



# Tópicos Fundamentais

Prof.<sup>a</sup> Catarina Costa



# Agenda

---

**01** Oi ☺

**02** Metodologias Ágeis

**03** SCRUM



<https://github.com/catarinacosta>



## Catarina Costa

Associate Professor - UFAC

[About](#)

[Research](#)

[Students](#)

[Publications](#)

[Service](#)



## Short Bio

Catarina Costa is a Professor at the Exact and Technological Sciences Center at the [Federal University of Acre \(UFAC\)](#). She holds a Ph.D. (2017) degree in Computer Science from [Fluminense Federal University \(UFF\)](#), under the supervision of [Professor Leonardo Murta](#), a M.S. (2010) degree also in Computer Science from [Federal University of Pernambuco \(UFPE\)](#), and a B.S. (2007) degree in Informatics from UFAC. Her research area is Software Engineering, and her current research interests include Configuration Management, Software Evolution and Data Mining.

Welcome to my homepage! For more information, see [Lattes](#) and [Scholar](#).

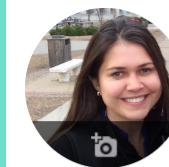
© 2020 - 2021 [Songzi Vong](#). All rights reserved. Powered by [Jekyll](#) & [Not Pure Poole](#).



[catarina.costa@ufac.br](mailto:catarina.costa@ufac.br)



<https://catarinacosta.github.io>



## Catarina de Souza Costa

UFAC

E-mail confirmado em ufac.br - [Página inicial](#)

Software Engineering Configuration Management Global Software

- | TÍTULO  | CITADO POR | AN   |
|---|------------|------|
| <a href="#">Challenges and solutions in distributed software development project management: a systematic literature review</a>         | 209        | 2010 |
| <a href="#">An evidence-based model of distributed software development project management: results from a systematic mapping study</a> | 44         | 2012 |
| <a href="#">TIPMerge: recommending experts for integrating changes across branches</a>  | 38         | 2016 |
| <a href="#">Version control in distributed software development: A systematic mapping study</a>   | 34         | 2018 |
| <a href="#">Collaboration models in distributed software development: A systematic review</a>   | 24         | 2020 |
| <a href="#">Characterizing the Problem of Developers' Assignment for Merging Branches</a>   |            |      |
- FQB da Silva, C Costa, ACC França, R Prikladnicki  
Global Software Engineering (ICGSE), 2010 5th IEEE International Conference ...
- FQB Da Silva, R Prikladnicki, ACC França, CVF Monteiro, C Costa, ...  
Journal of software: Evolution and Process 24 (6), 625-642
- C Costa, J Figueiredo, L Murta, A Sarma  
Proceedings of the 2016 24th ACM SIGSOFT International Symposium on ...
- C Costa, L Murta  
2013 IEEE 8th International Conference on Global Software Engineering, 90-99
- RGC Rocha, C Costa, C Rodrigues, R Ribeiro de Azevedo, IH Junior, ...  
CLEI Electronic Journal 14 (2), 1-1
- C Costa, JJC Figueiredo, G Ghiootto, L Murta  
International Journal of Software Engineering and Knowledge Engineering 24 ...

# Ementa

Métodologias  
Ágeis

Gerenciamento  
de  
Configuração



Artefatos / Conceitos principais  
Ferramentas  
Práticas ágeis  
SCRUM e Trello



Versões  
Mudanças  
GIT/GITHUB



# Agenda

---

**01** Oi 😊

**02** Metodologias Ágeis

**03** SCRUM

É sobre isso...

- A Engenharia de Software...
  - Abordagem disciplinada para o desenvolvimento de software
  - Grande diversidade de metodologias
- Ponto em comum nas metodologias:
  - Refinamentos sucessivos de artefatos

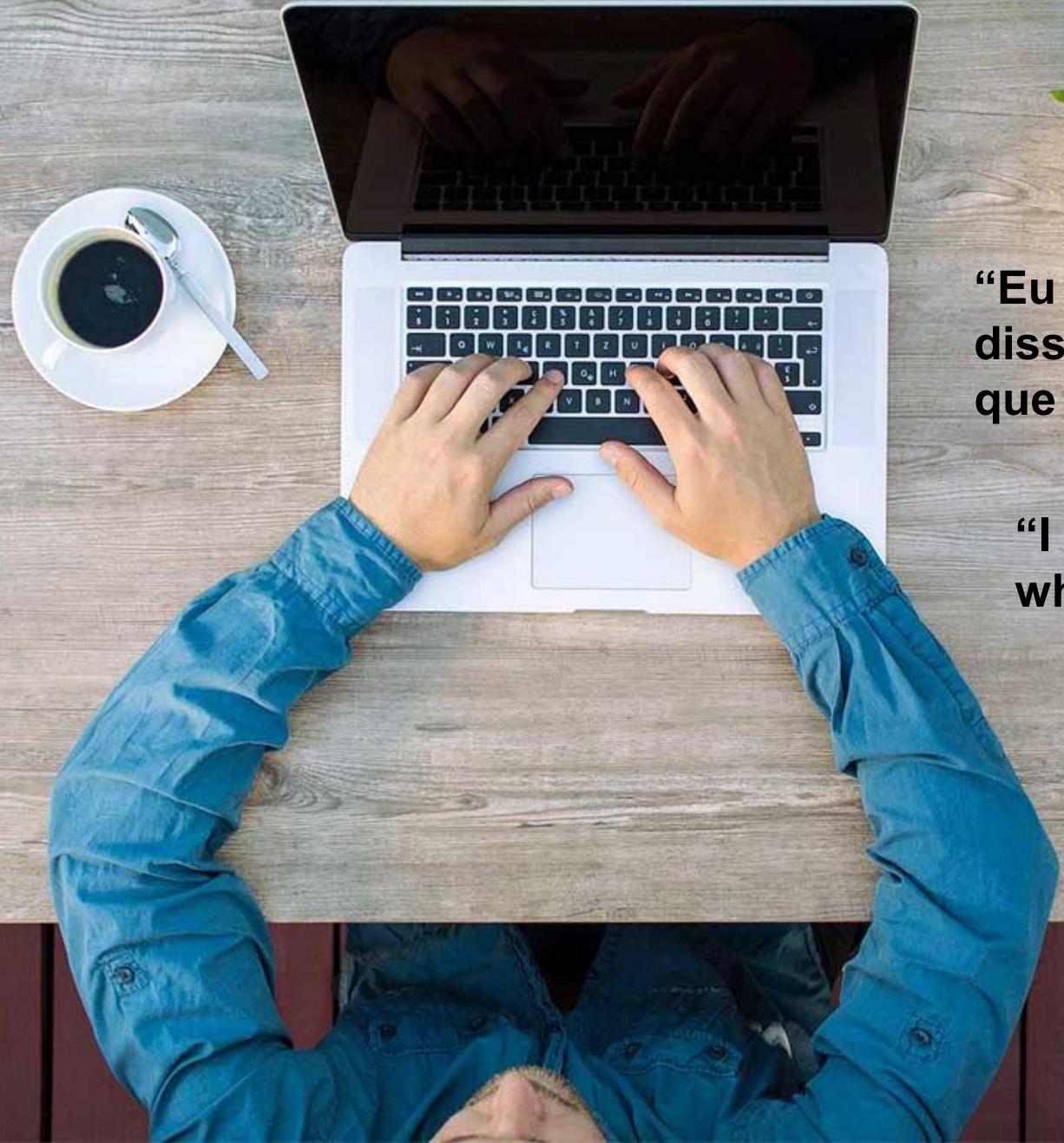




# Artefatos?

Artefatos são quaisquer documentos que puderem ser produzidos durante um projeto de desenvolvimento de software, incluindo:

- Diagramas
- Programas
- Documentos de texto
- Desenhos
- Contratos
- Projetos
- Planos
- Código fonte
- ...



## DESAFIOS:

**“Eu sei que você pensa que entendeu o que eu disse, mas o que você não entende é que aquilo que eu disse não era o que eu queria dizer”**

**“I can't tell you what I want, but I'll know it when I see it ” Barry Boehm**

**“The hardest single part of building a software system is deciding precisely what to build” Frederick P. Brooks**

# Engenharia de Software?

“Engenharia de Software é a aplicação de uma abordagem **sistemática, disciplinada e quantificável** ao desenvolvimento, operação e manutenção de software”

IEEE Std 610.12 , 1990

A Engenharia de Software é uma disciplina de engenharia relacionada com todos os aspectos da produção de um software, da especificação até a sua manutenção.

Sommerville, 2007

A Engenharia de Software abrange um processo, um conjunto de métodos (práticas) e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de altíssima qualidade.

Pressman, 2016

Engenharia de Software é o processo de estudar, criar e otimizar os processos de trabalho para os desenvolvedores de software.

Wazlawick, 2013

# Engenharia de Software?

“Engenharia de Software é a aplicação de uma abordagem **sistemática, disciplinada e quantificável** ao desenvolvimento, operação e manutenção de software”

IEEE Std 610.12 , 1990

A Engenharia de Software é uma disciplina de engenharia relacionada com todos os aspectos da produção de um software, da especificação até a sua manutenção.

Sommerville, 2007

A Engenharia de Software abrange um processo, um conjunto de métodos (práticas) e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de altíssima qualidade.

Pressman, 2016

Engenharia de Software é o processo de estudar, criar e otimizar os processos de trabalho para os desenvolvedores de software.

Wazlawick, 2013

# Engenharia de Software?

“Engenharia de Software é a aplicação de uma abordagem **sistemática, disciplinada e quantificável** ao desenvolvimento, operação e manutenção de software”

IEEE Std 610.12 , 1990

A Engenharia de Software é uma disciplina de engenharia relacionada com todos os aspectos da produção de um software, da especificação até a sua manutenção.

Sommerville, 2007

A Engenharia de Software abrange um processo, um conjunto de métodos (práticas) e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de altíssima qualidade.

Pressman, 2016

Engenharia de Software é o processo de estudar, criar e otimizar os processos de trabalho para os desenvolvedores de software.

Wazlawick, 2013

# Engenharia de Software?

“Engenharia de Software é a aplicação de uma abordagem **sistemática, disciplinada e quantificável** ao desenvolvimento, operação e manutenção de software”

IEEE Std 610.12 , 1990

A Engenharia de Software é uma disciplina de engenharia relacionada com todos os aspectos da produção de um software, da especificação até a sua manutenção.

Sommerville, 2007

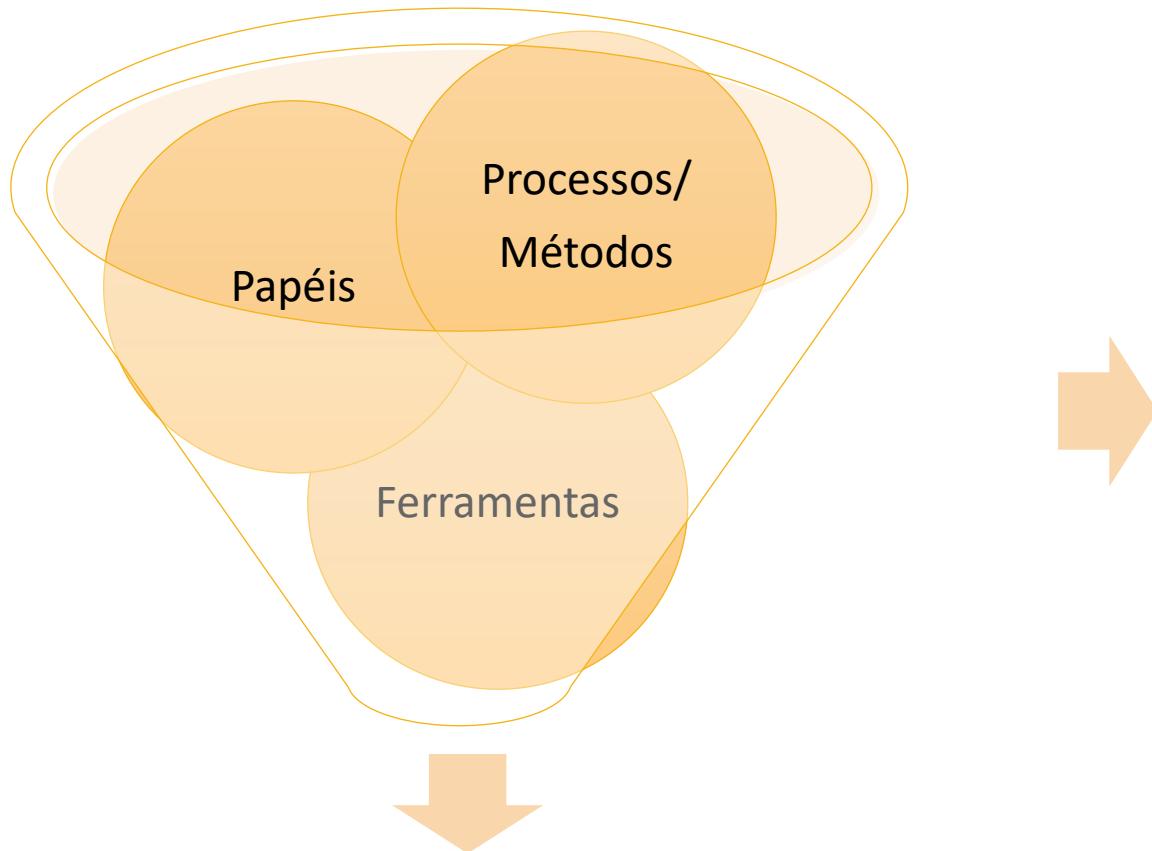
A Engenharia de Software abrange um processo, um conjunto de métodos (práticas) e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de altíssima qualidade.

Pressman, 2016

Engenharia de Software é o processo de estudar, criar e otimizar os processos de trabalho para os desenvolvedores de software.

Wazlawick, 2013

# Elementos da ES



Engenharia de Software

1. Coloque em uma **panela** funda o leite condensado, a margarina e o chocolate em pó.
2. Cozinhe [no **fogão**] em fogo médio e mexa sem parar com uma **colher de pau**.
3. Cozinhe até que o brigadeiro comece a desgrudar da **panela**.
4. Deixe esfriar bem, então **unte** as mãos com margarina, **faça as bolinhas** e **envolva**-as em chocolate granulado.



método



ferramenta

Processo

# Histórico (era pós-ES)

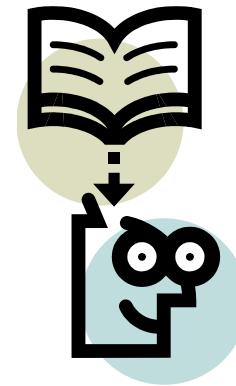
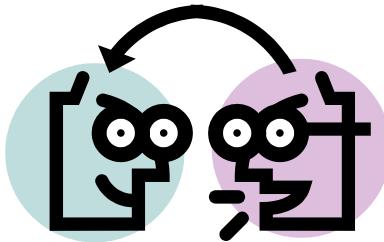
- 1970s:
  - Lower-CASE tools (programação, depuração, colaboração)
  - Ciclo de vida cascata
  - Desenvolvimento estruturado
- 1980s:
  - Ciclo de vida espiral
  - Desenvolvimento orientado a objetos
  - Controle de versões
  - Testes
- 1990s: Upper-CASE tools
  - Processos
  - Modelagem

# Histórico (era pós-ES)

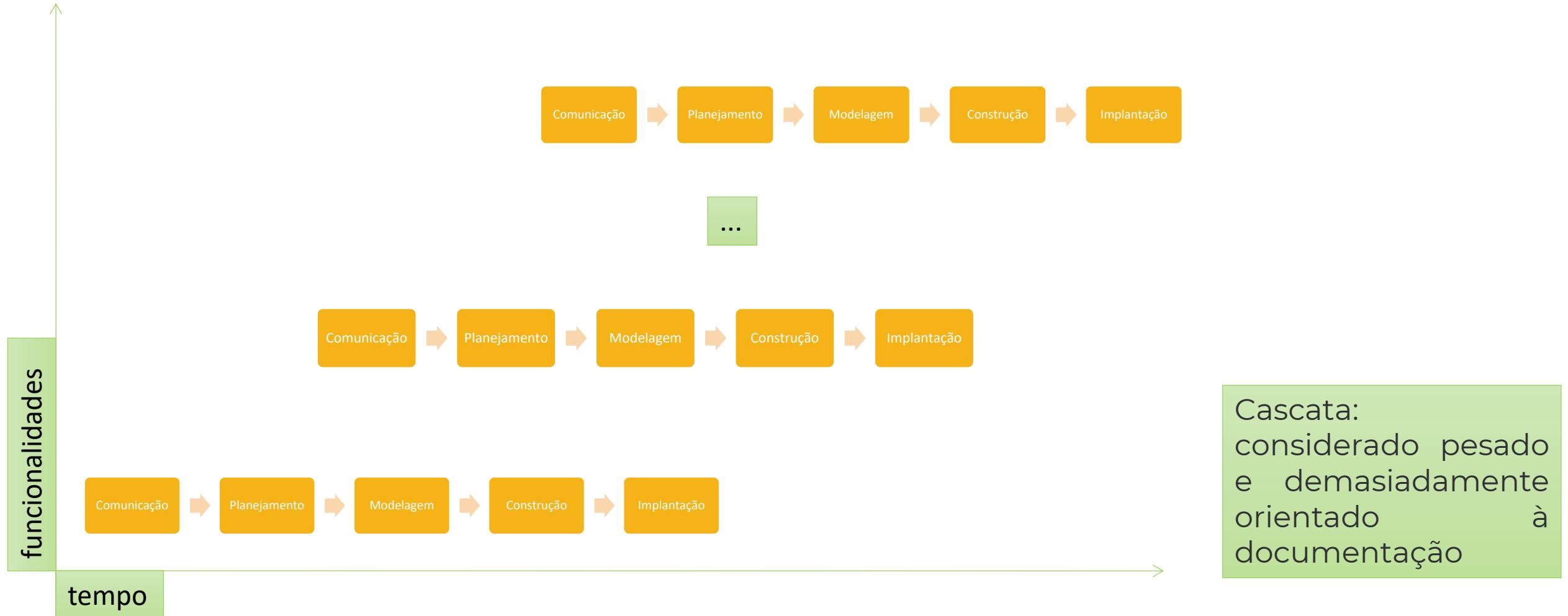
- 2000s:
  - Métodos ágeis
  - Desenvolvimento dirigido por modelos
  - Linhas de produto
  - Experimentação
- Atualmente
  - DevOps
  - Continuous
  - IA
  - ...

# Processos implícitos x explícitos

- Lembrem-se: Processos sempre existem, seja de forma implícita ou explícita!
  - Processos implícitos são difíceis de serem seguidos, em especial por novatos
  - Processos explícitos estabelecem as regras de forma clara

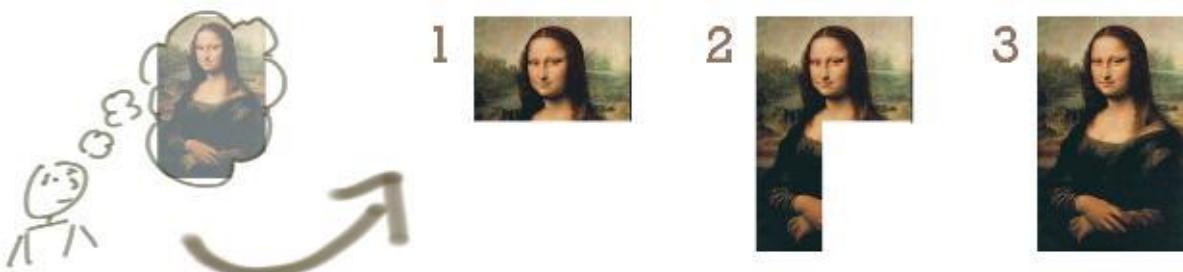


# Ciclo de vida Cascata x Incremental (Evolutivo)



# Ciclo de vida Incremental

- Faz entregas incrementais do software
  - Cada incremento é construído via um mini-cascata
  - Cada incremento é um software operacional
- Versões anteriores ajudam a refinar o plano
  - Feedback constante do cliente
- Diminuição da ansiedade do cliente
  - O cliente rapidamente recebe uma versão funcional do software



# Desenvolvimento Iterativo



A iteração deve ser fixa

Tarefas podem ser removidas ou incluídas

A iteração nunca deve passar da duração previamente estipulada



O resultado de cada iteração é um software...

Incompleto

Em desenvolvimento (não pode ser colocado em produção)

Mas não é um protótipo!!!

# Desenvolvimento Iterativo

- Esse software pode ser verificado e validado parcialmente
  - Testes
  - Usuários
- Podem ser necessárias diversas iterações (e.g. 10 a 15) para ter uma versão do sistema pronta para entrar em produção

# Agile -the Influencers



Kent Beck



Mike Beedle



Arie Bennekum



Alistair Cockburn



Ward Cunningham



Martin Fowler



James Grenning



Jim Highsmith



Andrew Hunt



Ron Jeffries



Jon Kern



Brian Marick



Robert C. Martin



Steve Mellor



Ken Schwaber



Jeff Sutherland

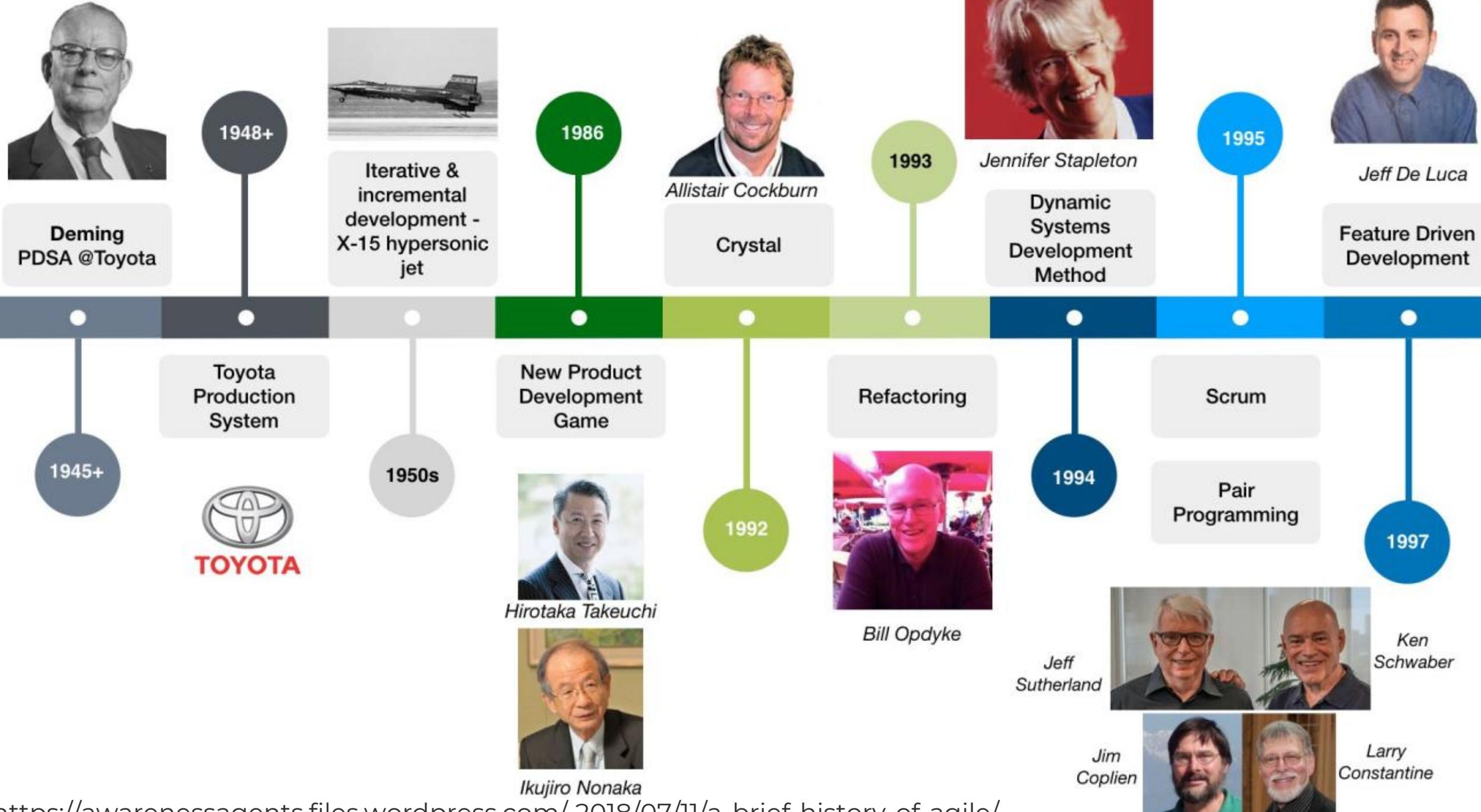


Dave Thomas



<http://www.manifestoagil.com.br>

Assinado por 17  
desenvolvedores com  
reconhecimento  
mundial em Utah em  
fevereiro/2001





Kent Beck

Extreme Programming

Pragmatic Programmer

2001



Kent Beck

Test-Driven Development

Craig Larman



2003



Bas Vodde

Large Scale Scrum



Patrick Debois



Andrew "Clay" Shafer



Scott Ambler

2012

Disciplined Agile Delivery

## THE AGILE MANIFESTO

1999

Adaptive Development



Jim Highsmith

2002



Mary & Tom Poppendieck

2005



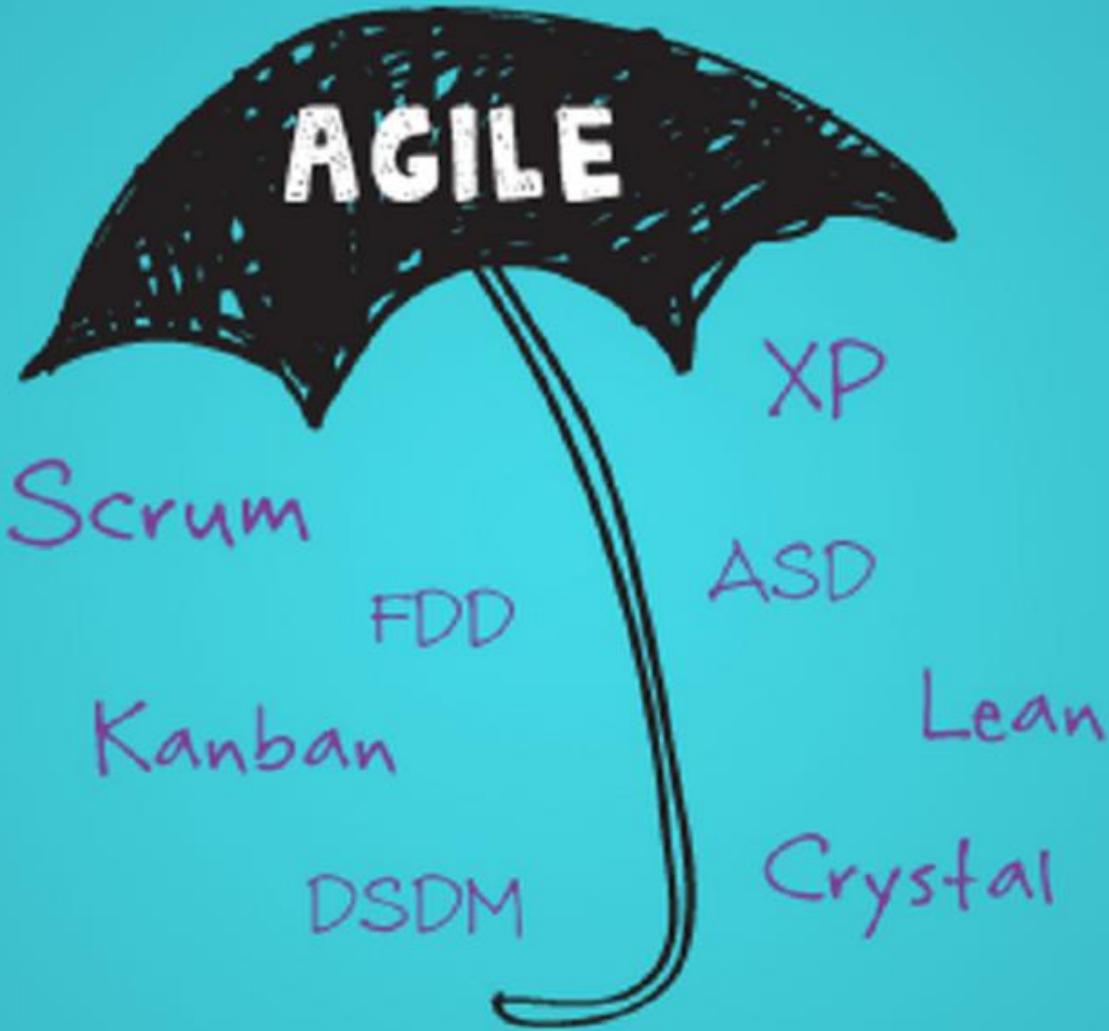
David J. Anderson

2009



Woody Zuill

2012



# Valores ágeis



# Os 12 princípios ágeis



1. Priorize o Cliente



2. Mudanças são bem-vindas



3. Entregas que geram valor



4. Cooperação entre squads e stakeholders



5. Confie e apoie



6. Conversa face a face



7. Priorize o que funciona



8. Desenvolvimento incremental



9. Excelência técnica



10. Mantenha a simplicidade



11. Times autônomos e auto-gerenciáveis



12. Retrospectiva e planejamento



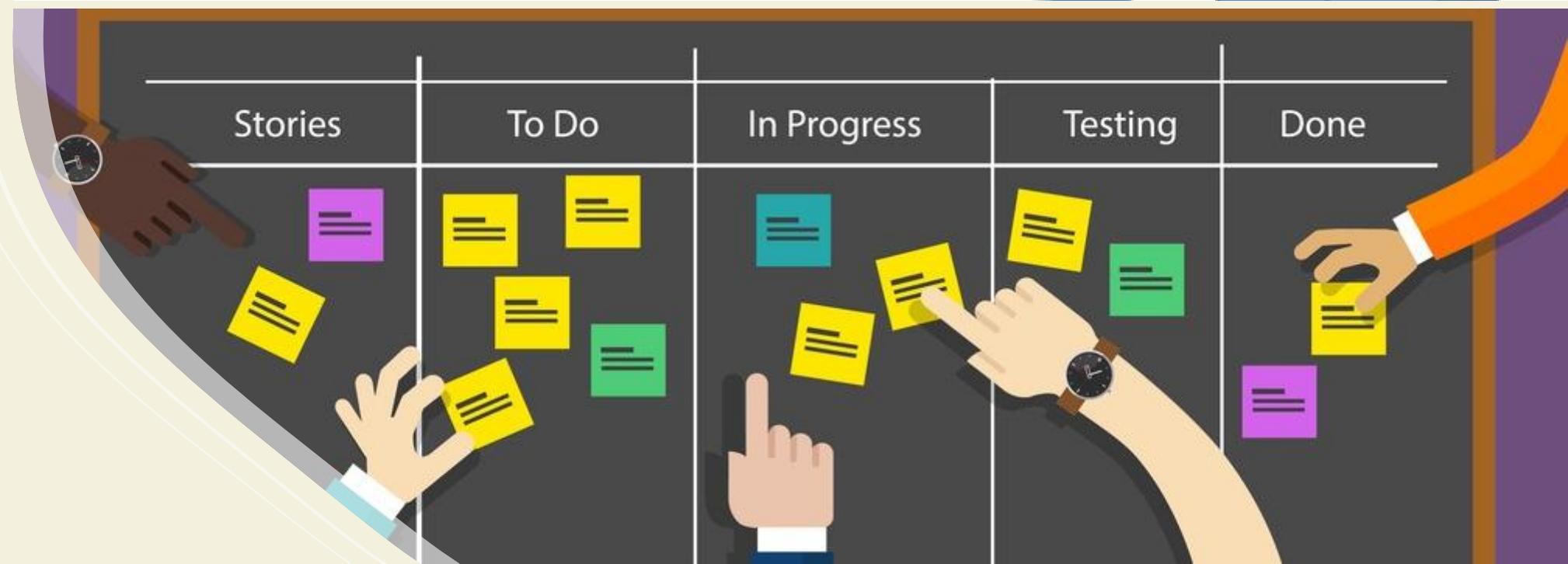
# Agenda

---

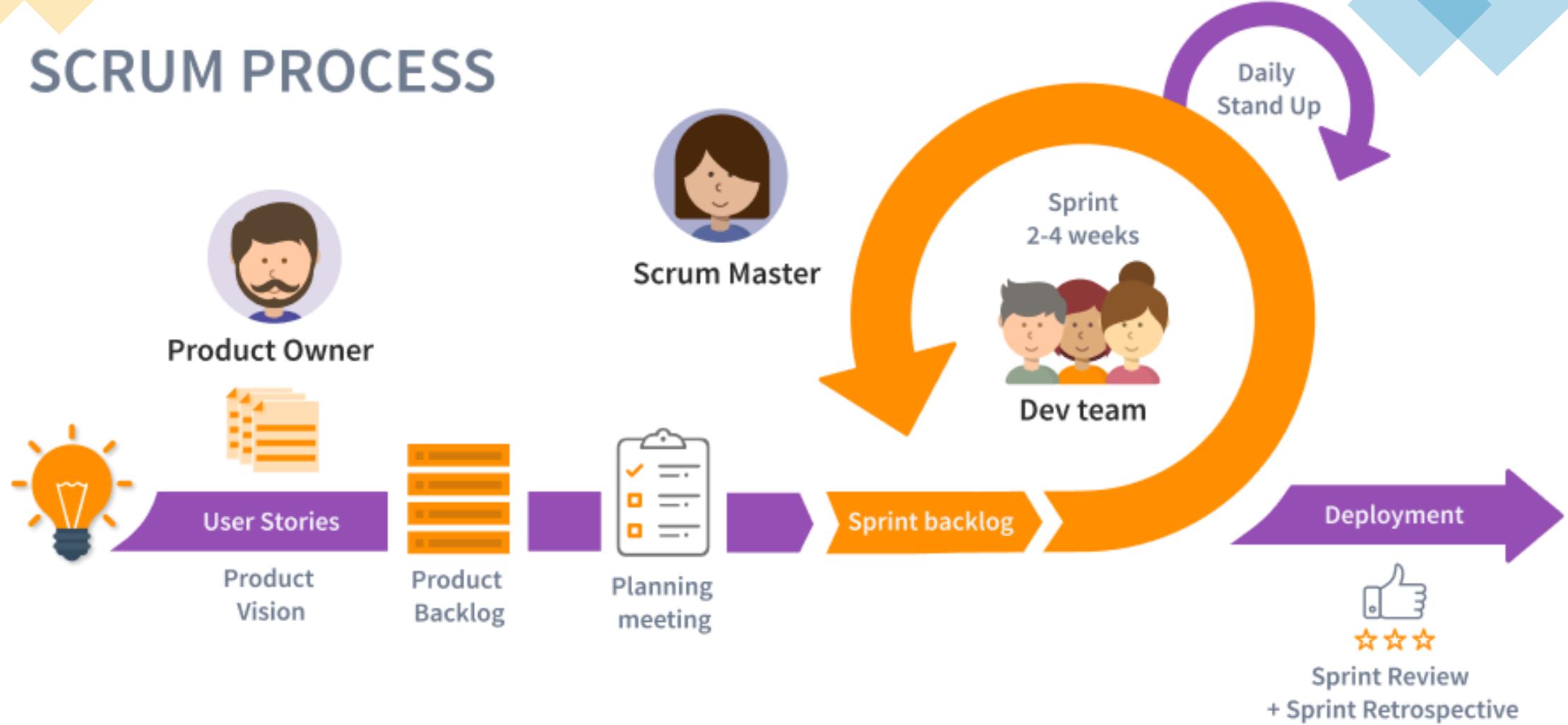
**01** Oi 😊

**02** Metodologias Ágeis

**03** SCRUM



# SCRUM PROCESS



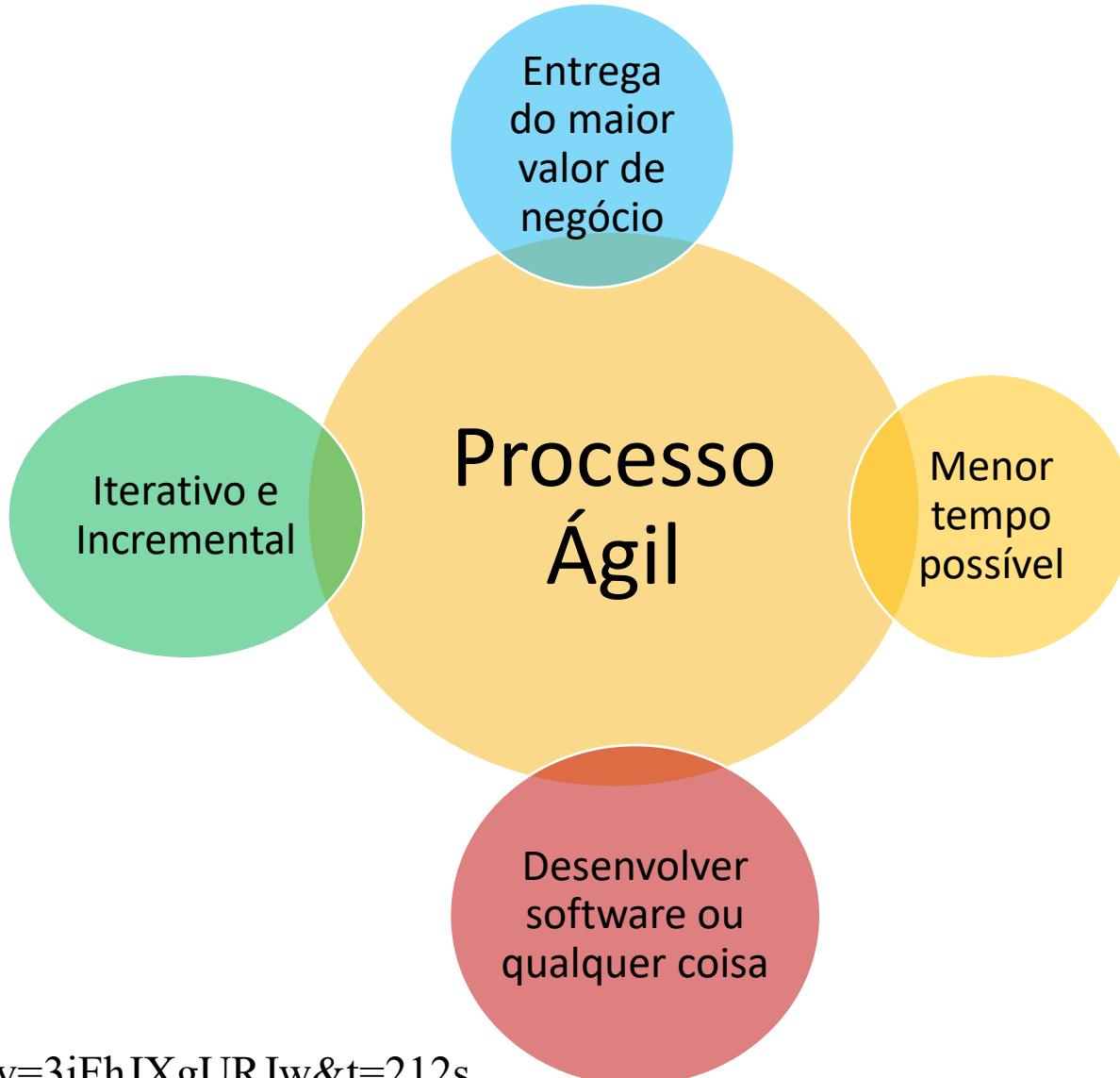


Estratégia em um  
jogo de rugby

Trabalho em  
equipe

Se um falhar,  
todos falham!

# O que é o Scrum?



<https://www.youtube.com/watch?v=3jFhJXgURJw&t=212s>

O Scrum não é um processo previsível, ele não define o que fazer em toda circunstância.

KEN SCHWABER 2004.

- Bastante objetivo
- Papéis e responsabilidades bem definidos
- Fácil adaptação
- É um *framework* e um conjunto de práticas



Read the official Scrum Guide online

[Read Online ➤](#)



Download the official Scrum Guide  
PDF in English

[Download \(English\) ➤](#)



Download the official Scrum Guide in  
over 30 different languages

[Select language & Download ➤](#)

## What is Scrum?

Scrum is a framework for developing and sustaining complex products. This Guide contains the definition of Scrum. This definition consists of Scrum's roles, events, artifacts, and the rules that bind them together. Ken Schwaber and Jeff Sutherland developed Scrum; the Scrum Guide is written and provided by them. Together, they stand behind the Scrum Guide.

## Share Your Ideas!

If you have ideas for improving the website or the Scrum Guide itself, please share them! You may do so by sending a message to our support e-mail.

[Share an idea](#)

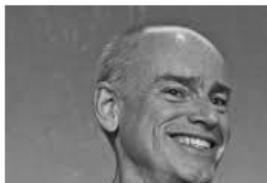
## About the creators of Scrum



### Meet Jeff Sutherland

Jeff is the co-creator of Scrum and a leading expert on how the framework has evolved to meet the needs of today's business...

[Read Jeff's Bio](#)



### Meet Ken Schwaber

Ken Schwaber co-developed the Scrum process with Jeff Sutherland in the early 1990s to help organizations...

[Read Ken's Bio](#)

# O que é o Scrum?

## Artefatos:

- **Product Backlog:** O Product Backlog é uma lista de todas as funcionalidades que devem ser implementadas no produto. Ele é priorizado pelo Product Owner, que é o responsável por representar os interesses dos usuários.
- **Sprint Backlog:** O Sprint Backlog é uma lista das funcionalidades que serão implementadas na próxima sprint.
- **Increment:** O Increment é o resultado de cada sprint. Ele é uma versão do produto que pode ser usada pelos usuários.

## • Papéis:

- Scrum Master;
- Product Owner;
- Desenvolvedores.

# Como o SCRUM trabalha?



Foco em gerenciamento

Pode ser combinado com outras metodologias (Ex.: XP)



Equipes pequenas (06 à 10 membros)



Uma série de *Sprints* (01 – 04 semanas)



Tempo limitado

Rápido desenvolvimento do produto

*Ei Porco, acho que deveríamos abrir um restaurante.*

*Não sei não. Qual seria o nome?*

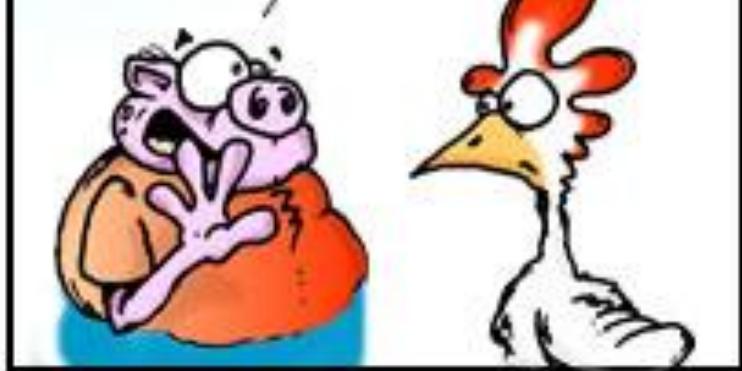


By Clark & Vizzdos

*Que tal "Pernil com Ovos".*



*Não obrigado, eu estaria comprometido, enquanto você estaria apenas envolvido!*



© 2006 implementingscrum.com

# Papéis e Responsabilidades do Scrum



Product Owner



Scrum Master



Time Scrum

# Scrum Master

- Coach e facilitador do time;
- Garante que a equipe está completamente funcional e produtiva;
- Facilita comunicação entre papéis e remove impedimentos;
- Protege a equipe contra interferências externas e excesso de otimismo;
- Assegura que o processo é seguido;
- Coordena os encontros diários, revisão e planejamento da iteração;
- Não é responsável por designar tarefas ao desenvolvedores, isso é papel do próprio time;
- Mantém o backlog do sprint (tarefas completadas, identifica eventuais problemas)
- Organiza as reuniões de 1º e 2º planejamento de sprint, diárias, revisão e retrospectiva.



# Product Owner



- Representa os *stakeholders* (tem poder de tomar (para o time) decisões pelo Cliente e Usuários);
- Define os requisitos (Product Backlog);
- Apresenta e explica o product backlog para o time;
- Define as prioridades de acordo com o valor de mercado;
- Participa do Sprint Planning e Sprint Review;
- Aceita ou rejeita os resultados do trabalho;
- Elimina a confusão de múltiplos chefes, opiniões diferentes e interferência.

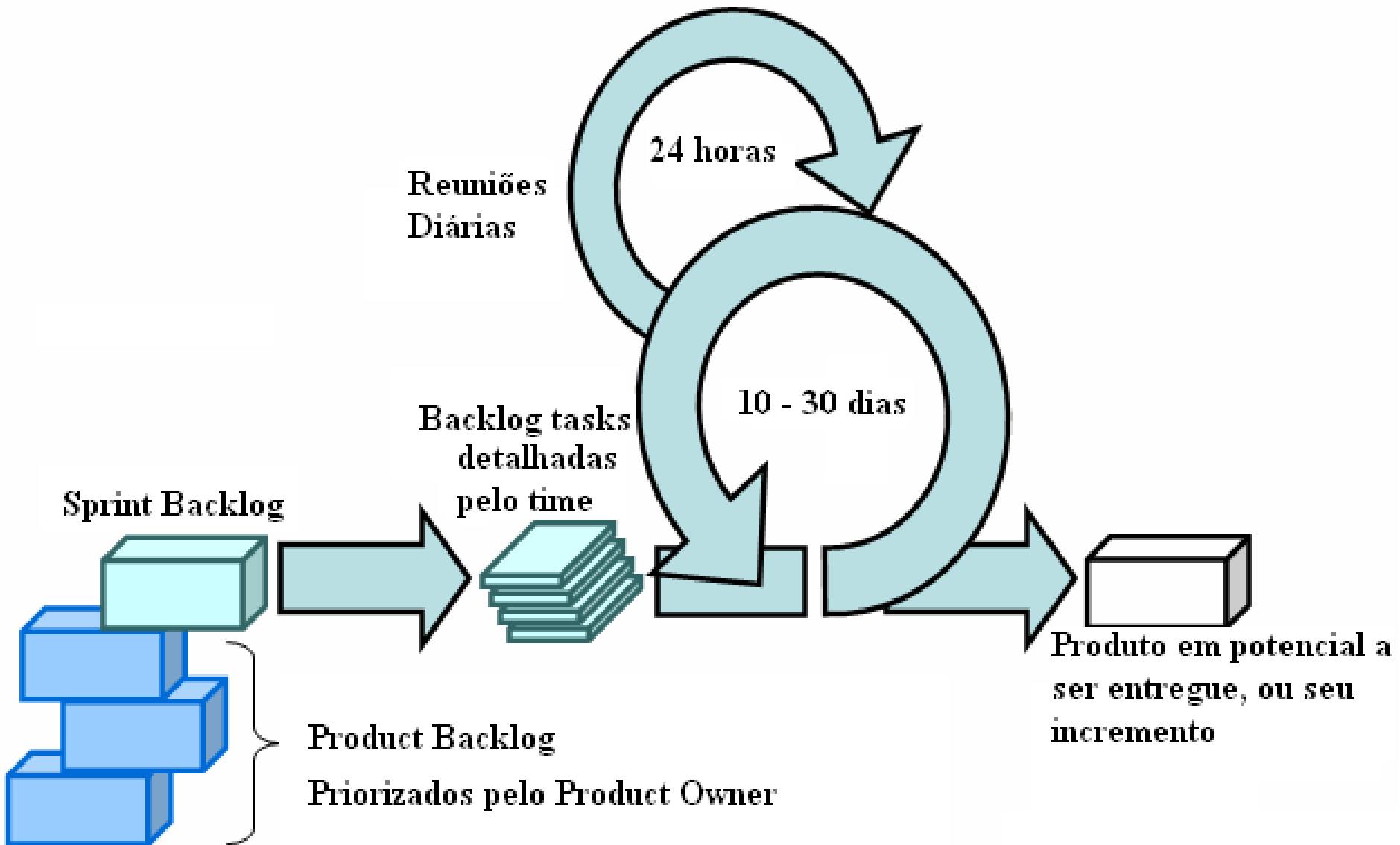


# Developers

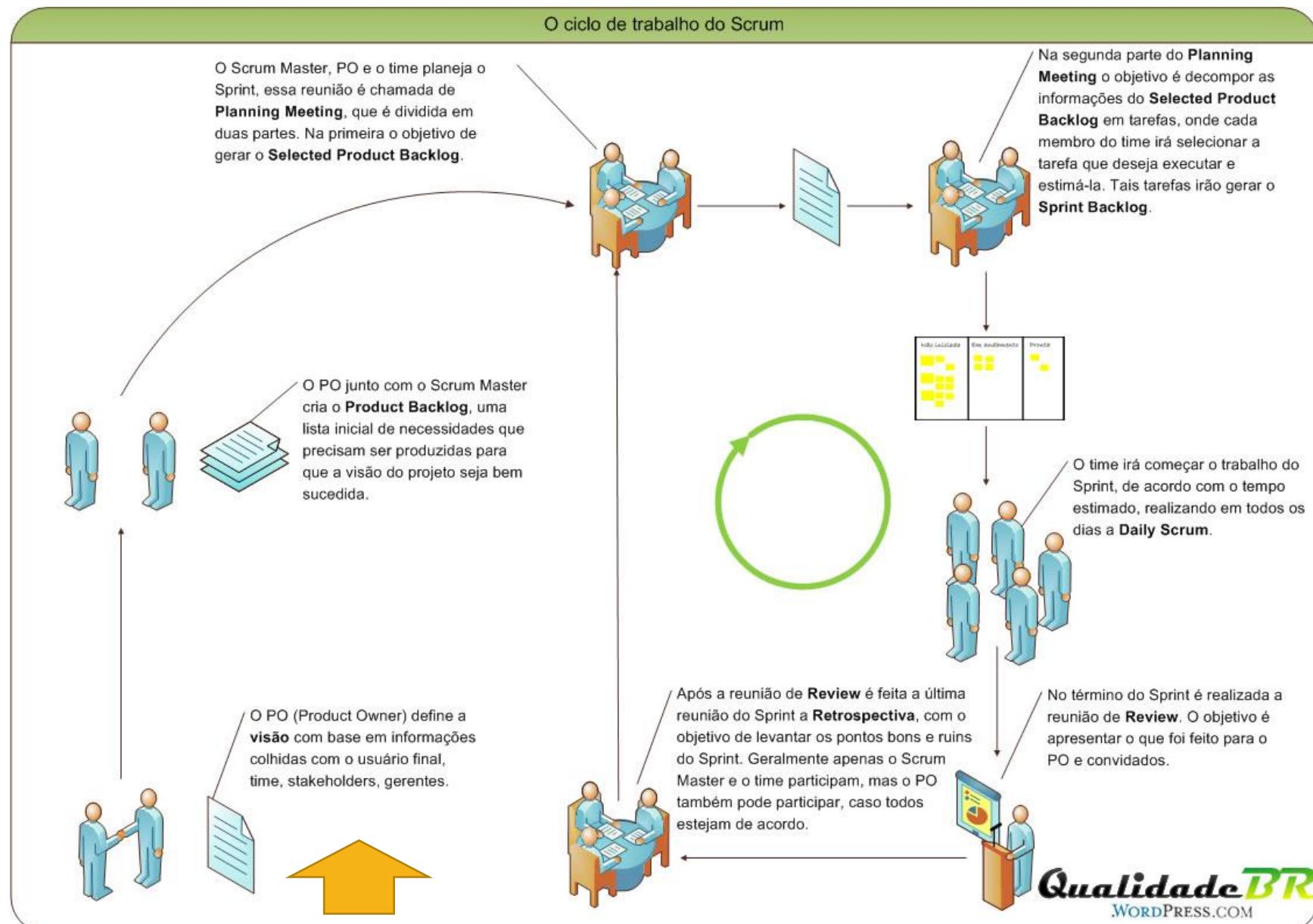


- Auto-organizado
- Sem nível hierárquico nem papéis, mas com especialidades necessárias para realizar o produto (desenvolvedores, DBA, SQA, CM...)
- Time precisa compreender a visão e os objetivos da sprint
- Completa autonomia durante a sprint
- Estão todos no mesmo barco
- Fazem de tudo para ganhar o jogo: entregar o produto!





# Ciclo







# Reuniões de Estimativa

- Preparação para o planejamento da Sprint
- Estimativa formal é feita pelo time
- Moderada pelo Scrum Master



# Planning Poker no Scrum

- Muito utilizado nas reuniões de estimativas;
- É uma técnica de estimativa baseada no consenso de toda a equipe;
- As histórias são apresentadas pelo product owner;
- É utilizado um conjunto de cartas com valores específicos que podem representar dias, horas ou tamanho;
- Sendo que normalmente esses valores recebem a escala de fibonacci.

<https://www.youtube.com/watch?v=J0FN4hRLsKg>

# Daily SCRUM Meeting

15 minutos

Mesmo local e mesmo horário todos os dias

3 Perguntas

- O que você fez desde o último SCRUM Meeting?
- O que você vai fazer até o próximo SCRUM Meeting?
- O que está lhe atrapalhando?

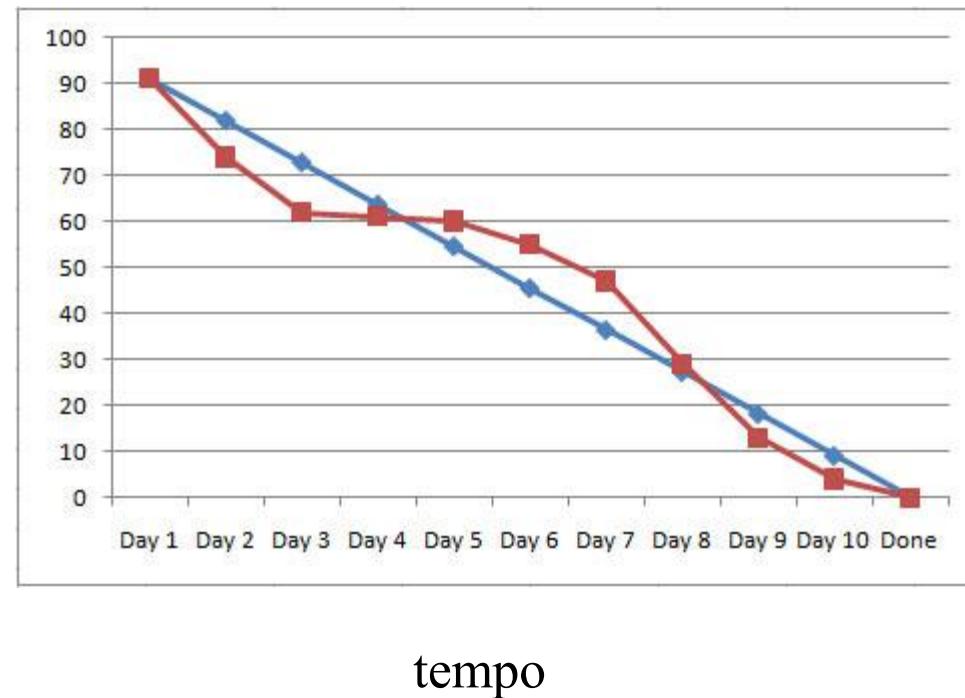
Impedimento e Decisões



# Acompanhamento

algumas pessoas utilizam horas, outros dias e assim por diante!

Burndown Chart



# Documento Tradicional X Histórias de Usuários

Os profissionais da indústria que propuseram métodos ágeis propuseram uma técnica que ficou conhecida pelo nome de **Histórias de Usuários**.

*História de Usuário = Cartão + Conversas + Confirmação*

**Conversas** entre clientes e desenvolvedores

**Confirmação**, que é basicamente um teste de alto nível

# Documento Tradicional X Histórias de Usuários

Antes de começar a escrever histórias, recomenda-se listar os principais usuários que vão interagir com o sistema

“

*Como um [papel de usuário], eu gostaria de [realizar algo com o sistema]*

“

*Como usuário típico, eu gostaria de realizar empréstimos de livros*

*Como usuário típico, eu gostaria de devolver um livro que tomei emprestado*

*Como usuário típico, eu gostaria de renovar empréstimos de livros*

*Como usuário típico, eu gostaria de pesquisar por livros*

*Como usuário típico, eu gostaria de reservar livros que estão emprestados*

*Como usuário típico, eu gostaria de receber e-mails com novas aquisições*

“

*Como professor, eu gostaria de realizar empréstimos de maior duração*

*Como professor, eu gostaria de sugerir a compra de livros*

*Como professor, eu gostaria de doar livros para a biblioteca*

*Como professor, eu gostaria de devolver livros em outras bibliotecas*

<https://engsoftmoderna.info/cap3.html>

## Histórias de Usuário

“

*Como funcionário da biblioteca, eu gostaria de cadastrar novos usuários*

*Como funcionário da biblioteca, eu gostaria de cadastrar novos livros*

“

*Como funcionário da biblioteca, eu gostaria de dar baixa em livros estragados*

*Como funcionário da biblioteca, eu gostaria de obter estatísticas sobre o acervo*

*Como funcionário da biblioteca, eu gostaria que o sistema envie e-mails de cobrança para alunos com empréstimos atrasados*

*Como funcionário da biblioteca, eu gostaria que o sistema aplicasse multas quando da devolução de empréstimos atrasados*

“

*Pesquisar por livros, informando ISBN*

*Pesquisar por livros, informando autor; retorna livros cujo autor contém a string de busca*

*Pesquisar por livros, informando título; retorna livros cujo título contém a string de busca*

*Pesquisar por livros cadastrados na biblioteca desde uma data até a data atual*

## E como seriam os testes de aceitação nas histórias?

<https://engsoftmoderna.info/cap3.html>

# Histórias de Usuário

- Não escreva histórias longas
- Não pense em detalhes técnicos
- Uma história poderá ser dividida em várias tarefas

História	Story Points	Iteração	Release
Cadastrar usuário	8	1	1
Postar perguntas	5	1	1
Postar respostas	3	1	1
Tela de abertura	5	1	1
Gamificar perguntas/respostas	5	2	1

## Task Board

Updated 2 minutes ago

Filter cards

Add cards

Fullscreen

## Project Backlog

RF9: Personalização de valor de taxa para minuto extrapolado

#9 opened by

enhancement

Sprint 2

## RF11: Fazer upgrade de ticket

#11 opened by

enhancement

Sprint 2

## RF12: Permitir consultar clientes e tickets a qualquer tempo

#12 opened by

enhancement

Sprint 2

## RF13: Geração de tickets parametrizados

#13 opened by

enhancement

Sprint 2

## RF14: Cancelamento de ticket emitido

#14 opened by

enhancement

Sprint 3

## RF15: Realizar venda/emissão de um ticket com horário personalizado

#15 opened by

Automated as To do

Manage

## Sprint Backlog

Poderia ter cards aqui

Added by:

## In progress

Aqui entram os cards em implementação

Added by:

## Aguardando Revisão

RF5: Realizar a abertura e o fechamento do caixa

#5 opened by

good first issue

Sprint 1

## Exibir lista de Tickets ativos

#19 opened by

## RF10: Gerar um novo ticket com tempo extrapolado

#10 opened by

enhancement

Sprint 2

## RF7: Gerar e imprimir recibo do ticket

#7 opened by

Sprint 1

## RF8: Emitir relatório de fluxo de caixa diário.

#8 opened by

good first issue

Sprint 1

## RF6: Realizar movimentações de caixa: suprimento e sangria

#6 opened by

good first issue

Sprint 1

## RF3: Disponibilização de tickets

#3 opened by

Automated as In progress

Manage



# Questão 1

---

Julgue o próximo item, relativos à metodologia Scrum.

Uma vez que o Scrum team é autogerenciável, com capacidade de decidir internamente como será executada a solução, cabe a ele criar e comunicar claramente os itens do product backlog.

Certo  
 Errado



# Questão 2

---

O que é um quadro Scrum?

- A) Uma ferramenta para monitorar e relatar problemas do projeto.
- B) Um sistema para desenvolvimento de sistemas do projeto.
- C) Um método para priorizar requisitos do projeto com base em sua importância.
- D) É uma ferramenta visual para organizar tarefas e monitorar o andamento do projeto e acompanhar o progresso das tarefas.
- E) Um sistema para rastrear o tempo gasto em cada fase do ciclo de vida do projeto.



# Questão 3

---

Na metodologia Scrum, os membros do time apresentam sua contribuição para o produto na etapa denominada

- A) retrospectiva.
- B) sprint planning.
- C) **sprint review.**
- D) burndown chart.
- E) daily.



# Questão 4

---

Julgue o seguinte item, a respeito dos métodos e das práticas ágeis para desenvolvimento de software.

O dono do produto é responsável por garantir que os ritos do scrum sejam adotados corretamente.

Certo

Errado



# Questão 5

---

Julgue o seguinte item, a respeito dos métodos e das práticas ágeis para desenvolvimento de software.

Uma equipe scrum é formada pelos seguintes artefatos: um dono do produto, o scrum master e a equipe de desenvolvimento.

Certo  
**Errado**



# Questão 6

---

Julgue o seguinte item, a respeito dos métodos e das práticas ágeis para desenvolvimento de software.

No método Scrum, o backlog da sprint é um documento entregue pela equipe de desenvolvimento e que contém os resultados alcançados durante a sprint.

Certo  
**Errado**



# Questão 7

---

Assinale a alternativa que apresenta corretamente o propósito de um Product Backlog no Scrum.

- A) Uma lista de tarefas específicas para cada membro da equipe executar durante uma Sprint.
- B) Uma descrição detalhada de todas as atividades realizadas durante o projeto.
- C) Um registro de todas as decisões tomadas pelo Scrum Master durante o projeto.
- D) Uma lista de itens de trabalho concluídos pelos membros da equipe.
- E) Uma lista ordenada de requisitos e funcionalidades desejadas para o produto.



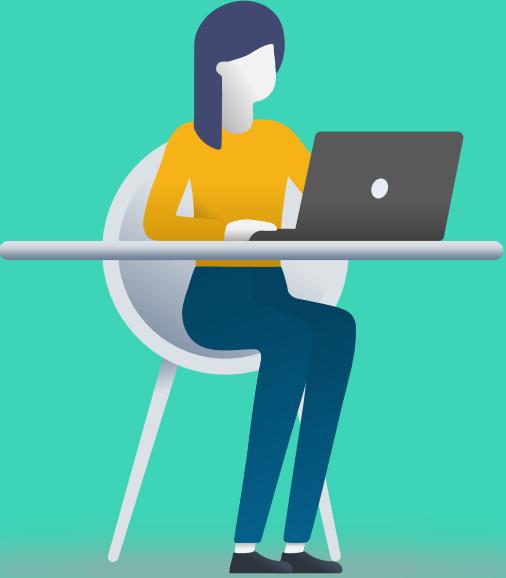
# Questão 8

---

Scrum é uma metodologia ágil para o desenvolvimento de sistemas que se baseia em histórias de usuário, priorizadas em termos do valor para o cliente, com o desenvolvimento fragmentado em períodos de prazo determinado, conhecidos como sprints. Tem sido considerada uma boa estratégia, trazendo como resultado um retorno em menor tempo e com melhor avaliação por parte do cliente.

Uma das características dos artefatos definidos no SCRUM é:

- A) cada sprint dura de 2 a 4 meses;
- B) utiliza ampla documentação, definindo alto formalismo nos processos;
- C) as histórias de usuário são agrupadas e priorizadas em uma lista denominada backlog;
- D) as entregas são feitas diariamente, mesmo que os testes não tenham sido efetuados;
- E) o desenvolvimento segue o modelo clássico em cascata, com testes efetuados ao final da implementação completa do sistema.



## DICAS DE LEITURA +:

### SAFe - Scaled Agile Framework

<https://www.mjvinnovation.com/pt-br/blog/scaled-agile-framework-safe/>

Kanban

<https://www.mjvinnovation.com/pt-br/blog/quadro-kanban/>

## ATIVIDADE +:

Criar contas em:

<https://trello.com/pt-BR>

<https://github.com/>