

# Metodologias ágeis

**Disciplina:** Análise e Projetos em Sistemas



## Conteúdos:

Metodologias ágeis de desenvolvimento.

## Habilidade(s):

- Conhecer as metodologias ágeis mais utilizadas no gerenciamento e desenvolvimento de projetos;
- Aplicar as metodologias ágeis em projetos de desenvolvimento de *softwares*.

# Bloco 1

---

## Responda rápido!

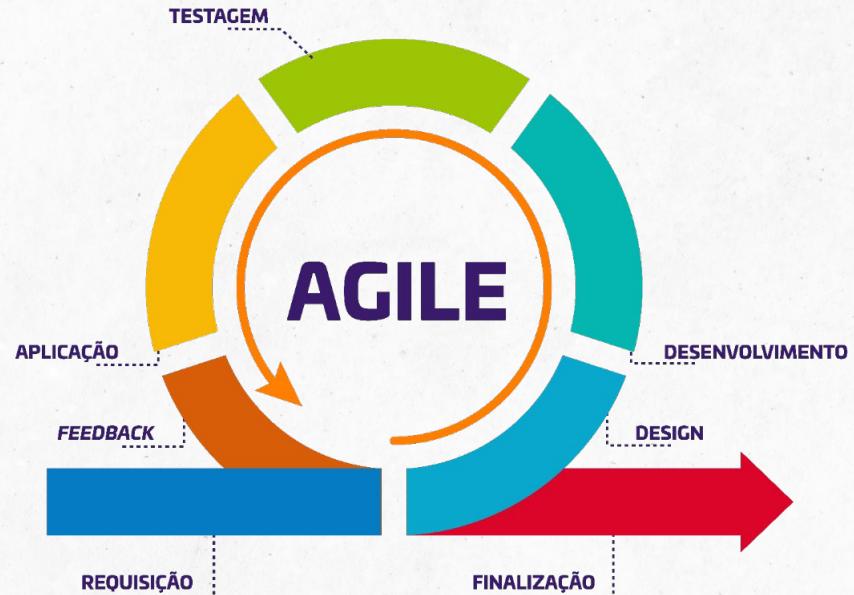
Quando você ouve falar no tema “metodologias ágeis”, qual é a primeira coisa que vem à sua mente?



# Metodologias ágeis

Será que existe uma maneira rápida e simples de cumprir cronogramas em um projeto?

Sim, existe! Com as metodologias ágeis, isso é possível. Elas buscam promover a **eficiência**, a **adaptabilidade** e a **colaboração** na criação de modelos, tendo como **foco a entrega contínua** de valor para os clientes.



## Você sabia?

A stylized illustration of a person with curly hair, wearing a white shirt. They are holding a large orange magnifying glass up to their eye, looking intently. Above their head are two orange question marks, suggesting inquiry or discovery.

Em 2001, o **Manifesto Ágil** foi elaborado por 17 desenvolvedores. Ele determina os valores e os princípios da metodologia ágil para o desenvolvimento de *softwares*.

A página oficial do manifesto está disponível em várias línguas, incluindo o português:  
[agilemanifesto.org/iso/ptbr/manifesto.html](http://agilemanifesto.org/iso/ptbr/manifesto.html)



# Modelagem de sistemas

VS

Metodologias ágeis

Scrum

Kanban

*Extreme programming*

*Lean software development*

*Scaled agile framework*

Metodologias tradicionais

Cascata

Prototipação

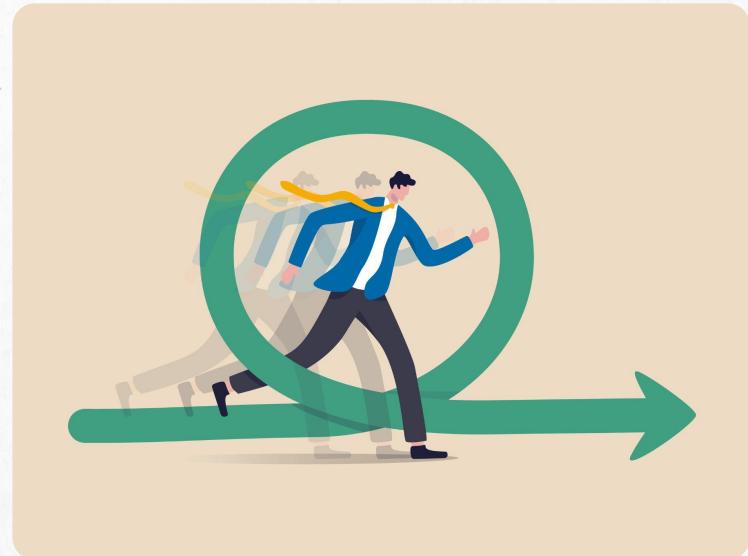
Espiral

# Há muitas diferenças?

	Metodologias ágeis	Metodologias tradicionais
Abordagem	Iterativas e incrementais, são divididas em ciclos curtos e repetitivos, chamados de iterações ou <i>sprints</i> .	Seguem uma abordagem sequencial e linear.
Flexibilidade e mudanças	Altamente adaptáveis às mudanças nos requisitos, visto que é produzida em ciclos.	As especificações são definidas no início do projeto e, geralmente, as mudanças são complexas e custosas.
Envolvimento do cliente	Incentivam um envolvimento contínuo do cliente durante todo o processo de desenvolvimento.	O cliente se envolve no início do projeto para definir os requisitos. Depois, ele só vê o produto final.
Entrega de valor	O produto é entregue em momentos planejados do <i>sprint</i> e de forma incremental.	O produto final é entregue somente após a conclusão do projeto.
Gestão de riscos	Tratam os riscos de forma mais flexível, com uma abordagem adaptativa.	Têm uma abordagem mais rígida para a gestão de riscos, com a análise e o planejamento antecipado.

# Os benefícios das metodologias ágeis

- Mais adaptável se tratando de mudanças;
- Entrega contínua;
- Maior envolvimento dos clientes;
- Comunicação mais aberta e eficiente;
- Maior qualidade do produto;
- Possibilidade de melhoria contínua;
- Maior autonomia para as equipes;
- Riscos menores;
- Maior transparência.



# Os desafios das metodologias ágeis

- Requer mudanças na cultura empresarial;
- Exige comunicação e coordenação;
- São difíceis de escalar para projetos grandes;
- Requer boa gestão para lidar com mudanças;
- Possuem prazos mais curtos e expectativas mais altas;
- Capacidade de autogerenciamento;
- Precisa de engajamento dos *stakeholders*.



# Bloco 2

---

# Há muitas diferenças?

	Metodologias ágeis	Metodologias tradicionais
Abordagem	Iterativas e incrementais, são divididas em ciclos curtos e repetitivos, chamados de iterações ou <i>sprints</i> .	Seguem uma abordagem sequencial e linear.
Flexibilidade e mudanças	Altamente adaptáveis às mudanças nos requisitos, visto que é produzida em ciclos.	As especificações são definidas no início do projeto e, geralmente, as mudanças são complexas e custosas.
Envolvimento do cliente	Incentivam um envolvimento contínuo do cliente durante todo o processo de desenvolvimento.	O cliente se envolve no início do projeto para definir os requisitos. Depois, ele só vê o produto final.
Entrega de valor	O produto é entregue em momentos planejados do <i>sprint</i> e de forma incremental.	O produto final é entregue somente após a conclusão do projeto.
Gestão de riscos	Tratam os riscos de forma mais flexível, com uma abordagem adaptativa.	Têm uma abordagem mais rígida para a gestão de riscos, com a análise e o planejamento antecipado.

# Metodologia Scrum

A metodologia Scrum foi criada na década de 1990 por Ken Schwaber e Jeff Sutherland, dois dos 17 signatários do Manifesto Ágil. Por isso, essa metodologia é baseada nos princípios e valores ágeis, como:

- flexibilidade;
- colaboração;
- comunicação;
- entrega contínua de valor.

No Scrum, o trabalho no projeto **se divide em iterações**, um processo de repetições chamadas de **sprints**. Geralmente, possui duração de duas a quatro semanas.



# Elementos Scrum

Nos projetos e durante cada *sprint*, a **equipe de desenvolvimento se divide em papéis**, cada um com um objetivo na produção. As informações importantes que ajudam a equipe a definir o produto e o trabalho para construí-lo recebem o nome de **artefato**. Além disso, as constantes reuniões e os momentos de planejamentos são chamados de **eventos**.

## Papéis

- *Product owner;*
- *Scrum master;*
- Equipe de desenvolvimento.

## Artefatos

- *Product backlog;*
- *Sprint backlog;*
- Incremento.

## Eventos

- Planejamento de *sprint*;
- *Daily Scrum*;
- Revisão de *sprint*;
- Retrospectiva de *sprint*.

# Por que usar Scrum?

A metodologia fornece uma **estrutura flexível e adaptativa** para lidar com as mudanças constantes que ocorrem em projetos de desenvolvimento.

Ao permitir entregas frequentes e uma interação contínua com o cliente, o Scrum torna-se uma poderosa abordagem para **aumentar a eficiência, a transparência e a qualidade do desenvolvimento de softwares**.





## Estudo de caso

**Formem grupos de quatro pessoas e analisem o seguinte cenário:** imaginem-se como parte de uma equipe de desenvolvimento de *software* que trabalha para um grande banco. Esse banco está **realizando mudanças no seu aplicativo de investimento para dispositivos móveis**. O novo aplicativo ajudará os usuários a organizarem os seus investimentos, acompanhando a sua progressão econômica.

Agora, apliquem a metodologia Scrum para resolver essa situação. **Escolham quem vai assumir cada papel, quais atividades cada membro deverá desempenhar nesse contexto e qual será a produção de cada pessoa durante o desenvolvimento do aplicativo.**

# Bloco 3

---

# As funções de Antônio



Antônio trabalha na equipe de desenvolvimento de uma empresa. A metodologia Scrum é utilizada pelos profissionais e Antônio é a pessoa responsável por facilitar o processo de desenvolvimento, garantindo que a equipe siga as práticas e princípios ágeis. Pode-se dizer, portanto, que a função de Antônio é:

a

*Product owner.*

b

*Scrum master.*

c

Desenvolvedor sênior.

# As funções de Antônio



Antônio trabalha na equipe de desenvolvimento de uma empresa. A metodologia Scrum é utilizada pelos profissionais e Antônio é a pessoa responsável por facilitar o processo de desenvolvimento, garantindo que a equipe siga as práticas e princípios ágeis. Pode-se dizer, portanto, que a função de Antônio é:

a

*Product owner.*

b

*Scrum master.*

c

Desenvolvedor sênior.

# Kanban

Kanban é uma metodologia ágil baseada na demanda do cliente. Ela se concentra na **visualização do fluxo de trabalho** e no **gerenciamento eficiente das tarefas**.

Essa abordagem teve origem na década de 1940, graças ao Sistema Toyota de Produção (TPS). Em 2007, foi adaptada para o desenvolvimento de *software* e, desde então, inspira outras metodologias.



# Características do Kanban

O Kanban, assim como a sua aplicação em projetos, pode trazer uma série de benefícios. No entanto, o seu uso costuma se relacionar com algumas características, tais quais:

- uma visão clara do **status** de cada tarefa graças ao quadro Kanban;
- a utilização dos **limites de trabalho em progresso (WIP)** para evitar sobrecargas;
- iterações rápidas e com foco na **entrega constante** de projetos;
- busca pela identificação de problemas e gargalos para gerar **melhoria contínua**;
- oferecer um modelo de trabalho mais **flexível**.



# Quadro de atividades

Historia do Usuário	A fazer	Em Progresso	Testes	Feito

## Hora de praticar!

**Formem grupos de quatro pessoas e analisem o seguinte caso:** um grupo de desenvolvedores trabalha para uma empresa eletrônicos, a Click, e, no momento, estão focando nas funcionalidades da loja *on-line*. Porém, ao olhar o **quadro do Kanban da equipe, podemos ver que está um tanto desorganizado.**

Sabendo que o desenvolvimento já se encontra em progresso, observe algumas partes das etapas e as encaixe em **backlog, a fazer, em progresso, testes e feito.**

Após construir a tabela, veja quais são os itens espalhados.



# Itens do quadro Kanban da equipe

- 1 Pensar em uma maneira de visualizar o histórico de pedidos.
- 2 Definir os requisitos para a funcionalidade de adicionar itens ao carrinho.
- 3 Implementar uma página de pagamento.
- 4 Identificar e corrigir *bugs* na visualização do histórico de pedidos.
- 5 Apresentar a nova funcionalidade do carrinho para o *product owner*.
- 6 Analisar o recebimento das notificações pelos clientes.
- 7 Elaborar uma forma de enviar notificações de promoções.
- 8 Codificar a funcionalidade de adicionar itens ao carrinho.
- 9 Notificar os clientes da possibilidade de pagamento em vários cartões.

# Bloco 4

---

## Tente responder

Você consegue se ver utilizando a metodologia Scrum  
em quais cenários? E a Kanban?



## **Extreme Programming (XP)**

*Extreme Programming* é uma metodologia ágil criada por Kent Beck, um dos 17 signatários do manifesto ágil, no final da década de 1990.

Ela **enfatiza a colaboração, a comunicação e a qualidade do código** e se tornou uma das abordagens ágeis mais populares nos dias atuais. Essas características a tornam **adequada para projetos nos quais os requisitos podem mudar com frequência**.

# Práticas em XP

Para garantir a aplicação da abordagem XP, existem algumas práticas que podem ajudar bastante. Algumas delas são:

desenvolvimento orientado a testes (TDD);

programação em pares;

integração contínua;

pequenas *releases*;

clientes presentes.

# Vamos praticar?

Dividam-se em grupos de quatro pessoas para realizar a atividade!

## Primeiro momento

10 minutos

Escolham uma das práticas vista na página anterior. Depois, pesquisem e discutam em quais cenários poderiam aplicá-la, abordando, também, os seus benefícios. As práticas são:

- TDD;
- programação em pares;
- integração contínua;
- pequenas *releases*;
- cliente presente.

## Segundo momento

20 minutos

Analisem o caso da próxima página e apliquem o modelo escolhido anteriormente, construindo uma funcionalidade específica.



## Estudo de caso



A EcoPet é uma plataforma *on-line* que conecta abrigos de animais, protetores independentes e potenciais tutores de animais de estimação. A plataforma tem como objetivo facilitar o processo de adoção responsável. Ela permite que os abrigos e os protetores independentes cadastrem animais resgatados, adicionando informações detalhadas sobre eles para facilitar a adoção.

- **TDD:** escrever testes para a funcionalidade “cadastrar um animal resgatado”;
- **Programação em pares:** implementar a funcionalidade “adotar um animal”;
- **Integração contínua:** configure um ambiente de integração contínua para a funcionalidade “visualizar detalhes do animal”;
- **Pequenas releases:** planeje e implemente a funcionalidade “encontrar abrigos próximos” em pequenas incrementações;
- **Cliente presente:** atue como se fosse um representante do cliente e colabore na implementação da funcionalidade “enviar mensagens para o abrigo”.

# Bloco 5

---

# Práticas em XP

Para garantir a aplicação da abordagem XP, existem algumas práticas que podem ajudar bastante. Algumas delas são:

desenvolvimento orientado a testes (TDD);

programação em pares;

integração contínua;

pequenas *releases*;

clientes presentes.

## ***Lean Software Development (LSD)***

A *Lean Software Development* é uma abordagem de desenvolvimento baseada nos princípios e práticas do *Lean Manufacturing*, inspirada no sistema Toyota de produção (TPS), o mesmo que serviu de base para o Kanban. A *Lean Manufacturing*, no entanto, se desenvolveu uma década depois, em 1950 e 1960.

O objetivo da LSD é:

- **maximizar o valor entregue ao cliente;**
- **eliminar desperdícios;**
- **reduzir o tempo de entrega;**
- **manter um alto padrão de qualidade.**

# Os princípios da metodologia LSD

Para atingir os seus objetivos, a abordagem LSD se sustenta em alguns princípios ágeis e específicos do seu modelo.

- Priorizar a **entrega rápida e frequente** de incrementos de *software* funcional;
- Identificar e **eliminar todas as atividades que não agregam valor** ao produto final;
- Buscar **aprender com os sucessos e as falhas** ao longo do projeto;
- Dar **autonomia à equipe** de desenvolvimento para tomar decisões e resolver problemas;
- **Entregar o produto completo** com todas as funcionalidades necessárias para atender às necessidades do cliente.

# Ferramentas e práticas

Algumas ferramentas e práticas, incluindo outras metodologias ágeis, podem ser utilizadas na metodologia LSD. Alguns exemplos são:

## Quadro Kanban

Os quadros Kanban ajudam a visualizar e a gerenciar o fluxo de trabalho da equipe.

## Minimização do tempo de ciclo

Busca-se reduzir o tempo de ciclo entre o conceito e a entrega do *software*.

## Desenvolvimento iterativo e incremental

Consiste em trabalhar com pequenas iterações, adicionando funcionalidades em incrementos, possibilitando obter *feedbacks* mais rapidamente. Desse modo, também é possível fazer ajustes.

## Priorização do cliente

Coloca-se o cliente no centro do desenvolvimento, priorizando as suas necessidades e os requisitos estabelecidos.

??



Imagine que o seu grupo está desenvolvendo um sistema de gerenciamento de tarefas *on-line*. Foi solicitado, então, adicionar a **funcionalidade** a ser desenvolvida: a **capacidade de adicionar novas tarefas** ao sistema.

Como vocês acrescentariam essa funcionalidade, respeitando os princípios da *Lean Software Development*?

# Bloco 6

---

# Atualmente, qual é a sua maior dificuldade acerca desse tema?



Compartilhe as suas  
dúvidas!

## **Scaled Agile Framework (SAFe)**

A *Scaled Agile Framework* (SAFe) é uma metodologia ágil criada por Dean Leffingwell e combina princípios do *Agile*, do *Lean* e dos sistemas de pensamento.

Porém, diferente de outras metodologias que costumam ser utilizadas em projetos menores, ela fornece a **possibilidade de escala** para as empresas. Essa metodologia foi pensada para permitir a **colaboração eficiente entre várias equipes, entregando soluções complexas** de forma coordenada e, por isso, **possui três níveis**:

- *team level;*
- *program level;*
- *portfolio level.*

# Principais características

Existem algumas práticas que são utilizadas em diversos níveis da SAFe. A seguir, vejamos algumas delas.

## Agile Release Train (ART)

Consiste em fazer uma linha de entregas rápidas. Para isso, as equipes trabalham juntas em um cronograma sincronizado, geralmente com ciclos que vão de oito até 12 semanas.

## PI planning

É um planejamento em conjunto que ocorre no início de cada incremento de programa (PI). Nele, as equipes definem metas e prioridades para a próxima entrega.

## Inspecionar e adaptar

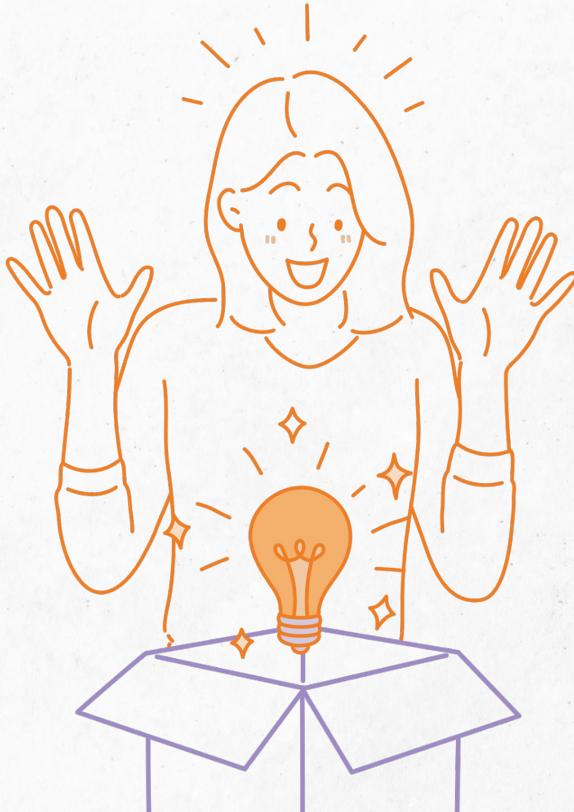
Inspecionar e adaptar (*inspect and adapt*) é uma prática que consiste em retrospectivas feitas para identificar as oportunidades de melhoria.

## Valor de negócio

O valor de negócios deve ser incremental. Ou seja, entrega-se o ou as funções pouco a pouco e com frequência, permitindo que o cliente tenha acesso a novas funcionalidades de forma mais rápida.

► □ ○ ▶ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷

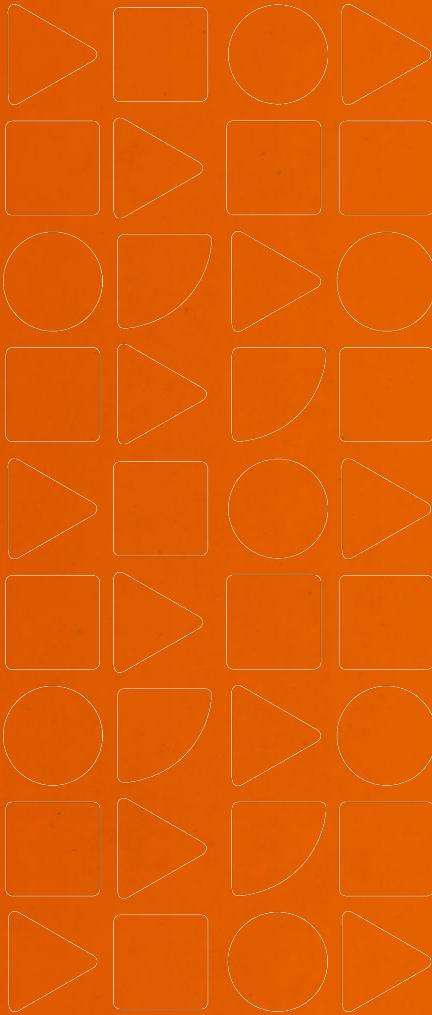
E agora? Em quais momentos você se enxerga utilizando as metodologias ágeis?



## Para saber mais...



*Post Manifesto para Desenvolvimento Ágil de Software*, escrito por Kent Beck et. al (2001). Disponível em:  
<https://agilemanifesto.org/iso/ptbr/manifesto.html>. Acesso em: 27 set. 2023.



# Referências Bibliográficas

PROZ, Apostila de Técnico em Desenvolvimento de Sistemas. 2023.

LOSNAK, Giulia. *Metodologia Ágil*: o que é. Alura, 4 jul. Disponível em: <https://www.alura.com.br/artigos/o-que-e-metodologia-agil>. Acesso em: 27 set. 2023.

KANBANIZE. *Kanban*: O que é e como Funciona. Kanbanize, s.d. Disponível em: <https://kanbanize.com/pt/recursos-kanban/primeiros-passos/o-que-e-kanban>. Acesso em: 27 set. 2023.

PIIKKILA, Jessica. *O que é o SAFe?*. Atlassian, s.d. Disponível em: <https://www.atlassian.com/br/agile/agile-at-scale/what-is-safe>. Acesso em: 27 set. 2023.

ATLASSIAN. *Scrum*: o que é, como funciona e como começar. Atlassian, s.d. Disponível em: <https://www.atlassian.com/br/agile/scrum>. Acesso em: 27 set. 2023.