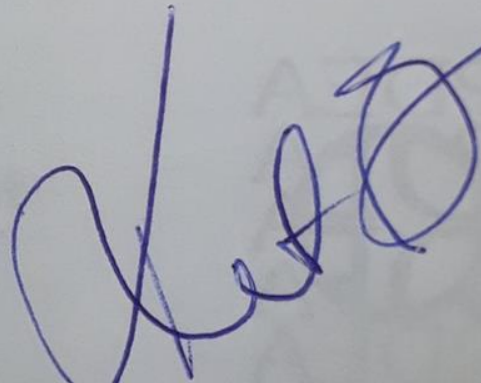
6/12/2002 (Sexta-feira)

09:00	<u>XP is Better in Smalltalk</u> Palestrante: James Robertson (auditório / 8º andar)	<u>Prevayler</u> Palestrante: Ramon Tramontini e Alexandre Nodari sala 601 (6º andar)	<u>Testes de unidade em Java com JUnit</u> Palestrante: Heider da Rocha sala 602 (6º andar)	<u>Extraindo o melhor do XP, Agile Modeling e RUP para melhor produzir software</u> Palestrante: Jefferson B Santos sala 603 (6º andar)	Openspace Technology sala 604 (6º andar)	XP Fest 9º andar
10:30	Coffee Break (6º andar)					
11:00	<u>The Customer's Side</u> Palestrante: Lowell Lindstrom (auditório / 8º andar)	<u>Mini Painel: Ensinando e Aprendendo XP</u> Palestrantes: Fábio Kon e Vinícius Teles sala 601 (6º andar)	<u>Orientação a Aspectos em Java para Metodologias Ágeis</u> Palestrantes: Jorge A. Ditz sala 602 (6º andar)	<u>Pontes seguras e cantos escuros ao implantar XP</u> Palestrante: Ronie Ullana sala 603 (6º andar)	Openspace Technology sala 604 (6º andar)	XP Fest 9º andar
12:30	Almoço (5º andar)					
13:30	<u>Dicas, Truques e Patterns Para o Sucesso da XP</u> Palestrante: Rob Mee (auditório / 8º andar)	<u>Integração de XP com PADS</u> Palestrante: Guilherme Lacerda sala 601 (6º andar)	<u>Técnicas de Refatoramento</u> Palestrante: Fábio Kon sala 602 (6º andar)	<u>Testes de integração em J2EE com Jakarta Cactus</u> Palestrante: Heider da Rocha sala 603 (6º andar)	Openspace Technology sala 604 (6º andar)	XP Fest
15:00	Coffee Break (6º andar)					
15:30	<u>Painel O Futuro de XP e Processos Ágeis</u> Kent Beck, Scott Ambler, Lowell Lindstrom, Rob Mee e Flavio Diomedea (auditório / 8º andar)					
17:30	Enceramento					

eXtreme programming

Guia Prático

Lacena
2002



5 DEC 02
STB PAULO



Pra entender quanto tempo faz, o computador tinha 512MB de RAM! :D



5º Fórum Internacional Software Livre



GUMA 10 Anos
(2014)

Manifesto para Desenvolvimento Ágil de Software

Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

Indivíduos e interações mais que processos e ferramentas
Software em funcionamento mais que documentação abrangente
Colaboração com o cliente mais que negociação de contratos
Responder a mudanças mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

Princípios

- **Satisfazer o cliente**, entregando o software em tempo hábil e continuamente
- **Aceitar as mudanças de requisitos**, em qualquer fase do projeto
- **Entregar software na menor escala** de tempo possível
- **Equipe de desenvolvimento e cliente** são do mesmo time
- Construir projetos com **indivíduos motivados e comprometidos** com o resultado

Princípios

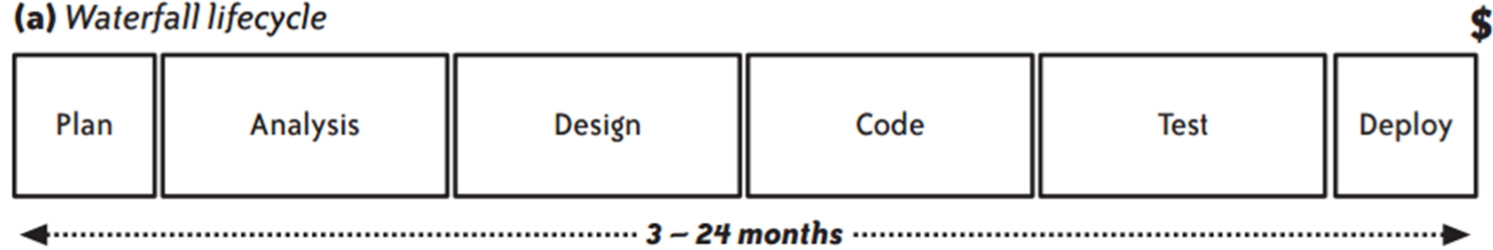
- Usar a **comunicação** efetiva
- Ter o **software em funcionamento** é a principal medida de **progresso**
- Atenção contínua à **excelência técnica**
- As melhores arquiteturas, requisitos e projetos provêm de **equipes organizadas**
- Refletir sobre como **se tornar mais eficaz**, ajustando e adaptando o comportamento da equipe

Princípios

Simplicidade – a arte de maximizar a quantidade de trabalho não realizado – é essencial.

O desenvolvimento de software

(a) Waterfall lifecycle



(b) Iterative lifecycle



\$ = Potential release

Falta de Comunicação



Falta de Comunicação



Resultado dos testes de "stress"



Como o software foi instalado pela equipe de operação



Como foi feita a Gerência de Riscos



Como o projeto foi documentado



Como o software foi suportado

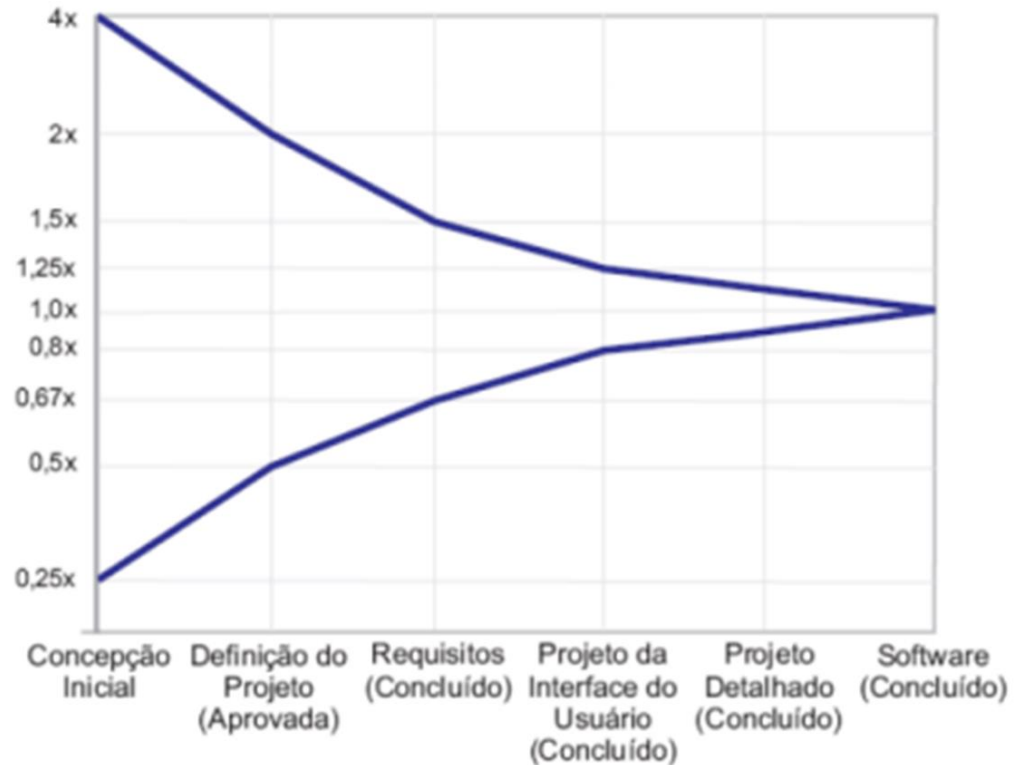


Quando o software foi entregue

Falta de Comunicação

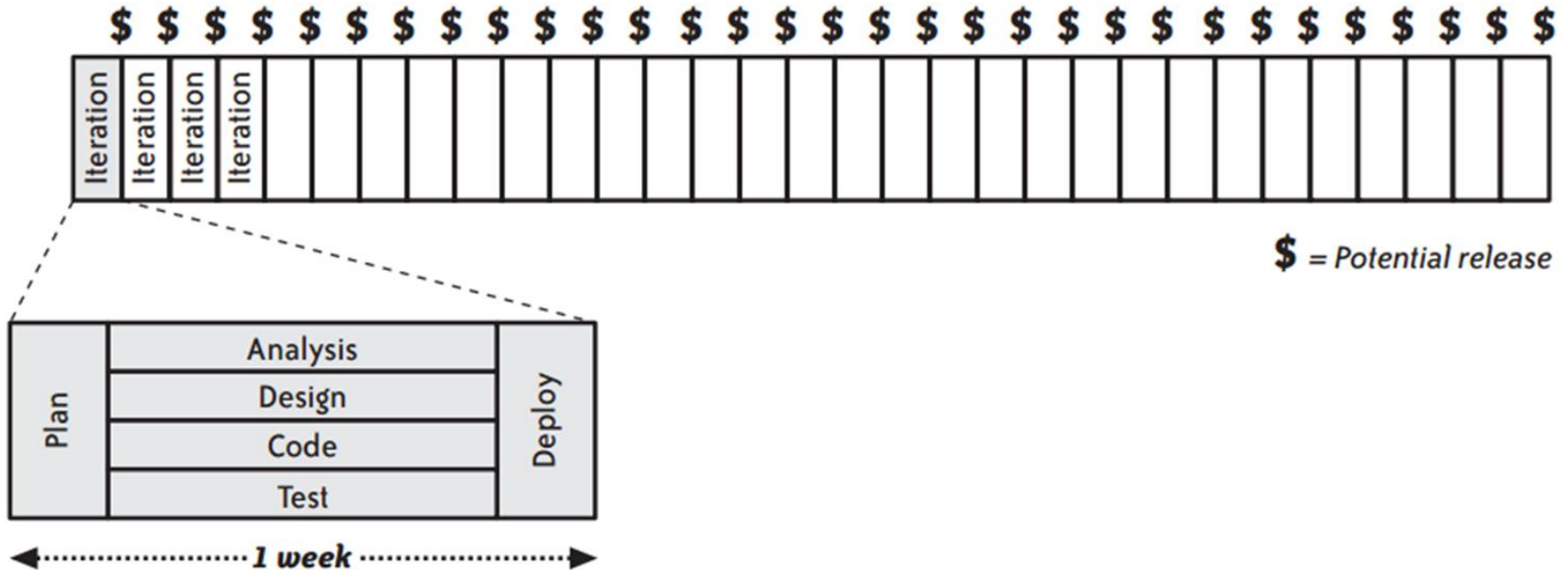


Cone da Incerteza



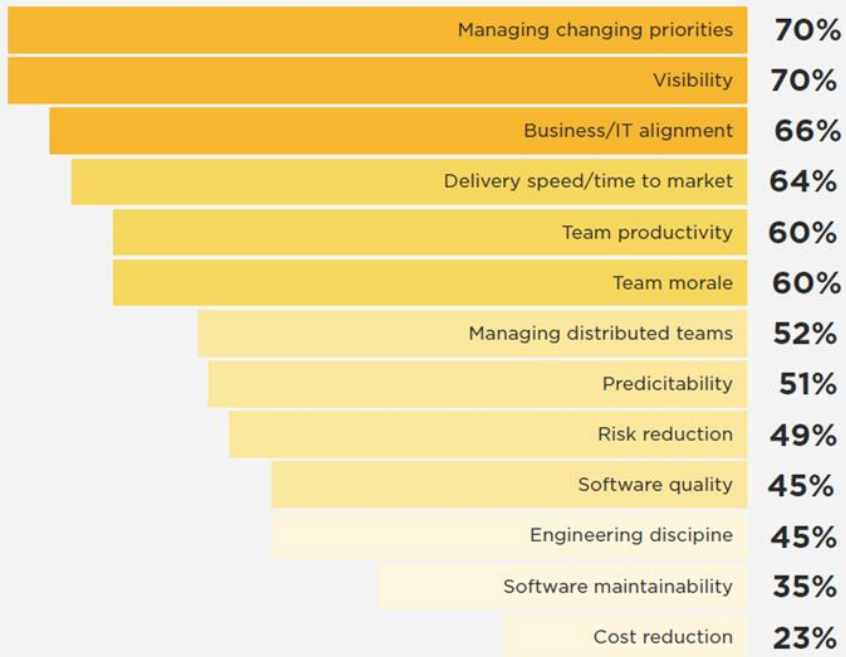
Barry Boehm, *Software Engineering Economics*, Prentice Hall (1981)

Desenvolvimento Ágil

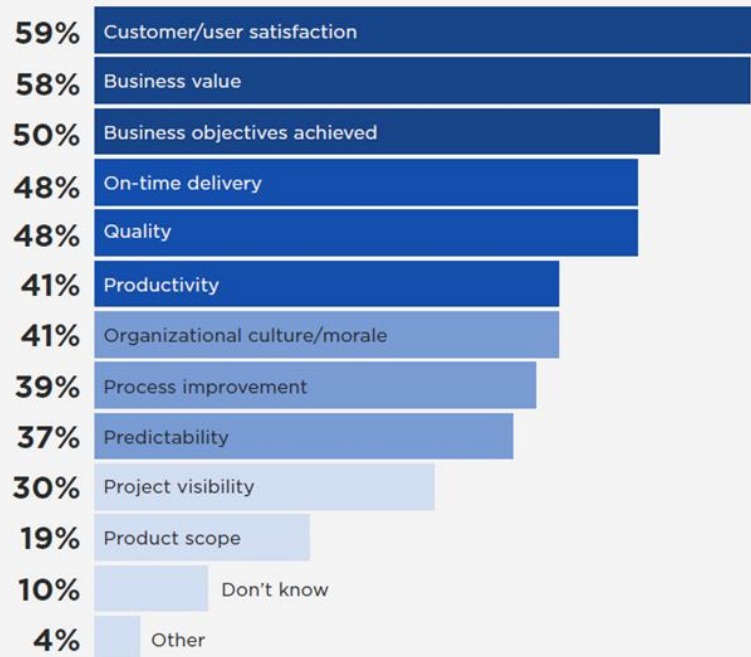


Adoção da Agilidade

Has the implementation of agile positively impacted each of the following areas within your company?

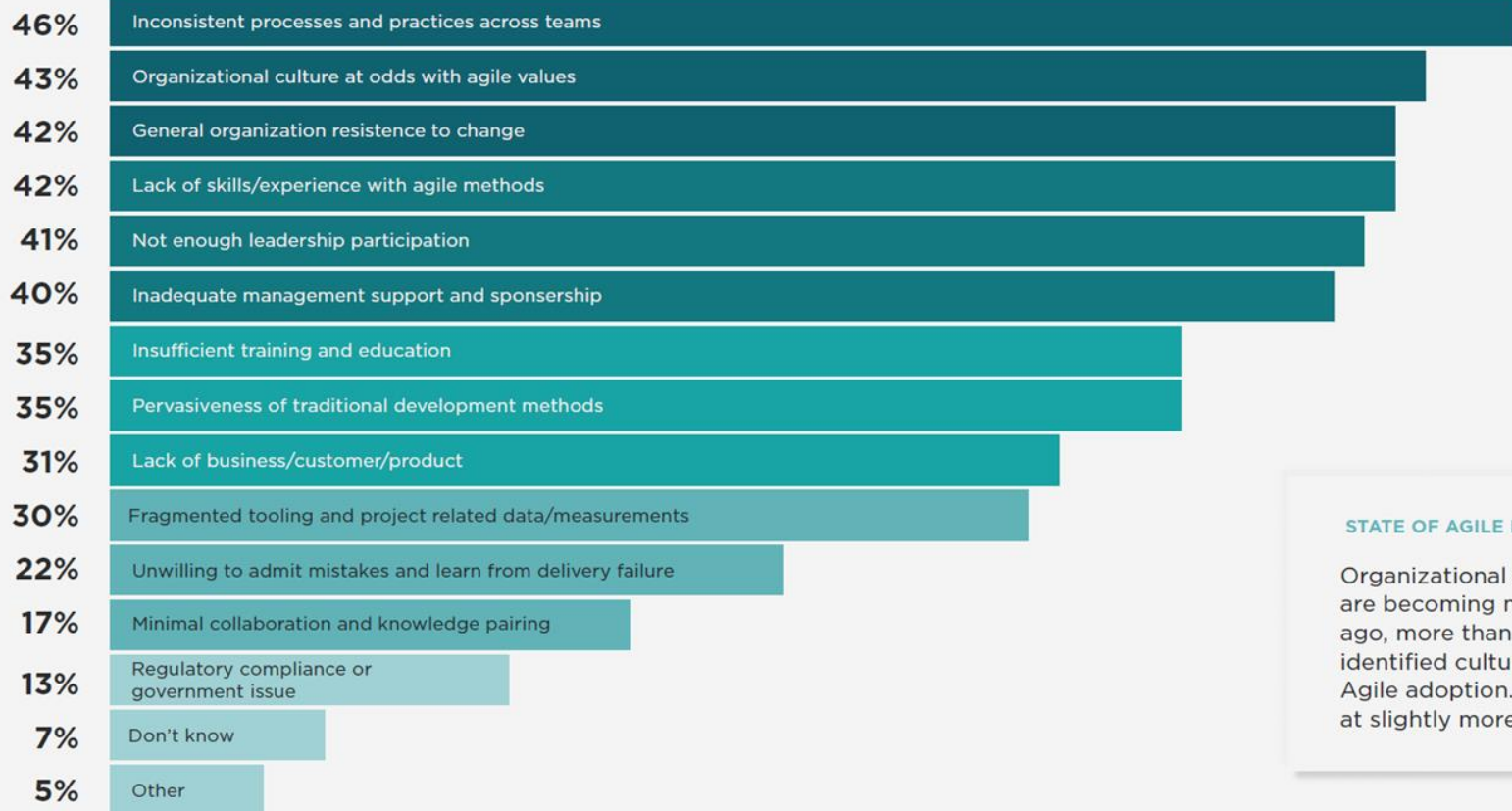


How does your organization measure success of Agile transformations?



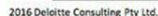
Desafios

15th State of Agile Report



STATE OF AGILE LOOK BACK

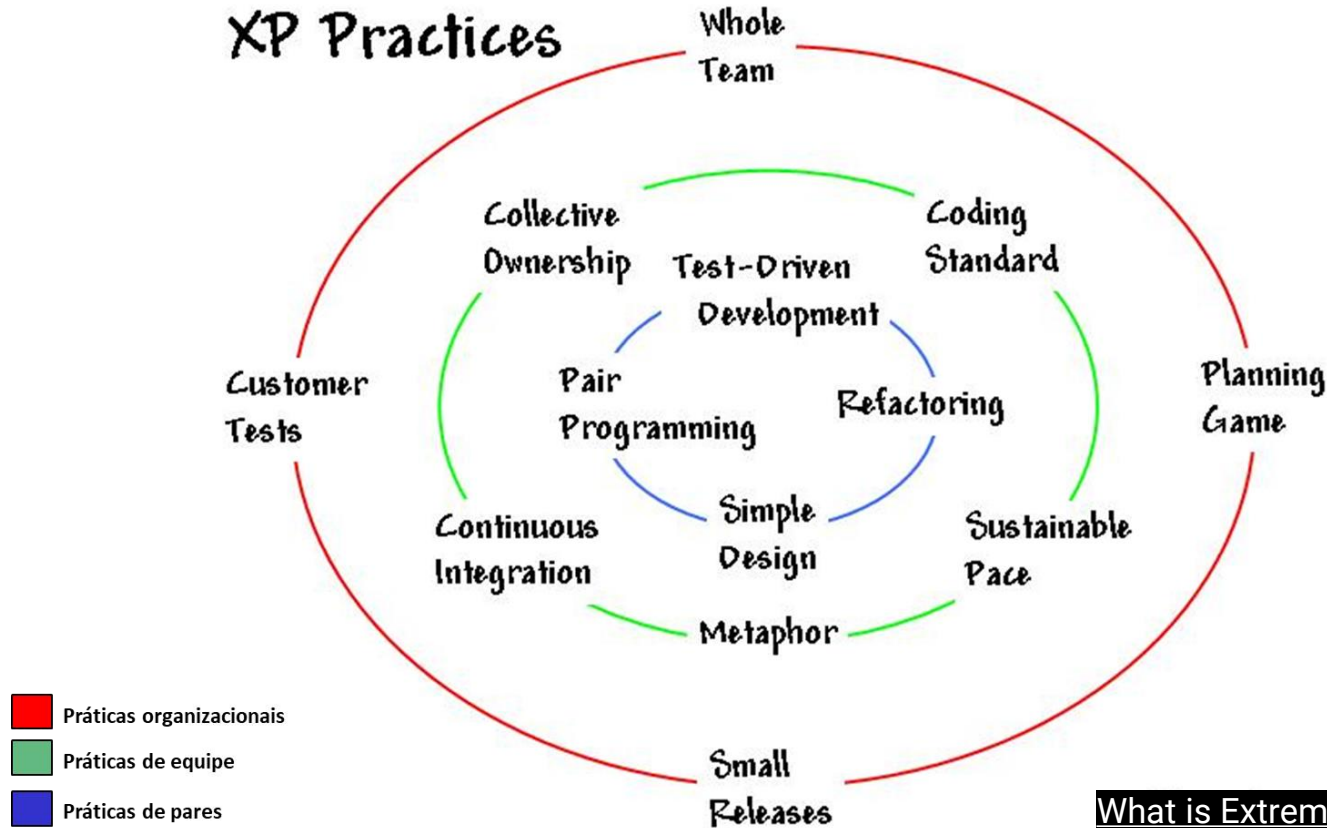
Organizational culture and Agile values are becoming more aligned. A few years ago, more than 8 out of 10 respondents identified culture as a significant barrier to Agile adoption. That number now stands at slightly more than 4 in 10.



Engenharia

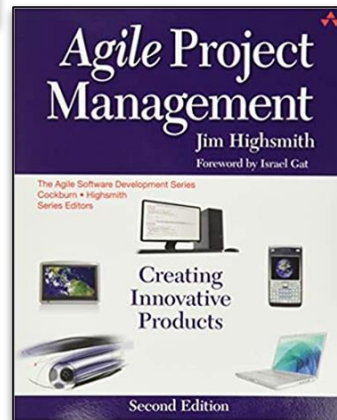
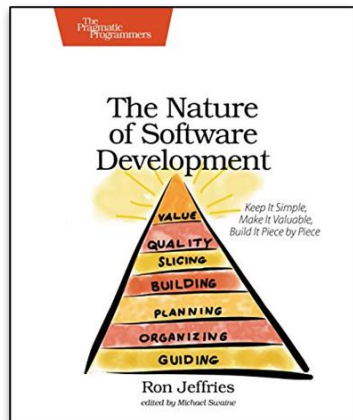
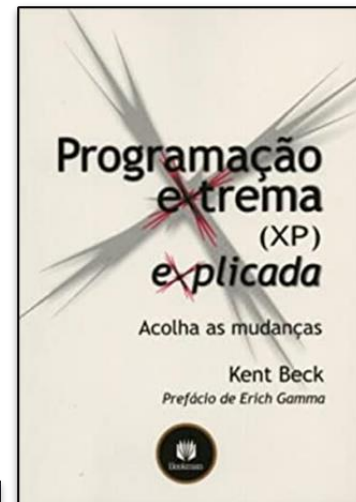
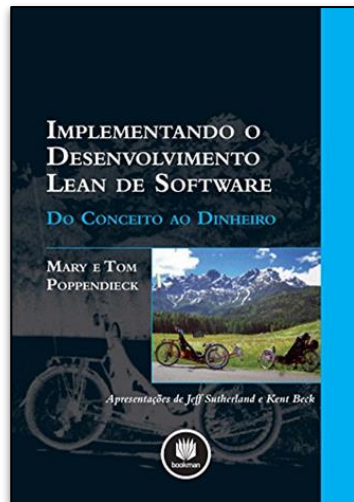
- Assim como a gestão, a **engenharia** precisou se renovar
- Conjunto de **práticas** e **ferramentas** para apoiar o **processo**
 - *Automações (testes, tarefas, processos)*
 - *Refatoração e heurísticas de limpeza*
 - *Revisões de código*
 - *Princípios de Design*
- Entender que **software** é um **produto**
 - *Diferente de outros produtos (virtual, ausente de propriedades físicas)*
 - *Manutenção e evolução*

Práticas do eXtreme Programming



What is Extreme Programming?

Para aprofundar os estudos...



Fim Parte 1

Parte 2

Agenda - Introdução

Parte 2

1. 3C
2. Spikes
3. Ritmo Sustentável
4. Formação de times

Conceito 3C

- Cartão: informação limitada, opera como um lembrete
- Conversação: quem nos ajuda a amplificar o conhecimento e desenvolver exemplos sobre o que estamos aprendendo?
- Confirmação: conforme conversamos com clientes conseguimos comprovar os atendimentos, através de validação com o mundo real.

A importância das entregas frequentes!

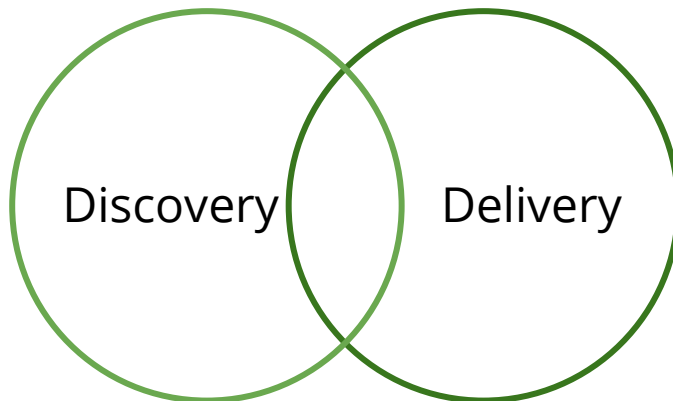
Spike Solution

- Aprender antes, para depois poder descobrir e habilitar a entrega
- Tempo de investigação
- Preparação para o fazer
- Remoção de risco técnico
- Modo descoberta e modo entrega



O que se faz em discovery e delivery?

- Entendimento do problema.
- Escolha de hipóteses de negócio a serem validadas.
- Complemento/Novas Ofertas?
- Marketing + Produto + Negócios trabalhando juntos
- [tech] Spikes a serem realizados para eliminar risco sobre entendimento de negócio?
- [tech] Entendimento de métricas de negócio e produtos para modelar problema a ser resolvido.
- [tech] Equipe técnica apóia no trabalho de descoberta. Visão consistente com realidade.



- Problema de negócio desdobrado em solução técnica.
- Entendimento das entregas de valor a serem realizadas e testadas (pivot / persist?)
- Clientes a serem usados como validação conhecidos.
- Métricas que serão analisadas após conclusão estão entendidas.
- Time técnico trabalha para entregar solução técnica.
- Entregas semanais, validação de produtos e clientes pré selecionados.

Ritmo Sustentável

- Como funciona a sua vida?
- Qual tempo existe para você?
- Quando você aprende?
- Quando você não faz nada?
- Ritmo sustentável trata de tempo de presença, tempo de foco.
- Horas extras? Eventualmente vai acontecer. Você vai assumir por entender suas responsabilidades.



Formação de Times

- Multidisciplina
- Carreira T, Pi, W?
- Como a equipe aprende?
- O que você faz quando não se sabe o que fazer?
- O que você faz quando não se tem o que fazer?

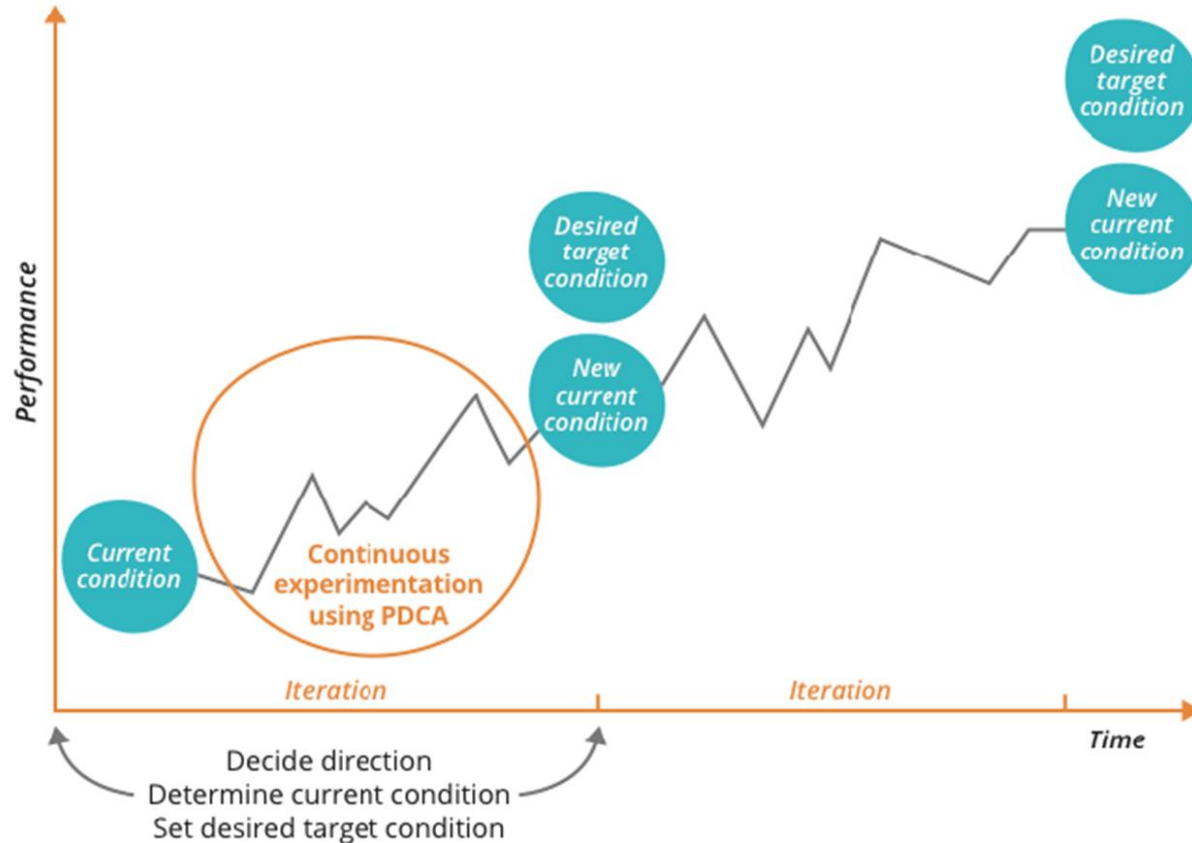


Lean Enterprise

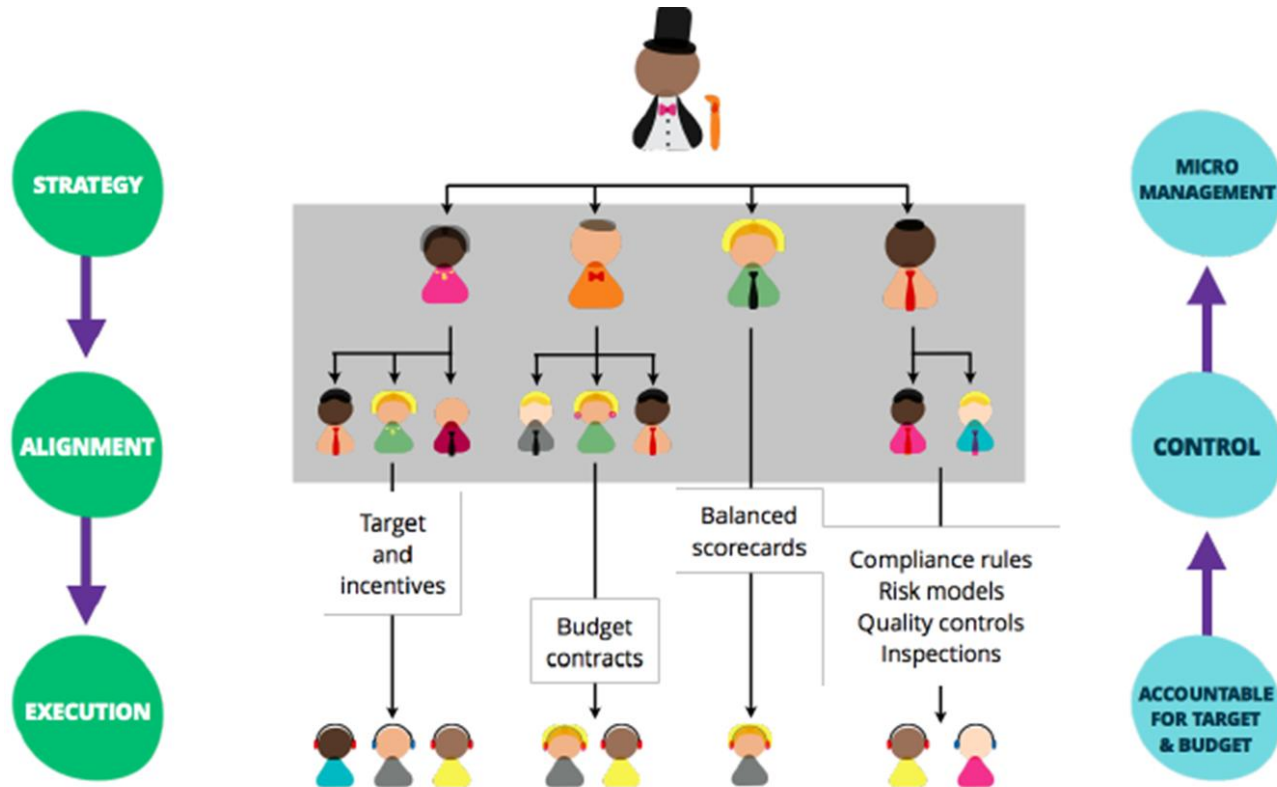
- Empresa
 - *Um sistema complexo e adaptativo composto de pessoas que compartilham um propósito comum*



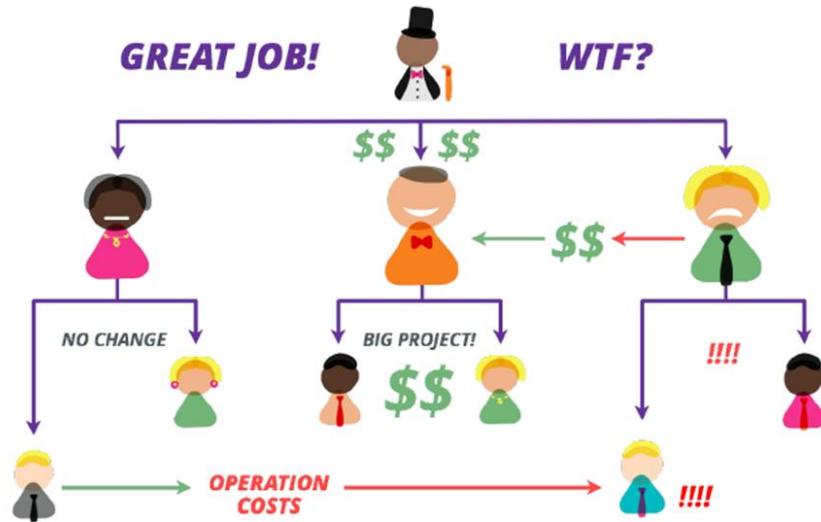
Lean Enterprise



Comando e Controle



Abordagem tradicional, de projetos



BUSINESS

Many 'good' ideas -
WE NEED \$\$\$!



PROJECTS

We're working -
GIVE US \$\$\$!

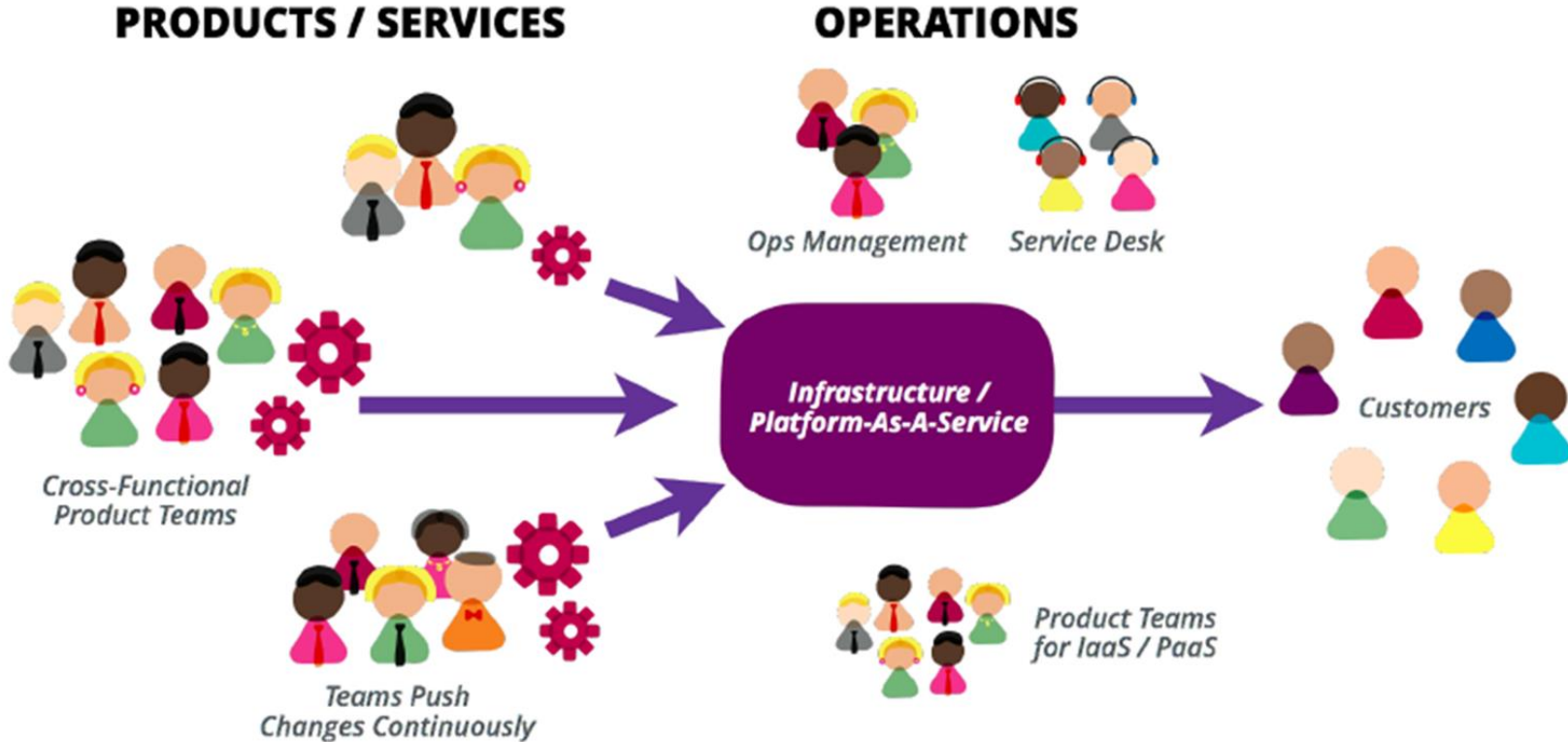


OPERATIONS

That doesn't work -
NEED MORE \$\$\$!



Produtos/Serviços



4 fundamental topologies

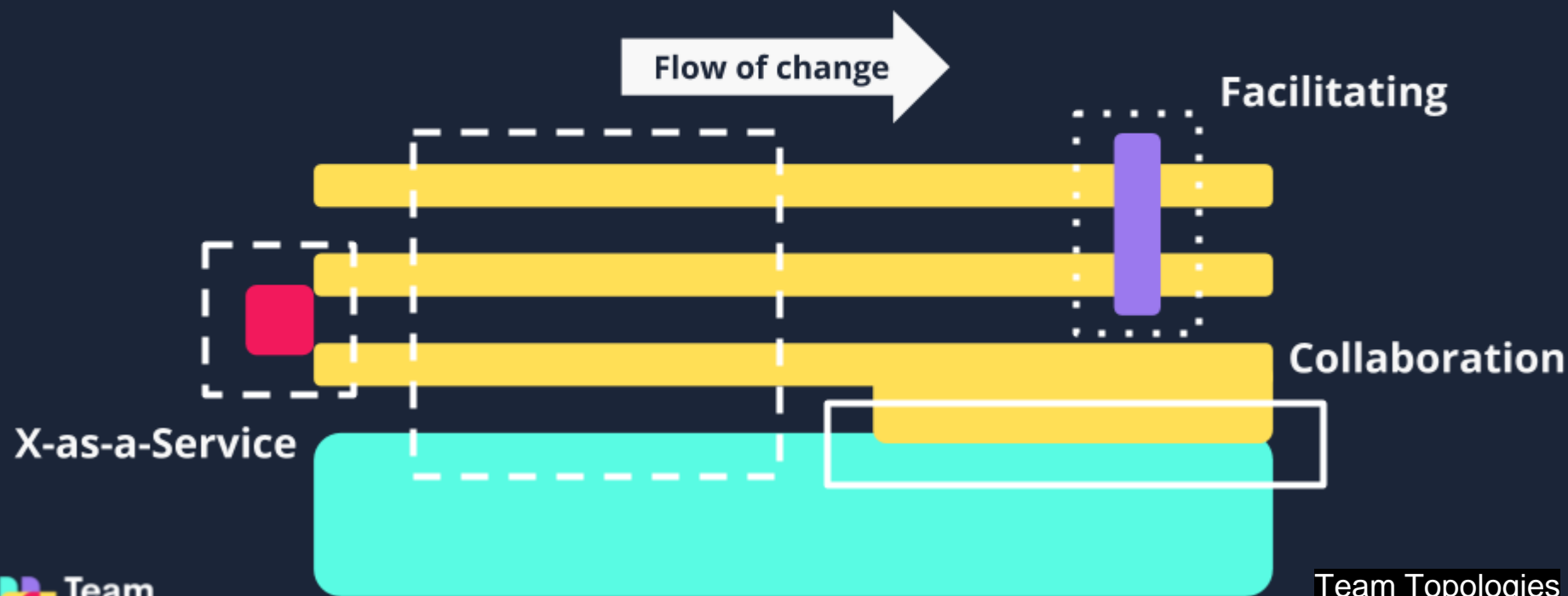
 Stream-aligned team

 Enabling team

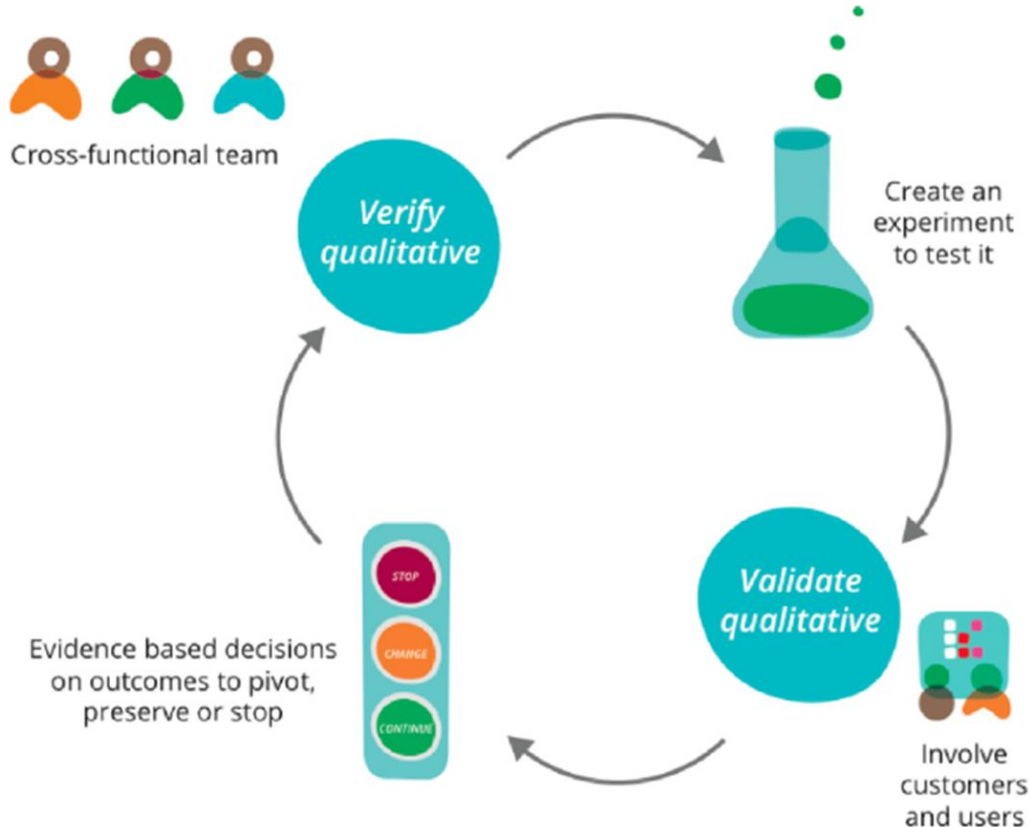
 Complicated Subsystem team

 Platform team

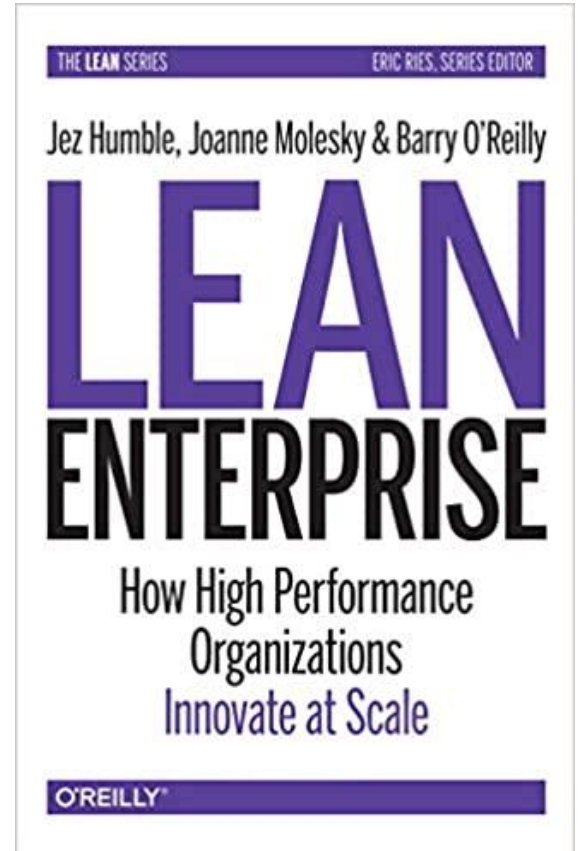
3 core interaction modes



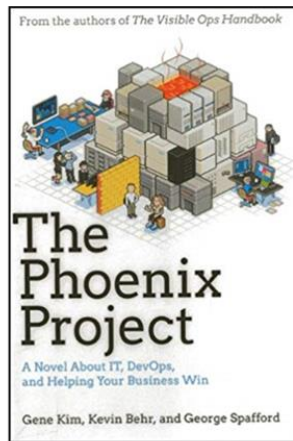
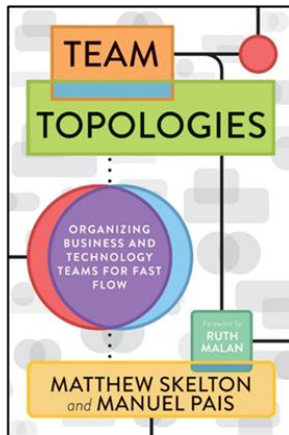
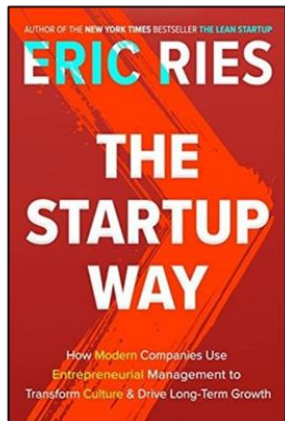
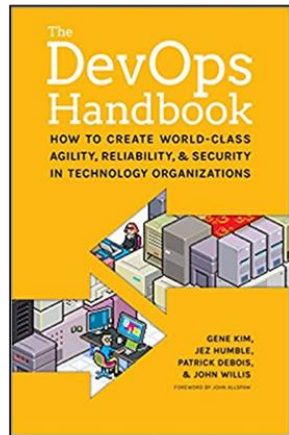
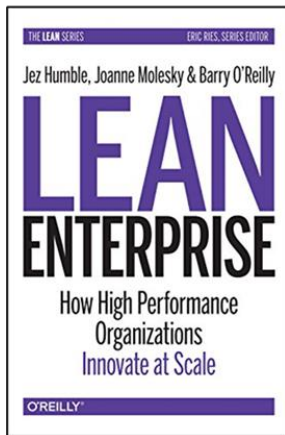
Ciclo de Aprendizado



Humble, Molesky, O'Reilly, Lean Enterprise: How High Performance Organizations Innovate At Scale



Para aprofundar os estudos...



Fim Parte 2

PUCRS online  **uol**edtech.