

---

---

# Produto

— Pessoas e Processos —

---

---

O que é produto?

**Atenção, se liga  
aí que é hora da  
revisão!**

# Pessoas





PO/PM



F.E. DEV



TESTER

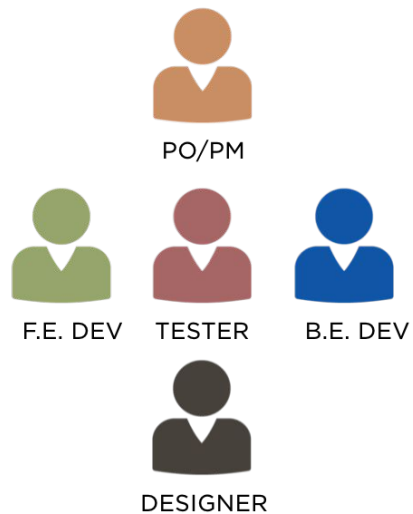


B.E. DEV



DESIGNER

Vários  
profissionais com  
skills  
complementares  
que se juntam  
para atacar uma  
feature específica  
do produto  
formam um  
**SQUAD.**



Vários  
profissionais com  
skills  
complementares  
que se juntam  
para atacar uma  
feature específica  
do produto  
formam um  
**SQUAD.**

**Tudo gira em torno da  
equipe de produtos!**

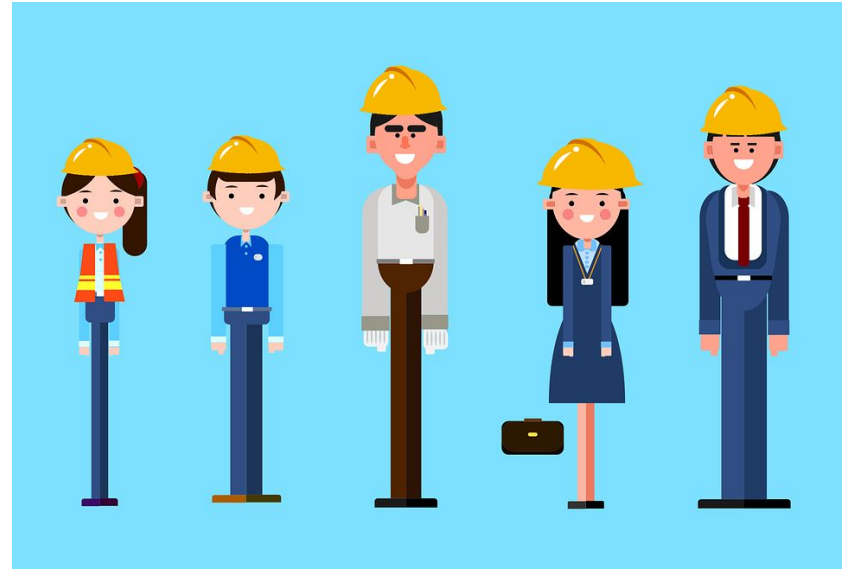
# Missionários

*“Nós precisamos de equipes de missionários, não de equipes de mercenários.”*

- Mercenários constroem tudo o que lhes pedem. Missionários acreditam verdadeiramente na visão e são comprometidos em resolver problemas de seus clientes;

# Composição da Equipe

- Entre 2 e 10 engenheiros
- Um designer
- Um gerente de produto



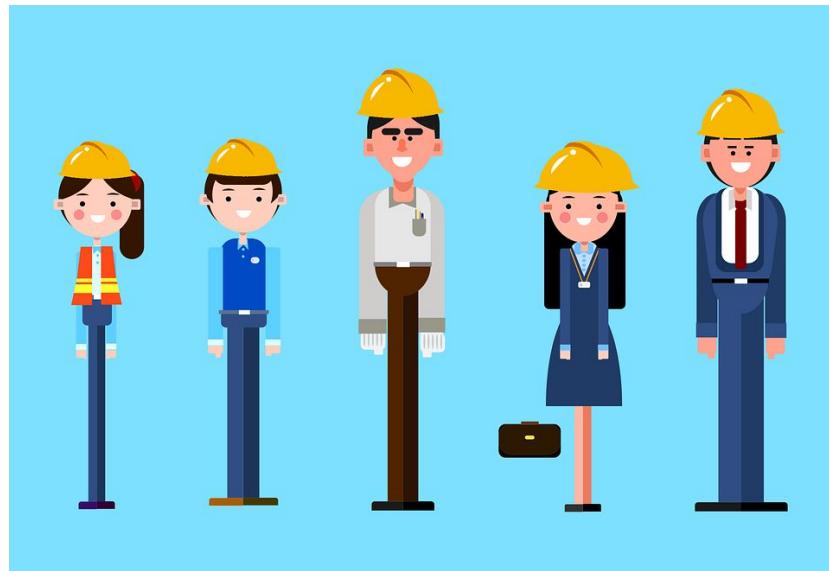
# Composição da Equipe

- Entre 2 e 10 engenheiros
- Um designer
- Um gerente de produto

**eng. de automação de teste**

**gerente de marketing**

**analista de dados**





# Composição da Equipe

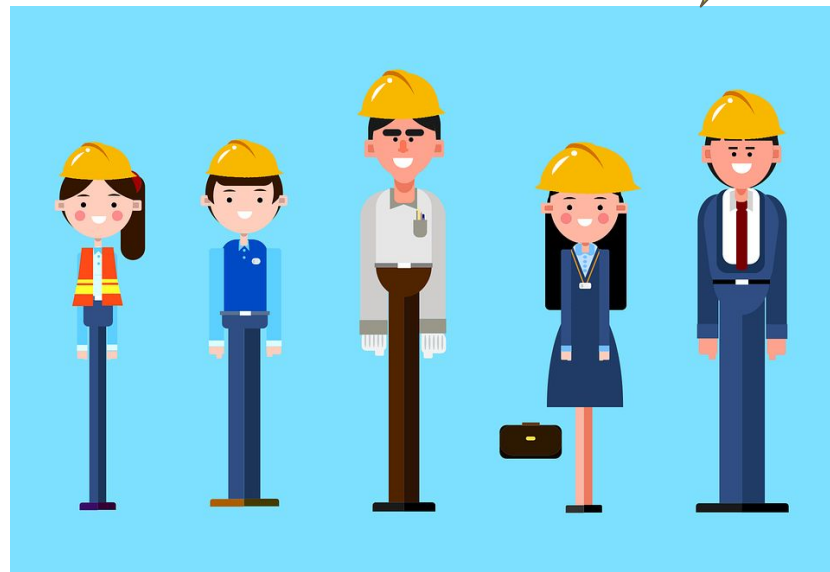
- Entre 2 e 10 engenheiros
- Um designer
- Um gerente de produto

**eng. de automação de teste**

**gerente de marketing**

**analista de dados**

**EMPODERAMENTO DA EQUIPE**

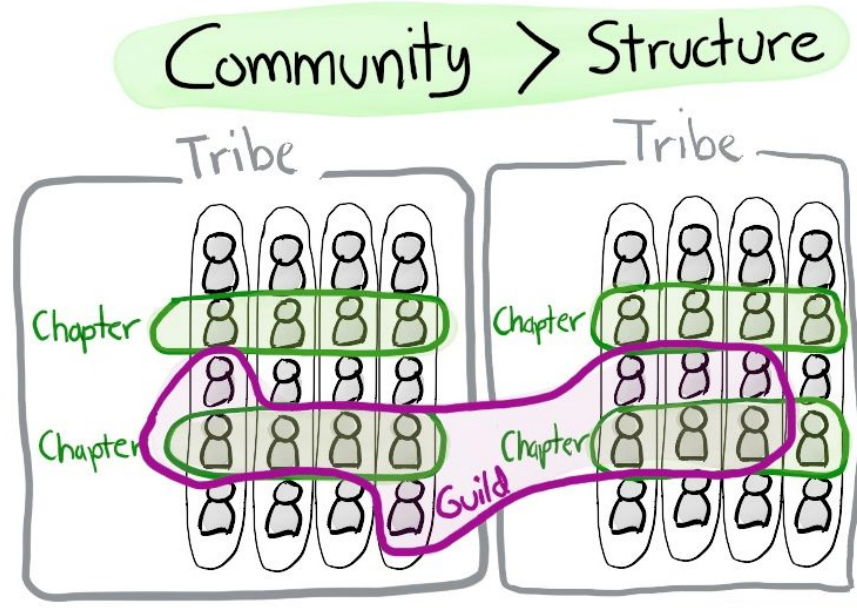


**Quantas pessoas devem ter numa equipe?**

Quantas pessoas devem ter numa equipe?



<https://youtu.be/hQDbLYvY9RY>

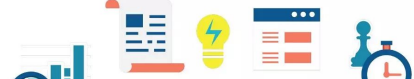


# Formas de trabalhar do Gerente de Produto

1. Gerente de produto pode escalar todo problema e decisão até o CEO
2. Gerente de produto pode exigir uma reunião com todos os stakeholders em uma sala e depois deixa-os resolver
3. O gerente de produto pode fazer o seu trabalho



# Formas de trabalhar do Gerente de Produto

1. Gerente de produto pode escalar todo problema e decisão até o CEO (adm de backlog)
  2. Gerente de produto pode exigir uma reunião com todos os stakeholders em uma sala e depois deixa-os resolver(adm de roadmaps)
  3. O gerente de produto pode fazer o seu trabalho
- 



# Gerente de Produto

- Quando o produto tem sucesso, é porque todos no time fizeram o que precisavam fazer. Mas, quando o produto fracassa, a culpa é do gerente de produto

# Gerente de Produto

- Quando o produto tem sucesso, é porque todos no time fizeram o que precisavam fazer. Mas, quando o produto fracassa, a culpa é do gerente de produto
- Avaliar oportunidades
- Garantir que vale a pena desenvolver o que está no backlog

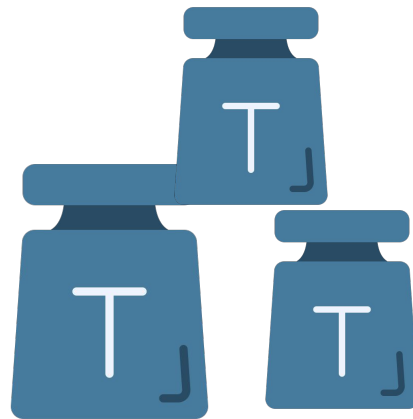


# Habilidades do Gerente de Produto

- Profundo conhecimento do cliente
- Profundo conhecimento dos dados
- Profundo conhecimento do Negócio
- Profundo conhecimento do seu mercado e indústria

# Habilidades do Gerente de Produto

- Profundo conhecimento do cliente
- Profundo conhecimento dos dados
- Profundo conhecimento do Negócio
- Profundo conhecimento do seu mercado e indústria



# Habilidades do Gerente de Produto

- Inteligência: curioso, aprender e aplicar rápido
- Criatividade: pensar fora da caixa
- Persistência: empurrar a empresa para fora da zona de conforto

# PO vs PM

PO: é o nome do papel em um time Agile. É a pessoa responsável pelo backlog do produto.

O PM dever ser também o PO.

# Designer de Produto

Responsabilidades:

- Descoberta do produto
- Experiência do usuário
- Prototipagem
- Testes de usuário
- Design visual e de interação

# Designer de Produto

Responsabilidades:

- Descoberta do produto
- Experiência do usuário
- Prototipagem
- Testes de usuário
- Design visual e de interação

**Ter um design é um Diferencial competitivo**

# Engenheiros (Desenvolvedores; Programadores)

- Missionários ou Mercenários
- Autonomia
- Parceria

# Gerentes de Marketing

- Fundamental para disseminar o produto





# Papéis de Apoio

- Pesquisadores de usuário
- Analista de dados
- Engenheiros de automação e teste

# O Produto certo

No que a equipe deve trabalhar?

# O Produto certo

No que a equipe deve trabalhar?

ROADMAP → lista priorizada de funcionalidades → é uma boa ideia?

# O Produto certo

No que a equipe deve trabalhar?

ROADMAP → lista priorizada de funcionalidades → é uma boa ideia?

Típicos roadmaps são a causa raiz de muito desperdício e esforços fracassados em empresas de produtos.

PORQUE AINDA EXISTEM ROADMAPS?

# O Produto certo

Problemas:

- metade das ideias irão falhar
- várias iterações até a entrega

O problema não é o roadmap, mas a partir do momento que você o faz, as pessoas da empresa o interpretam como um compromisso

# Visão e Estratégia de Produto

- **Visão:** descreve o futuro que nós estamos tentando criar
  - Comece pelo porque
  - Apaixone-se pelo problema, não pela solução
  - Não tenha medo de pensar grande na visão
  - Não tenha medo da disrupção, se você não fizer, alguém fará
  - A visão de produto precisa inspirar
  - Determine e adote tendências relevantes e significativas
  - Vá para onde as coisas estão caminhando
  - Seja teimoso com a visão, mas flexível com os detalhes
  - Qualquer visão de produto é um voto de confiança
  - Evangelize contínua e incansavelmente

# Visão e Estratégia de Produto

- **Estratégia:** sequência de produtos ou lançamentos que planejamos entregar até alcançar a visão de produto
  - Foca um mercado-alvo ou persona de cada vez
  - Estratégia de produto precisa estar alinhada com estratégia de negócio
  - Estratégia de produto precisa estar alinhada com estratégia go-to-market e vendas
  - Seja obcecado por clientes, não por concorrentes
  - Comunique a estratégia para a empresa

# Algumas técnicas

Descoberta de produtos

Descoberta de clientes

Protótipo

Testes



**Qual o seu produto?**

# Referências interessantes

- <https://medium.com/mulheres-de-produto/traduzindo-alguns-termos-da-%C3%A1rea-de-produto-5cb145c318cf>
- [https://www.infoq.com/br/minibooks/priming-kanban-jesper-boeg/#idp\\_register/](https://www.infoq.com/br/minibooks/priming-kanban-jesper-boeg/#idp_register/)
- <https://www.cursospm3.com.br/blog/guia-de-frameworks-para-product-managers/>
-

---

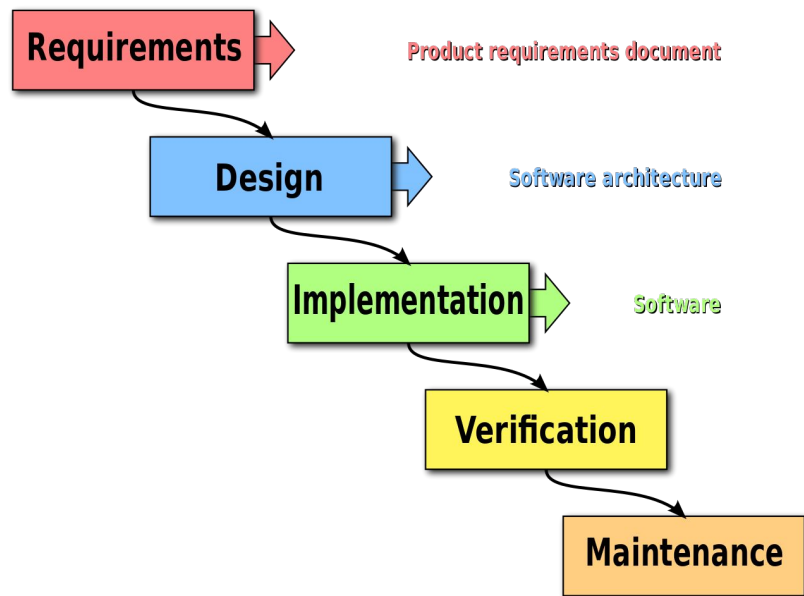
---

# Metodologias Ágeis

---

---

# Como era feito e às vezes ainda é...



- Um processo em cascata requer que uma equipe escreva uma descrição do software no início do projeto, e então construa exatamente o que eles escreveram.
- O processo em cascata dificulta a resposta à mudança devido ao foco na documentação em vez da colaboração.
- As equipes que fazem o trabalho em cascata o fazem adotando práticas e princípios de software eficazes, especialmente aqueles que melhoram a comunicação.

# Problemas 'comuns'

Prazo



< Nós frequentemente entregamos fora do prazo >

Demanda



< Nosso time está atolado de trabalho >

Organização



< Não sabemos quem está fazendo o que >

Tarefas



< O trabalho chega ao time por várias fontes >

# Um pouco 'sobre'

## Step 01

As metodologias de desenvolvimento ágil tiveram seu início na década de 90 em reação aos métodos tradicionais

## Step 02

As metodologias de desenvolvimento ágil focam em realizar micro entregas, uso de práticas ágeis e comunicação em tempo real

## Step 03

Flexibilidade com mudanças do projeto e o cliente já começa a fazer uso do software

## Step 04

Manifesto ágil (versão em português) - [link](#)

# Ágil

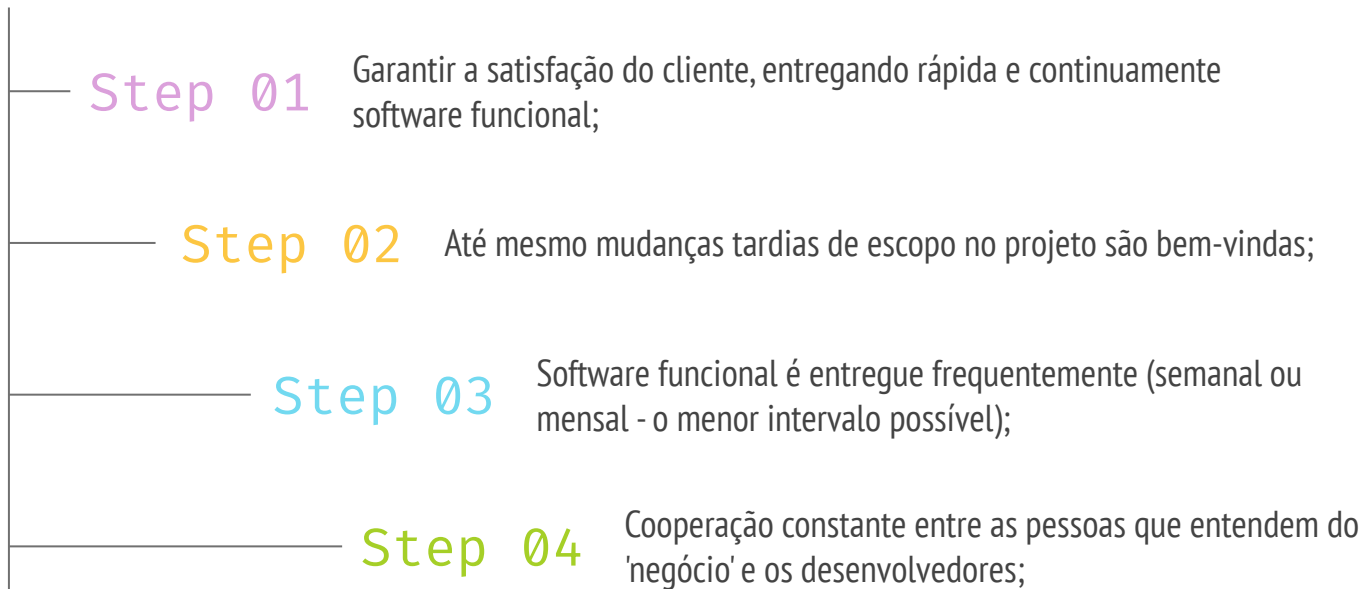
- Ágil é diferente de veloz
- É diferente de rapidinho

**MINDSET!**

## MANIFESTO ÁGIL



# 12 PRINCÍPIOS



[https://pt.wikipedia.org/wiki/Manifesto\\_%C3%81gil](https://pt.wikipedia.org/wiki/Manifesto_%C3%81gil)



# 12 PRINCÍPIOS

## Step 05

Projetos surgem por meio de indivíduos motivados, devendo existir uma relação de confiança;

## Step 06

A melhor forma de transmissão de informação entre desenvolvedores é através da conversa 'cara a cara'

## Step 07

Software funcional é a principal medida de progresso do projeto;

## Step 08

Novos recursos de software devem ser entregues constantemente. Clientes e desenvolvedores devem manter um ritmo até a conclusão do projeto;

[https://pt.wikipedia.org/wiki/Manifesto\\_%C3%81gil](https://pt.wikipedia.org/wiki/Manifesto_%C3%81gil)

# 12 PRINCÍPIOS

- Step 09 Design do software deve prezar pela excelência técnica;
- Step 10 Simplicidade – a arte de maximizar a quantidade de trabalho que não é feito – é essencial;
- Step 11 As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis;
- Step 12 Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento.

[https://pt.wikipedia.org/wiki/Manifesto\\_%C3%81gil](https://pt.wikipedia.org/wiki/Manifesto_%C3%81gil)

# Ágil



# Um passo atrás

**1960:** Kanban

**1969:** design thinking

**1975:** computador pessoal

**1986:** scrum

**1994-2000:** bolha da internet

**1998:** google

**2001:** manifesto ágil

**2007:** startup-weekend

**2010:** business model canvas

# Ágil

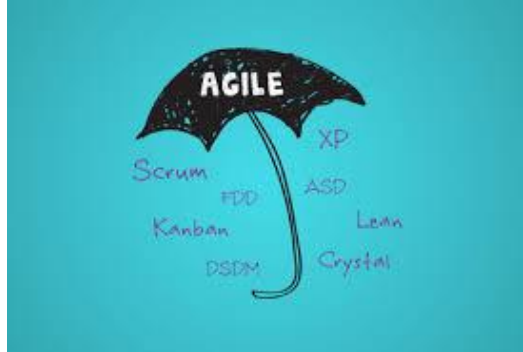


A ilusão!

<https://youtu.be/z3ueD1s9vLw>

**REVELX**  
REALIZING GROWTH

# Frameworks



## Scrum

< Metodologia baseada em ciclos de entrega, possui práticas e métricas para mensurar o trabalho >



## Kanban

< As tarefas são entregues por ordem de prioridade, o fluxo de trabalho é apresentado em um quadro >



## Extreme Programming (XP)

< Prioriza a entrega das tarefas de maior interesse pelo cliente e possui práticas que ajudam a fazer entregas rapidamente >

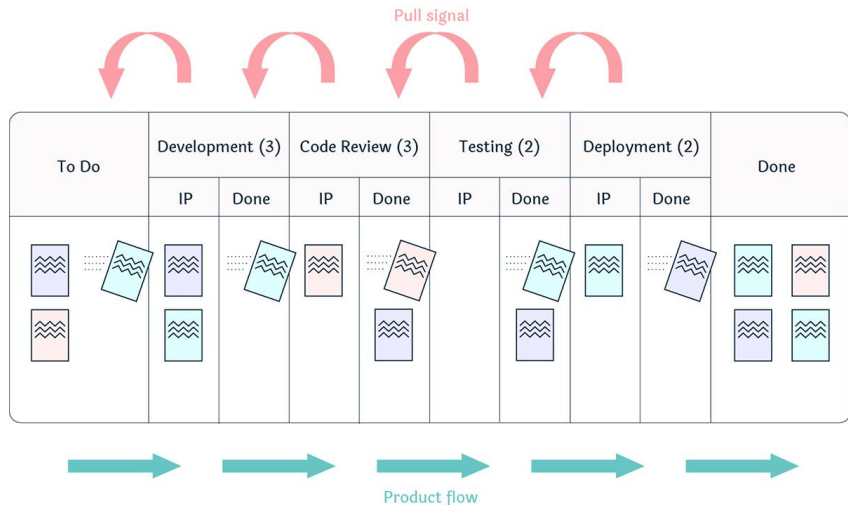
# Scrum



Utiliza etapas ou ciclos de desenvolvimento – denominadas *sprints* – que permitem qualidade nas entregas e possibilidade de mudança de requisitos ao longo do processo. O framework sustenta-se em pilares e papéis bem definidos: os clientes se tornam parte da equipe de desenvolvimento e podem validar ou redefinir entregas. Desta forma, os riscos são melhores trabalhados e reduzidos, já que os progressos e atrasos são monitorado

# Kanban

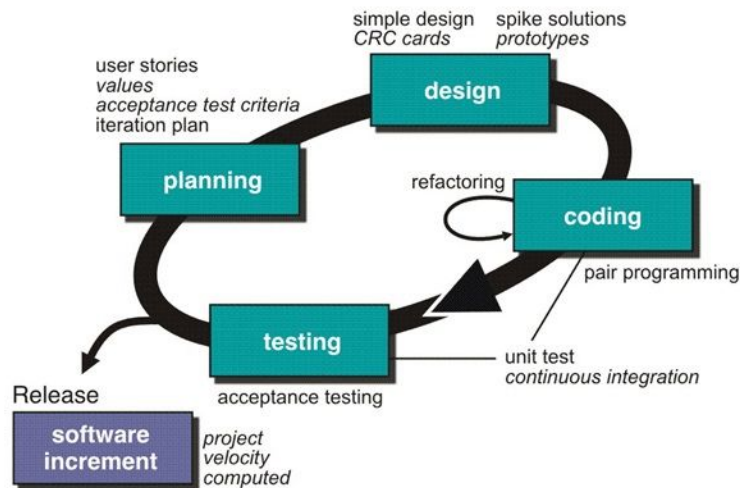
<https://www.infoq.com/br/minibooks/priming-kanban-jesper-boeg/>



Ferramenta introduzida, primeiramente, para dar uma panorâmica do fluxo de tarefas realizadas nas linhas de montagem do Sistema Toyota de Produção: o objetivo é que todo **trabalho a ser realizado seja visualizado pela equipe**, com as atividades divididas e direcionadas à cada responsável, obedecendo um processo de “a realizar” (to do), “em andamento” (WIP – work in progress) e “já realizada” (done).



# Extreme Programming



Para garantir projetos de software valiosos, leva ao extremo um conjunto de práticas como **testar, revisar, desenvolver, integrar, simplificar e realizar iterações curtas**. Este framework garante eficiência ao antecipar as mudanças necessárias para reduzir custos posteriores ao ciclo de vida do produto: o mesmo é validado pelo cliente durante seu desenvolvimento. Assim, tanto o time desenvolvedor, quanto o dono do negócio, devem estar em sintonia para que os feedbacks sejam mais assertivos.

# Scrum



# Scrum

< Papéis >

## Product owner

< Responsável por direcionar o projeto, definir cronograma e prioridades >

## Scrum master

< Busca manter as práticas, ser um facilitador e acompanhar o andamento do projeto >

## Scrum team

< Toda a equipe de desenvolvimento e segue as orientações do product owner >

# Scrum

< Organização das atividades >

## Product backlog

< Todas as atividades do projeto que deverão ser realizadas >

## Sprint backlog

< Grupo de atividades priorizadas pelos PO para realizar no sprint >

# Scrum

< Organização do tempo >

## Sprint planning

< Planejamento da sprint, onde o PO define quais atividades do P. backlog vão para o S. backlog >

## Sprint

< Ciclo de tempo definido de 2 a 4 semanas para realizar as atividades do S. backlog >

# Scrum

< Cultura do feedback >

## Daily scrum

< Reunião diária de 15 min para alinhamento da equipe sobre progresso e impedimentos >

## Sprint review & retrospectiva

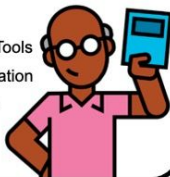
< Reuniões feita ao fim da sprint, onde é observado quais atividades foram entregues e quais impedimentos ocorreram. Toda a equipe decide quais ações deram certo e serão mantidas, quais deram errado e a possibilidade de testar novas ações >

# Produto no Scrum

- Usuários + Necessidade → oportunidade
- Visão
- Vale a pena investir no produto? Reduzir a incerteza
- Feedbacks e entregas rápidas
- Qual é o menor produto capaz de gerar valor? A cada sprint uma versão do produto

## Agile Manifesto

- **Individuals and Interactions** over Processes and Tools
- **Working Software** over Comprehensive Documentation
- **Customer Collaboration** over Contract Negotiation
- **Responding to Change** over Following a Plan



## Transparency, Inspect and Adapt

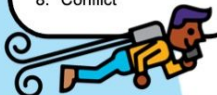
Scrum considers complex product development as empirical projects. An empirical project is managed by learning from observations and experiments. This data can be analyzed and based on that new learnings can be done.

In Scrum this is translated to transparency, inspection and adaptation.



## Eight Fundamentals of Agile

1. People
2. Functionality
3. Quality
4. Tools
5. Time
6. Value
7. Process
8. Conflict



## Scrum Process



# Agile Product Development

## Scrum Values

1. Commitment
2. Focus
3. Openness
4. Respect
5. Courage



The **Product Owner** is responsible for maximizing the value of the product resulting from work of the Scrum Team.

The **Scrum Team** consists of one Scrum Master, one Product Owner, and Developers. It is a cohesive unit of professionals focused on one objective at a time, the Product Goal.

The **Scrum Master** is accountable for establishing Scrum as defined in the Scrum Guide and is accountable for the Scrum Team's effectiveness.



MANAGEMENT 3.0