



# Metodologias Ágeis ✓

Material das Aulas

Metodologias Ágeis

Introdução

Nossos caminhos podem mudar

As coisas mudam, e se adaptam

Como as empresas trabalham

Modelo tradicional x Times ágeis

Times ágeis - interativo e incremental

Organização Tradicional

Organização Ágil

Modelo Tradicional

Modelo Agile

Ágil não é só pra Tecnologia

Manifesto Ágil

Ser Ágil ≠ Fazer Ágil

Linha histórico do Ágil

Manifesto Ágil

Princípios ágeis

Agilidade x Rapidez

1- Satisfação do cliente

2 - Mudança em favor da vantagem competitiva

3 - Prazos curtos

4 - Trabalho em conjunto

5 - Ambientação e suporte

6 - Falar na cara

7 - Funcionalidade

8 - Ambiente de sustentabilidade:

9 - Padrões altos de tecnologia e design:

10 - Simplicidade

11 - Autonomia

12 - Reflexões para otimizações

Agilidade é...

Transformação Ágil

Métodos Ágeis

Os principais Frameworks Ágeis

[Modelo em Cascata x Modelo Ágil](#)

## [SCRUM](#)

[O que é?](#)

[Roteiro do SCRUM](#)

[O que é Backlog?](#)

[Papéis no SCRUM](#)

[Rituais e Cerimônias](#)

[Artefatos](#)

[Guia para Desenvolvimento de Software ilustrado](#)

[Scrum na Prática I](#)

[Fluxo do Produto](#)

[Waterfall x Scrum](#)

[Refinamento do Backlog](#)

## [Kanban](#)

[O que é?](#)

[Princípios](#)

[Serve para...](#)

[Kanban no Scrum](#)

[Como usar o Kanban?](#)

## **Material das Aulas**

- [Slide](#)

## **Metodologias Ágeis**

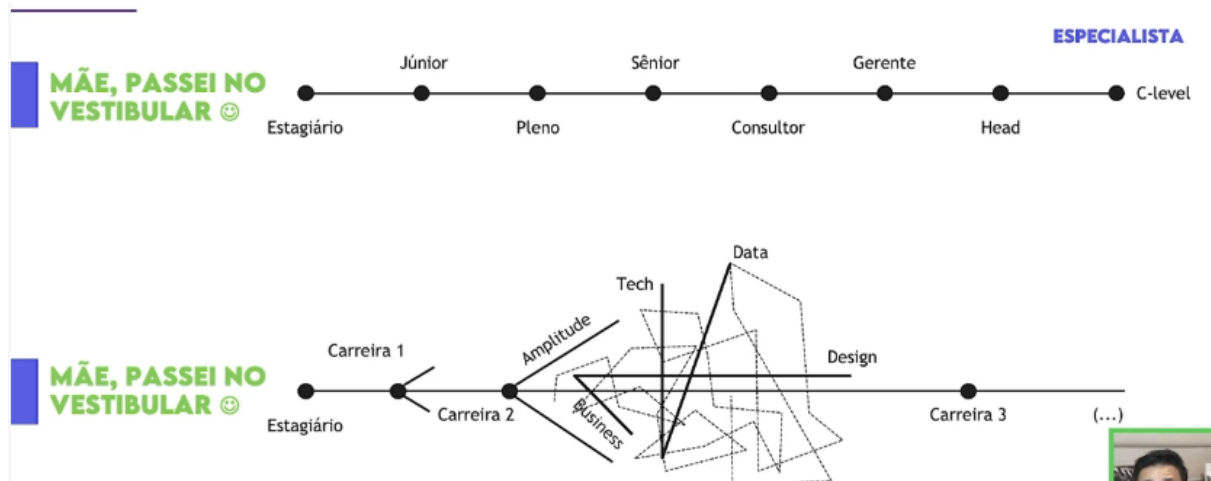
---

### **Introdução**

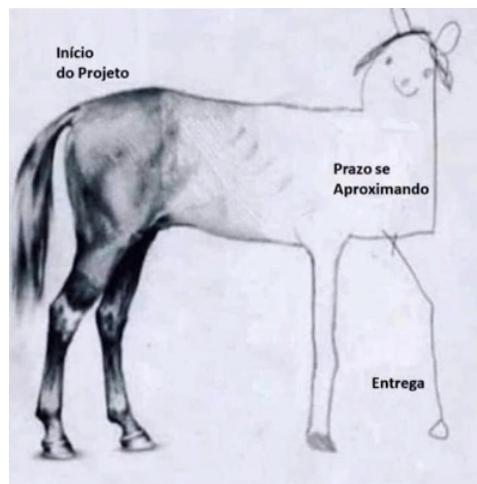
- Durante o desenvolvimento de projetos na área de tecnologia, o maior desafio que temos é manter o planejamento e conseguirmos gerenciar todas as mudanças que podem ocorrer no meio do caminho.
- Metodologias ágeis vêm sendo utilizadas pela maioria das empresas para enfrentar esse desafio.
- Vamos conhecer um pouco mais sobre o que são os métodos ágeis e como implementar os 4 principais!

### **Nossos caminhos podem mudar**

- Eu mesmo, comecei em Direito e agora estou em tecnologia 😊



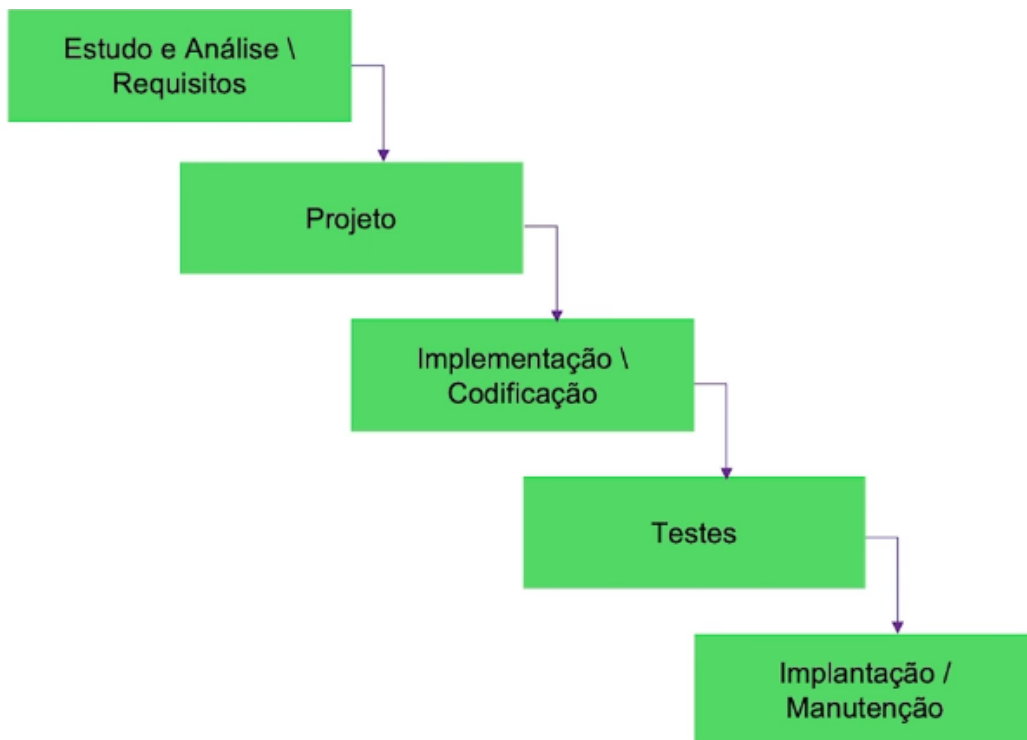
As coisas mudam, e se adaptam



## Como as empresas trabalham

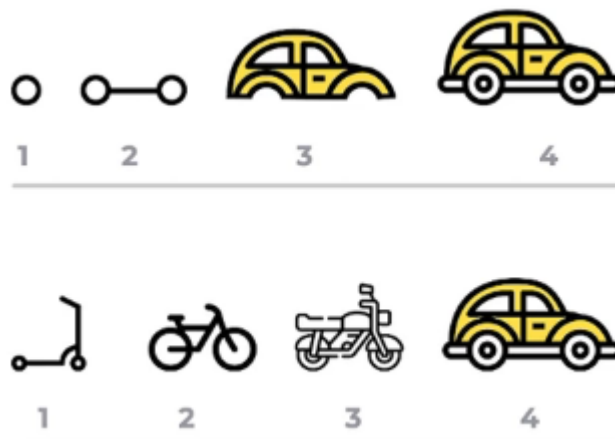
### Modelo Cascata/Waterfall

- Antes para os requisitos seguirem adiante, precisamos terminar uma etapa para seguir para outra.
- Era de forma **passo a passo**, não de forma paralela e incremental como na metodologia ágil
- O grande problema é que os requisitos não podiam ser revalidados, só após passar por todas as etapas.
- A adaptabilidade aqui não existia, muitos produtos finais entregues as empresas mal utilizavam, pois daqui que fosse entregue, a realidade já era outra.



## Modelo tradicional x Times ágeis

- O produto final pode ser diferente, a depender da necessidade do cliente
- No modelo Waterfall só podemos entregar valor na última etapa.



- Adicionando o MVP, fazendo o produto mais simples possível e depois vamos aprimorando.
- Em todas etapas o cliente já está utilizando e dando feedback, podemos entregar valor em todos os momentos.



## Times ágeis - interativo e incremental

### Vantagens

1. Entregas contínuas
2. Entregas com valor
3. Aprendizado Contínuo
4. Feedback constante do cliente
5. Flexibilidade na revisão de prioridade.

## Organização Tradicional



## Organização Ágil



## Modelo Tradicional



## Modelo Agile

- Tem o papel do facilitador que ouve a equipe e passa para o cliente e vice-versa
- Fala com a equipe e mostra a demanda, a equipe diz se consegue entregar em determinado prazo ou não
- O cliente pode dar o feedback e o valor entregue pode mudar de acordo com a necessidade do cliente



## Ágil não é só pra Tecnologia

- Apesar de ter se originada em empresas desse contexto, todas as áreas de empresas estão adotando a metodologia ágil
- A metodologia ágil é um ideal.
- É necessário definir prioridades, pra tudo na vida.



O ágil permite modelos para organizarmos melhor nossos objetivos.

- Podemos trazer o ágil para tudo na nossa vida.



Exemplo do ágil na organização de tarefas diárias.

- Adaptar e ser ágil
- O mais importante da agilidade não é ser rápido, mas entregar valor ao cliente.

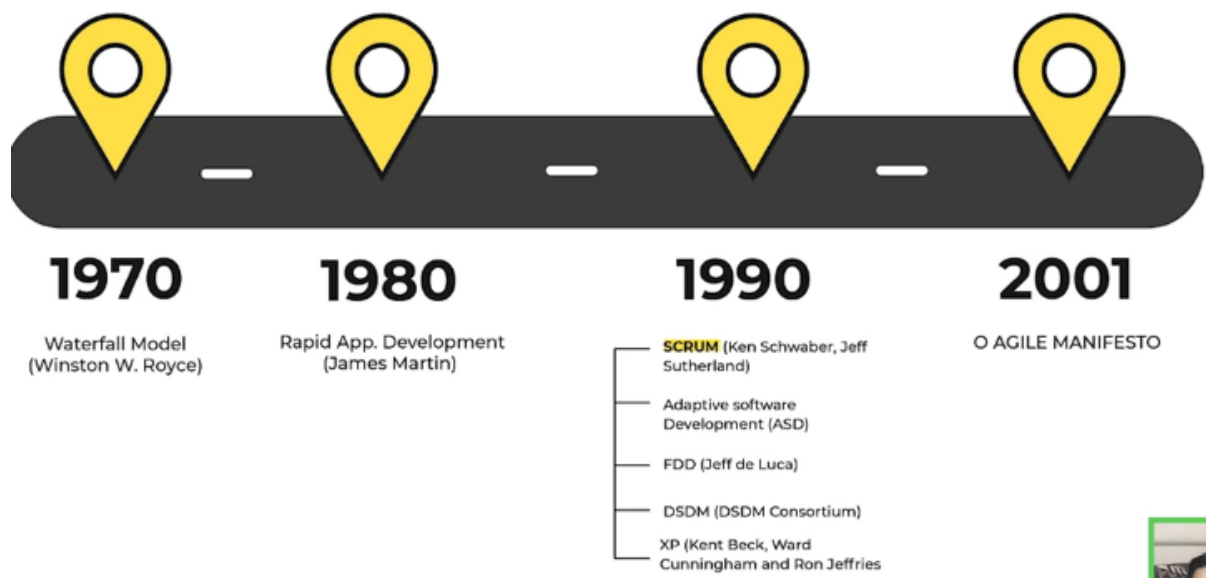
## Manifesto Ágil

### Ser Ágil ≠ Fazer Ágil

- Agilidade é um mindset, estabelecido por 4 valores e fundamentados por 12 princípios e manifestado através de diversas práticas diferentes.

### Linha histórico do Ágil

- Começou para área de Software e se expandiu



### Manifesto Ágil

- Em 2001 um grupo de 17 desenvolvedores se reuniram no estado norte-americano de Utah para discutir novas ideias sobre os atuais processos de desenvolvimento de software.
- A partir desta reunião eles elaboraram um documento que serviria como grito de guerra aos novos processos de desenvolvimento de software.
- Este manifesto é composto por 12 princípios ágeis que iremos ver a seguir.
- Mas caso queira acessar o site oficial do manifesto [clique aqui](#).





- Vamos dar prioridade, mas não esquecer com esse *mais que*.

## Princípios ágeis

### Agilidade x Rapidez

- Não é a mesma coisa
- Ágil e rápido são duas coisas diferentes.
- Para cada tamanho de empresa temos uma agilidade.
- Para cada caminho traçado também existe agilidade.

### 1- Satisfação do cliente

A maior prioridade está em satisfazer o cliente por meio da entrega adiantada e contínua de software de valor.

### 2 - Mudança em favor da vantagem competitiva

- Mudanças de requisitos são bem-vindas, mesmo em fases tardias do desenvolvimento.

### 3 - Prazos curtos

- Entregar software em funcionamento com frequência, desde a cada duas semanas até a cada dois meses, com uma preferência por prazos mais curtos.

### 4 - Trabalho em conjunto

- Tanto pessoas relacionadas a negócios como desenvolvedores devem trabalhar em conjunto, diariamente, durante todo o curso do projeto.

### 5 - Ambientação e suporte

- Para construir projetos ao redor de indivíduos motivados, é preciso dar a eles o ambiente e o suporte necessários, confiando que farão seu trabalho.

## **6 - Falar na cara**

- O método mais eficiente de transmitir informações tanto externas como internas para um time de desenvolvimento é por meio de uma conversa cara a cara.

## **7 - Funcionalidade**

- Um software funcional é a medida primária de progresso.

## **8 - Ambiente de sustentabilidade:**

- Processos ágeis promovem um ambiente sustentável, com patrocinadores, desenvolvedores e usuários sendo capazes de manter passos constantes.

## **9 - Padrões altos de tecnologia e design:**

- A contínua atenção à excelência técnica e ao bom design aumenta a agilidade.

## **10 - Simplicidade**

- Fazer algo simples é dominar a arte de maximizar a quantidade de trabalho que não precisou ser feito.

## **11 - Autonomia**

- As melhores arquiteturas, os requisitos e os designs emergem de times auto organizáveis.
- Errar é bom, errar rápido é aprender mais rápido.
- Não deixe pra demorar de errar, o grande problema do modelo de cascata é que você via que errava só no final.

## **12 - Reflexões para otimizações**

- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

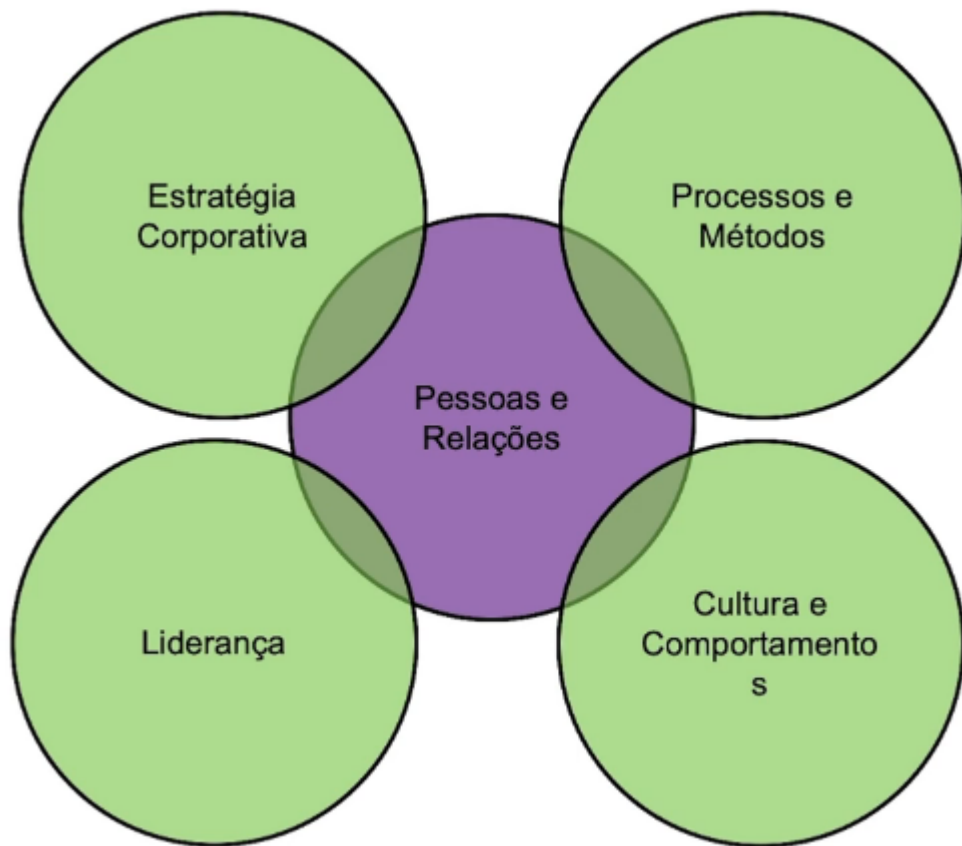


## Agilidade é...

- Responder a mudanças, sem traumas
  - Mudanças acontecem a todo momento
  - Foco no Valor
- Entregar mais rápido (frequência)
  - Visualização antecipada dos resultados
  - Feedback antecipado
- Assertividade
  - Maior frequência de planejamento
  - Maior qualidade das entregas

## Transformação Ágil

- Transformação ágil é caminhar para transformação digital



- Ágil não é só SCRUM, ágil é uma **metodologia** e **mindset**.



Não é fazer ÁGIL é ser ágil

## Métodos Ágeis

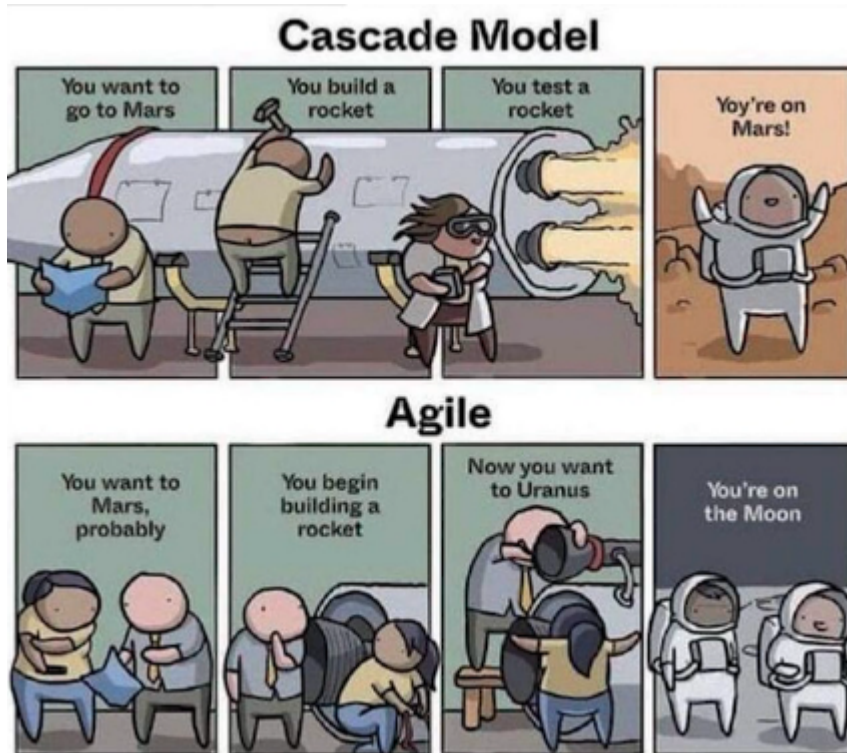
### Os principais Frameworks Ágeis

- SCRUM
- KANBAN



Não temos chefes nas metodologias ágeis, temos **FACILITADORES**

### Modelo em Cascata x Modelo Ágil

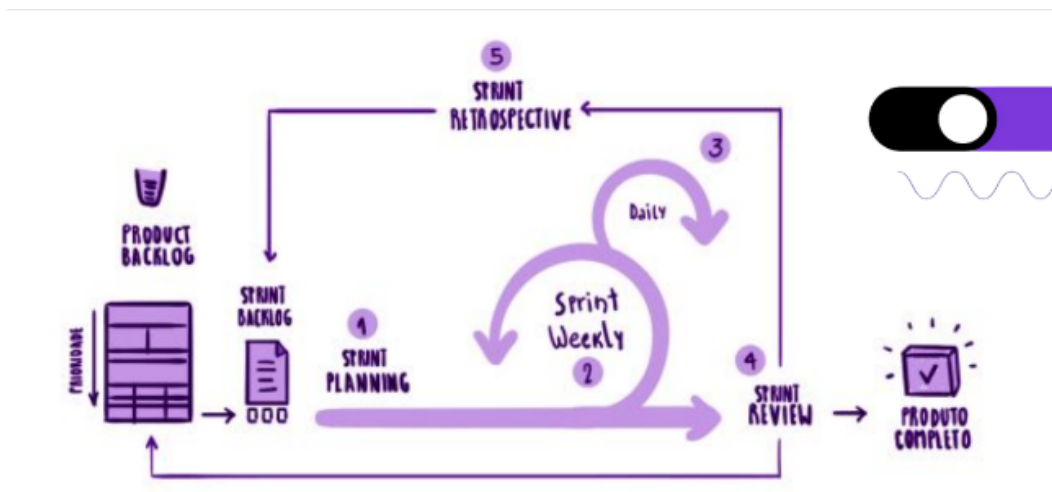


## SCRUM

### O que é?

- O framework Scrum é inspirado no Rugby, o trabalho sempre é feito em equipe com planejamento de tarefas e entregas.
- Temos algumas nomenclaturas específicas dessa metodologia e vamos ver o significado de cada uma delas.

### Roteiro do SCRUM



## O que é Backlog?

- Lista de coisas que o produto precisa ter.

## Papéis no SCRUM

### Product Owner

- Gerencia o Product Backlog
- O P.O. é o profissional que está diretamente em contato com o cliente, entende bem o objetivo final do projeto e o valor real do produto.
- Ele cria e gerencia o Product Backlog, priorizando as demandas de acordo com a necessidade do cliente.

- Mais conhecido como PO
- Dono do Backlog
- Conhece bem o produto e o negócio
- Garante ROI do produto
- Detalhar o Product Backlog
- Criar, manter e gerenciar o Product Backlog
- Priorizar o backlog
- Conhece as necessidades do cliente
- Repassar o real valor do produto para o Scrum Master e o time
- Estar disponível para o time e o Scrum Master
- Aceitar ou rejeitar as entregas

- Bom relacionamento
- Conhecer o negócio
- Habilidade de criar visão do produto
- Comunicativo
- Facilitador
- Comprometido
- Habilidade em negociação
- Proativo
- Decisivo

### Scrum Master

- Facilitador do projeto
- O Scrum Master remove impedimentos do time scrum, garante que o time tem tudo o que precisa para realizar as tarefas.
- Gerencia as cerimônias e regras da metodologia Scrum.
- Auxilia o P.O. com o **Product Backlog**.

- Remover Impedimentos do time (Não as tarefas do time)
- Garantir o uso do Scrum (cerimônias)
- Assegura que a equipe respeita e siga os valores e princípios do Scrum
- Protege o time de interferências externas, ou seja remove impedimentos
- Auxilia o PO com o Product Backlog
- Protege a equipe assegurando que ela se comprometa apenas com o que ela é capaz de entregar
- Organize e garante a realização das cerimônias
- Garante que os objetivos do PO, sejam realizados

- Conhece bem o Scrum
- Responsável
- Comunicativo
- Humilde
- Facilitador
- Comprometido
- Influente
- Observador
- Acessível
- Motivador
- Decisivo

## Time Scrum - Desenvolve o Projeto

- Os desenvolvedores do projeto definem as metas de cada entrega (cada sprint)
- O time tem como principal característica ser auto gerenciável e multi funcional, ou seja, dentro do mesmo time podemos ter programadores, testers, designer e outros profissionais.
- A responsabilidade das entregas é de todo o time, o trabalho em equipe é muito importante para que saia como planejado.
- O time estabelece o que consegue entregar, seus limites e o que pode entregar dentro do prazo.

- Define meta das interações
- Autogestão
- Multifuncional
- Desenvolve produto com qualidade e valor
- Comprometido com o trabalho, com a sprint e com as entregas
- Responsabilidade Compartilhada
- Auto-organizável

- Proativo
- Técnico
- Trabalho em equipe



Não existe hierarquia dentro do Ágil, todos são importantes para o **Produto**, todos precisam ser **proativos** e **trabalhar em equipe**.

## Rituais e Cerimônias

### Sprint Planning

- Nessa reunião o time decide quais demandas fazem parte da próxima Sprint com base na priorização do **Product Backlog**, quebra as demandas em

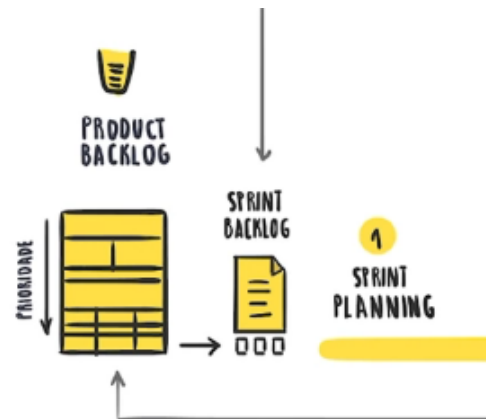
pequenas tarefas e faz uma estimativa de esforço para cada uma delas.

- Negociação do time com PO pra ver o que sai da **Sprint**
- O **resultado** desse evento é a lista de tarefas da sprint que chamamos de **Sprint Backlog**.

**1) Sprint Planning (Planejamento da Sprint)**

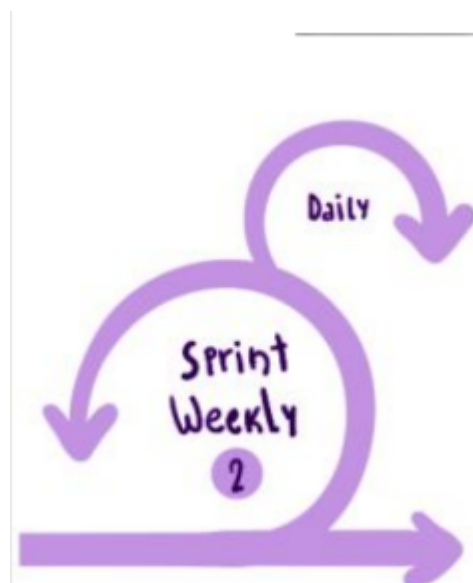
- PO Com Product Backlog
- Priorização do Product Backlog
- Identificação dos itens para o Sprint Backlog
- Quebra dos itens em pequenas tarefas
- Estimativa

**Resultado:** Sprint Backlog



## Sprint

- A sprint é o período de execução das tarefas, normalmente dura entre 1 a 4 semanas.
- O ideal é que todas as sprints tenham o mesmo tempo de duração (mantém o padrão de semanas desde a primeira Sprint).
- Todo o time fica focado no **Sprint Backlog** que foi definido na Planning.



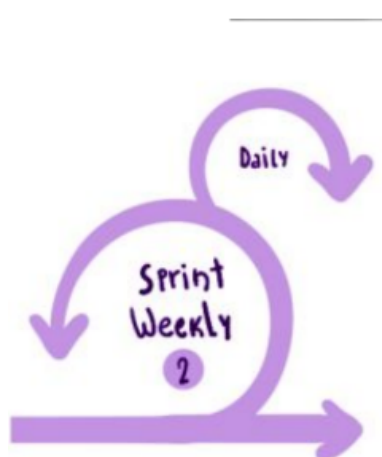


### Daily Meeting (reuniões diárias)

- Reunião diária ou stand-up meeting de alinhamento do time, deve durar no máximo 15min e ser bem objetiva, para que não se prolongue
- Sincronização da equipe nas últimas 24 horas e o plano das próximas 24 horas
- No máximo 15 minutos
- Scrum Master remove impedimento e faz a facilitação.

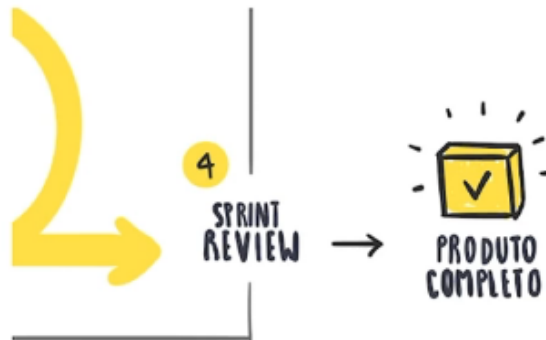
Cada colaborador do time responde a 3 perguntas:

1. O que fiz desde a última daily?
2. O que vou fazer até a próxima daily?
3. Tenho algum impeditivo?



### Sprint Review

- Reunião feita no final da sprint para apresentação dos resultados para o **P.O/cliente** validar e dar **feedback** para a equipe.
- Avaliação do **PO** se atingiu a meta (aceita da entrega)
- Feedback do **PO** para equipe
- Atualização do **Product Backlog** pelo **PO**



## Retrospectiva

- Retrospectiva também conhecida como **Retrô**, é o momento que o time analisa e última sprint, o que foi bom e o que foi ruim e precisa melhorar.
- Feita ao final da **Sprint Review**
- Levantar pontos de melhoria, pontos positivos, reconhecimentos
- O **resultado** desse evento é um Plano de Ação de melhorias para a próxima sprint.
- Essa é uma das maiores vantagens de metodologias ágeis, a otimização do trabalho é feita durante o projeto, não somente no final, possibilitando a melhoria contínua do time.

### Perguntas a serem respondidas na Retrospectiva

1. O que foi que acertamos?
2. O que erramos?
3. O que e como pode melhorar



Por isso é necessário um modelo incremental, para o time estar se adaptando e corrigindo o produto, de acordo com a necessidade de entrega do Cliente.

- **Entregamos o valor sempre ao final da Sprint para o cliente.**

## Artefatos

- Projetado para maximizar a transparência das informações.

## Product Backlog

- Lista ordenada de tudo que é necessário para o produto.
- A criação, ordenação por prioridade e gerenciamento dessa lista é feita pelo Product Owner.

### Sprint Backlog

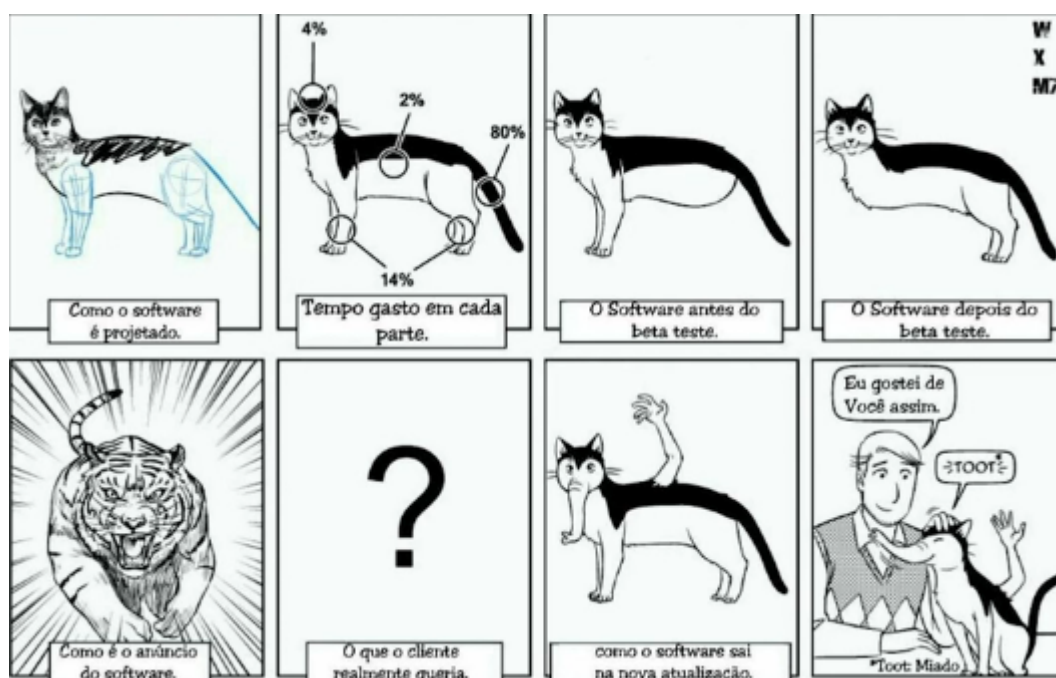
- Lista de itens selecionados do Product Backlog para a Sprint.
- Representa o período de 1 a 4 semanas (a ser definido pelo time)
- A definição dessa lista é feita somente na Sprint Planning e não pode ser alterada durante a execução da Sprint.

### Incremento

- Resultado da entrega que agrega funcionalidade e valor para o cliente.
- Como entregamos sempre algo de valor, o produto sempre terá um incremento.

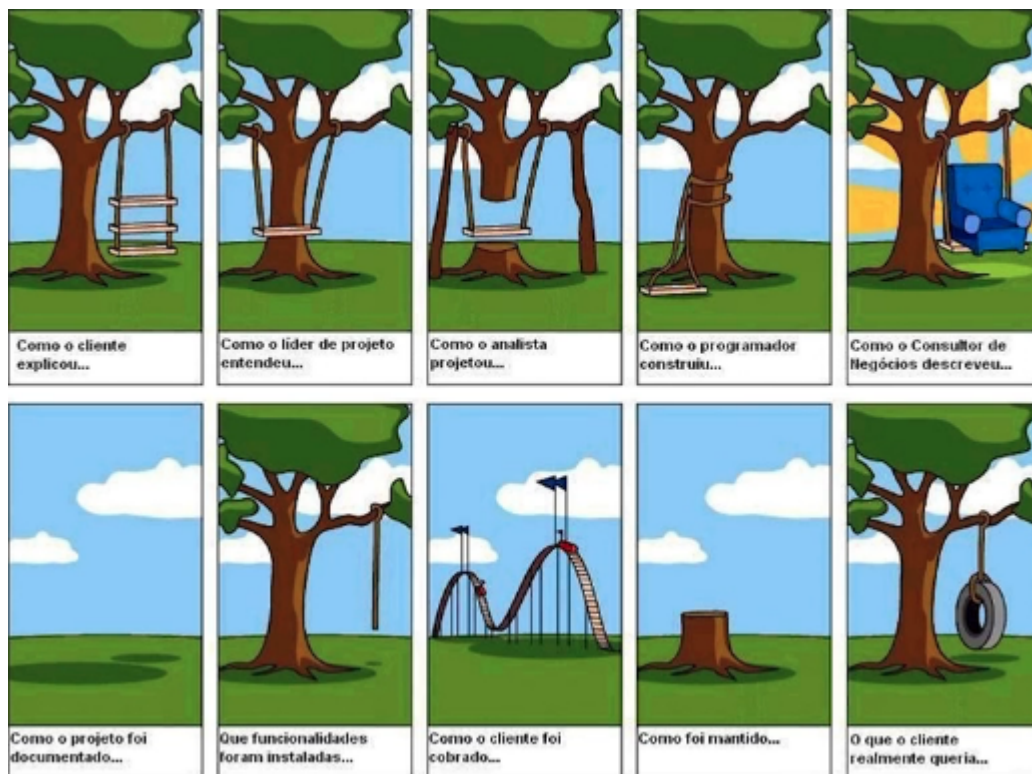
## Guia para Desenvolvimento de Software ilustrado

- No Ágil é possível errar, mas o erro é rápido devido às Sprints e o cliente tem sempre valor entregue durante esse processo.



## Scrum na Prática I

- Entregar sempre valor e descobrir o que o cliente quer.



## Fluxo do Produto

### Product Backlog

- Se inicia com Product Backlog, o PO entende com cliente o que gera valor pra ele, ele coloca no PB as funcionalidades e o que produto deve ter.
- É um Produto vivo, pode tirar e colocar funcionalidades dentro do Product Backlog, sempre vai ser atualizado
- Ele também detalha as funcionalidades de acordo com a prioridade
- O PO tem que priorizar o que gera valor para o cliente

### Sprint Backlog

- Decide o tempo para finalizar a Sprint, normalmente 2 semanas, as outras Sprints seguirão esse tempo.
- PO define com o time as prioridades e vão confirmar o que vai entrar na Sprint dentro das prioridades.
- É uma reunião em que o time diz as competências de entrega e é acordado, e o PO dá as prioridades
- O papel do Scrum Master é facilitar essa cerimônia e ver qual é o momento do time e do PO falar

## Sprint Planning

- Daily Meetings pra dizer as dificuldades, o que está sendo feito etc.
- Dura no máximo 15 minutos.

## Sprint Review

- O time demonstra as funcionalidades e o que foi implementado
- Mostrar a entrega pro Product Owner.
- Caso haja alguma coisa rejeitada pelo PO, vai para outra **Sprint**
- O PO pode e deve ser consultado durante a Sprint caso haja dúvida.
- O Scrum Master tira qualquer impedimento que ocorrer.
- A partir da retrospectiva é possível ver o que vai ser entregue na próxima Sprint, e associar um peso a cada funcionalidade, tendo uma métrica para a entrega do time.

## Retrospectiva

- Ver o que funcionou e o que não funcionou durante a Sprint e implementação para próxima Sprint.

→ Papel do PO durante a Sprint é atualizar sempre o Backlog, detalhar outros itens e deixar bem claro cada item para o time e para o produto em geral.

## Waterfall x Scrum

	Waterfall	Scrum
<b>Abordagem</b>	Escopo Fechado, Cronograma estimado	Escopo estimado, cronograma fechado
<b>Envolvimento Cliente</b>	No início e no fim	Colaboração Frequente
<b>Escopo</b>	Construir tudo do escopo	Construir de acordo com a necessidade do cliente, por prioridade
<b>Desenvolvimento</b>	Linear por fases	Iterativo, baseado em aprendizagem
<b>Entregas</b>	Tudo no final	Frequente, pequenos incrementos
<b>Custo das mudanças</b>	Alto	Baixo
<b>Comunicação do Time</b>	Hierarquia	Contínuo e sem hierarquia



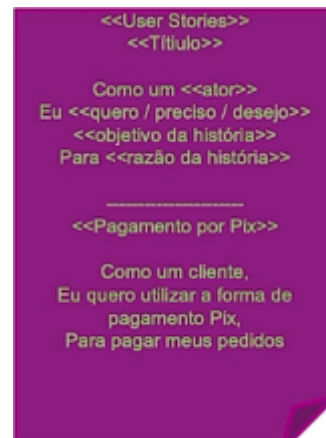
A grande assertividade do Scrum é que ele traz o cliente pra dentro do processo e faz vê como o produto está sendo feito.

## Refinamento do Backlog

- Garantir o entendimento de todos ordenados pela **prioridade** para o Cliente.
- Implementamos para diminuir o risco de implementar a coisa errada
- Para diminuir o desperdício de tempo e esforço, retrabalho e itens de maior valor que não são ordenados.
- PO sempre deve priorizar o Backlog

## User Stories

- Escrita do User Stories
- INVEST
  - Independente
  - Negociável
  - Valorosa
  - Estimável
  - Pequena
  - Testável



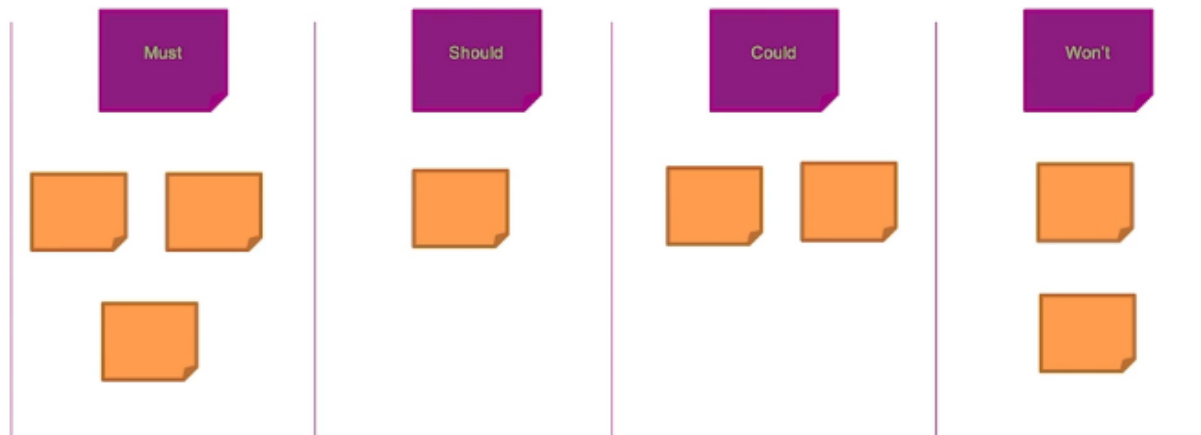
## DoD (Definition of Done)

- Definição do Pronto
- Ter a clareza e descrição do que é o pronto
- **Exemplo de DoD**
  - Codificado
  - Passado por testes

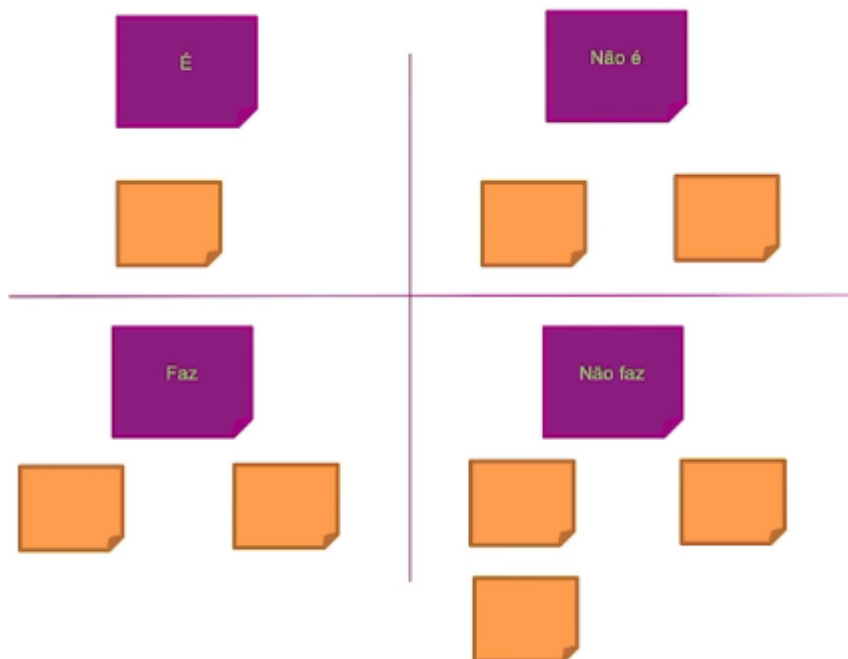
- Integrado com o Sistema X
- Documentado

### Método MOSCOW

- Ajuda o PO a definir a prioridades



### É, não é, faz e não faz (produto)



### Métricas dentro da Sprint



Comunicação não é o que você diz, é o que os outros entendem, seja claro.

## Kanban

### O que é?

- O sistema Kanban foi desenvolvido por Taiichi Ohno, engenheiro industrial e empresário, para a Toyota no Japão, em 1940.
- O intuito era criar um sistema de planejamento simples, a fim de controlar e gerenciar o trabalho e o estoque em todas as fases da produção de forma otimizada.
- Basicamente, ele começou a usar o sistema para movimentar e fornecer itens sem que houvesse o abastecimento de materiais sem solicitação.
- Os cartões de informação eram utilizados para controlar a produção de acordo com a demanda.
- O **Kanban** pode entrar no **Scrum**

### Princípios

- Começar com o que você já faz
- Aceitar a busca por uma mudança evolutiva e incremental
- Respeitar os processos, funções e responsabilidade atuais



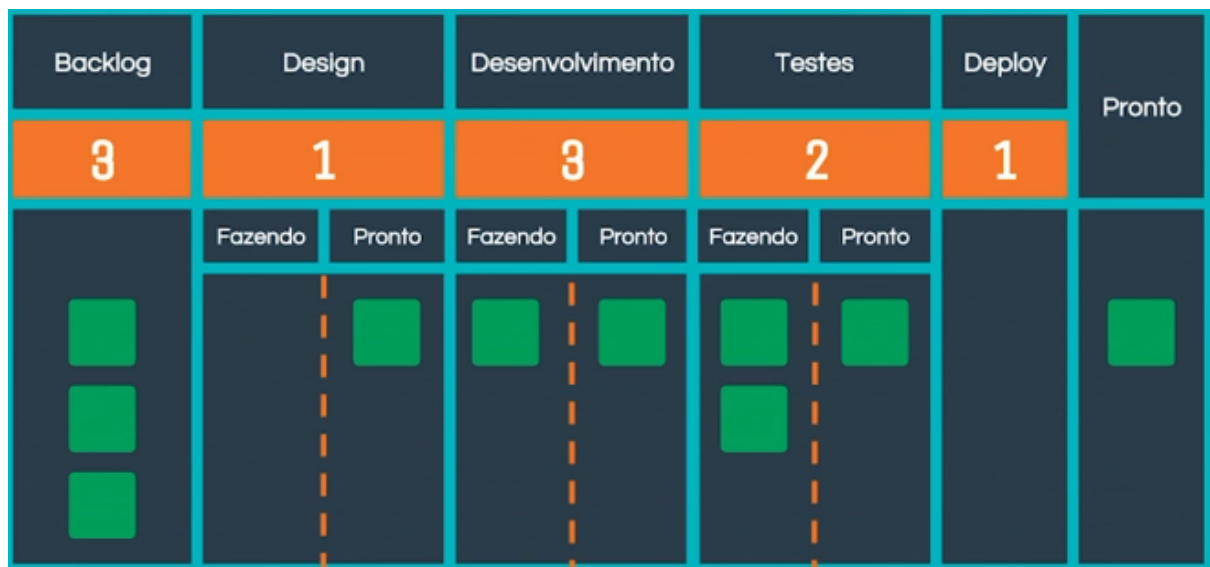
- Encorajar Atos de liderança em todos os níveis

💬 Pare de começar e comece a terminar.

## Serve para...

- Gerenciar fluxo de trabalho
- Equilibrar processos
- Limitar quantidade de trabalho

## Kanban no Scrum



## Como usar o Kanban?

- Simples e fácil de ser aplicado
- Clareza e colaboração
- Reduz desperdício
- Gestão visual
- Evita ociosidade

## Ferramentas

- Você pode utilizar softwares que simulam o que é feito em offline (clique aqui e conheça alguns deles), mas se prefere os métodos tradicionais você vai

precisar de uma lousa e de cartões coloridos como os post-its.

- Cada um deles representará uma tarefa do seu processo.
- Trello, Slack (mais comum) e até discord.

## Guias

Divida o seu quadro em três colunas conforme abaixo:

### To do (a fazer)

- Coloque os cartões com as tarefas que você precisa fazer nesta lista;

### Doing (fazendo)

- Aqui, coloque os post-it com as tarefas que você está fazendo no momento;

### Done (feito)

- Nessa lista, você colocará as tarefas que já foram concluídas.

## Conclusão

- Então, se você já começou a fazer uma tarefa que estava na lista “to do” é só arrastá-la para a coluna “doing”.
- E se você já finalizou um trabalho que estava na coluna “doing”, arraste-a para a coluna “done”.



Kanban é muito utilizado na criação de Rotina para crianças e determinar prioridades em uma tarefa, ele dá muita visibilidade no que se tem que fazer.