

CENTRO UNIVERSITÁRIO DE BELO  
HORIZONTE - UNIBH

# Modelagem de Software

Samara Soares Leal - [samara.leal@prof.unibh.br](mailto:samara.leal@prof.unibh.br)

Otaviano Silverio de Sousa - [otaviano.sousa@prof.unibh.br](mailto:otaviano.sousa@prof.unibh.br)



# OBJETIVOS

## ANÁLISE E PROJETO ORIENTADO A OBJETOS

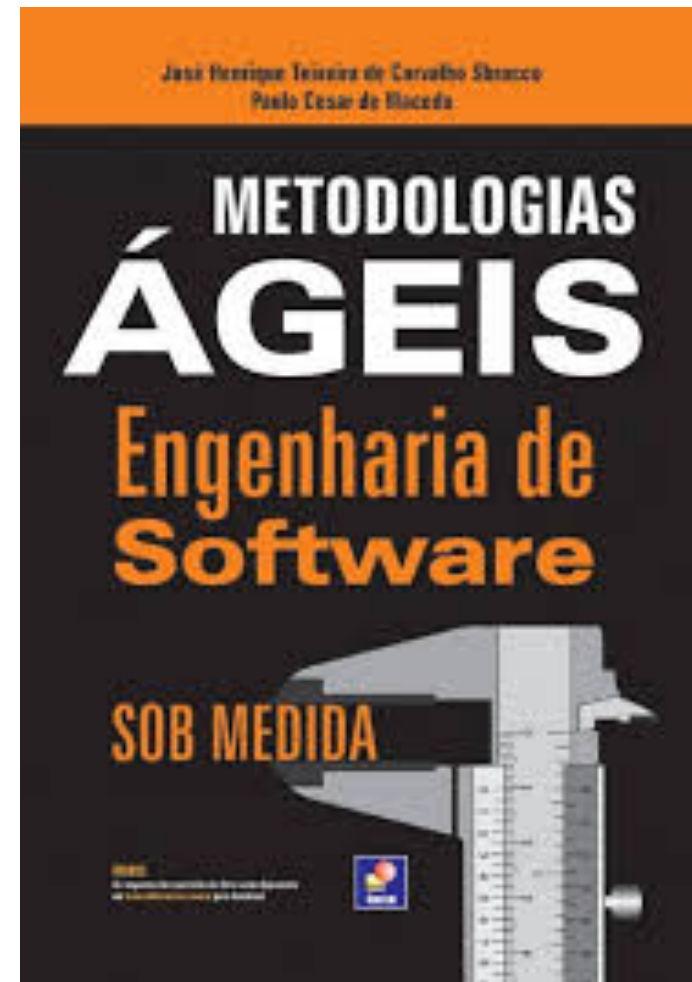
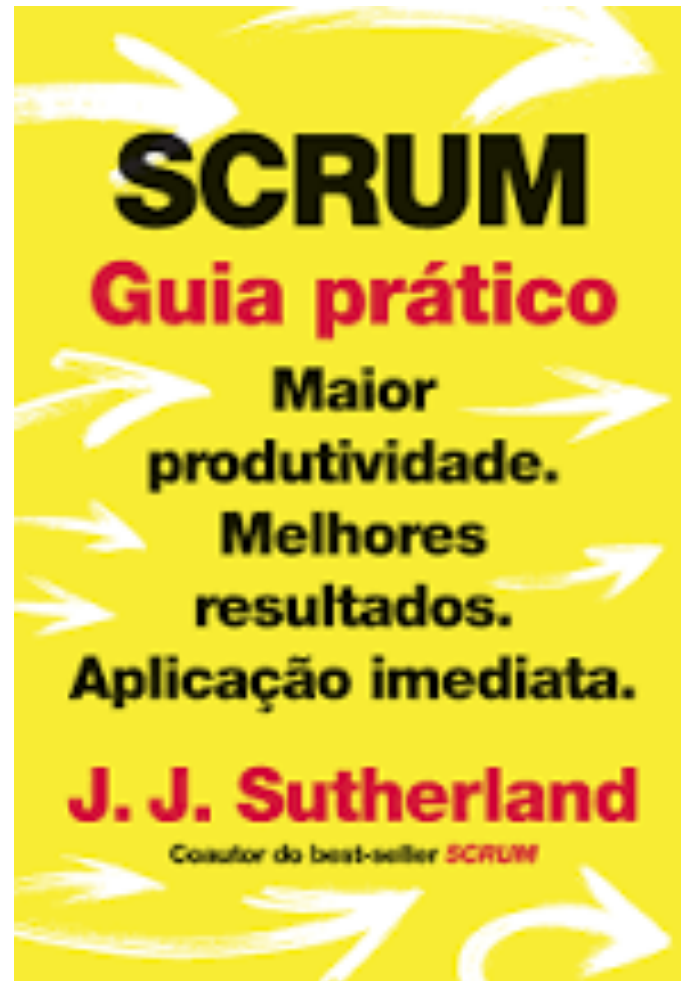
Mapeamento dos relacionamentos entre as classes do software orientado a objetos para o desenvolvimento do projeto (Apoio à codificação).

## MODELAGEM DE PROBLEMAS

Modelagem dos requisitos do software a partir de diagramas da UML e histórias de usuários (*user stories*)

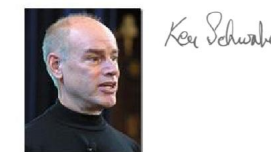
## MODELAGEM DE BANCO DE DADOS

Modelo entidade e relacionamento.  
Modelo relacional e normalização.  
Modelo lógico e físico de banco de dados



## The Scrum Guide

The Definitive Guide to Scrum:  
The Rules of the Game



July 2013

Developed and sustained by Ken Schwaber and Jeff Sutherland



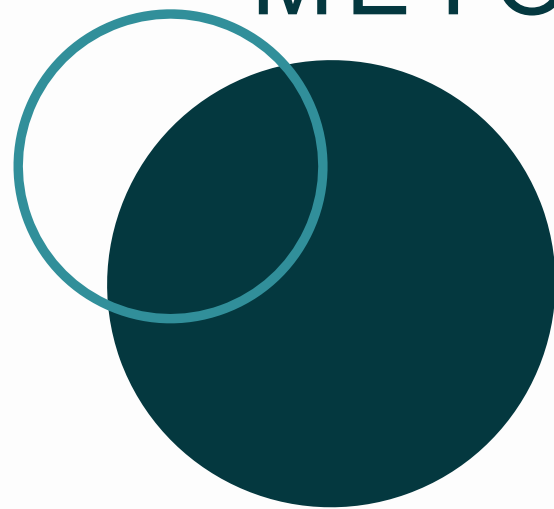
Improving the Profession of Software Development

LONGO, Hugo Estevam Romeu; SILVA, Madalena Pereira. **A Utilização de Histórias de Usuários no Levantamento de Requisitos Ágeis para o Desenvolvimento de Software.** International Journal of Knowledge Engineering and Management (IJKEM), v. 3, n. 6, p. 1-30, 2014.

SOUZA, Diogo Rodrigues de et al. **Implantação da metodologia ágil Scrum em um ambiente de desenvolvimento.** 2014.

# HISTÓRIAS DE USUÁRIOS PARA DESCREVER O QUE O SISTEMA DEVE FAZER

- METODOLOGIA ÁGIL DE DESENVOLVIMENTO DE SOFTWARE: SCRUM
  - HISTÓRIAS DE USUÁRIO



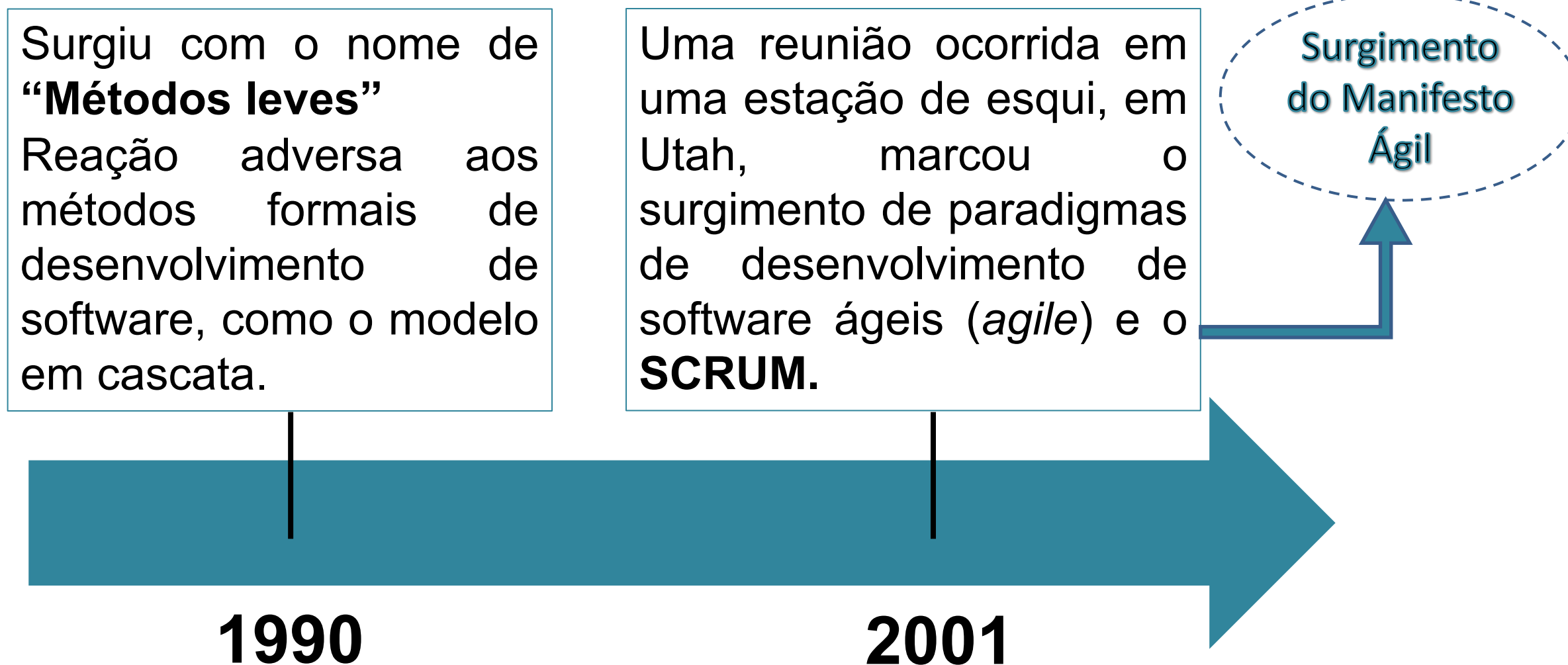
# METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

“O uso de uma metodologia é importante para apoiar o processo de **Engenharia de Software** e aumentar as chances de entregar um projeto de qualidade, no prazo contratado, com os requisitos contemplados e de acordo com o orçamento.”



# METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

## Histórico - Metodologia de Desenvolvimento de Software Ágil:





# METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

## Metodologias tradicionais **versus** Metodologias Ágeis

### Metodologias Tradicionais

**Características:** Divididas em etapas muito bem definidas e englobam atividades como Análise, Modelagem, Desenvolvimento e Testes.

Cada fase concluída gera um marco, que geralmente é algum documento, protótipo ou versão do sistema.

**Exemplo:** Modelo em **cascata**

**Foco principal:** Previsibilidade dos requisitos, projetos completamente planejados, mantendo sempre uma linha, caracterizando o processo como bastante **rigoroso**.



# METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

## Metodologias tradicionais **versus** Metodologias Ágeis

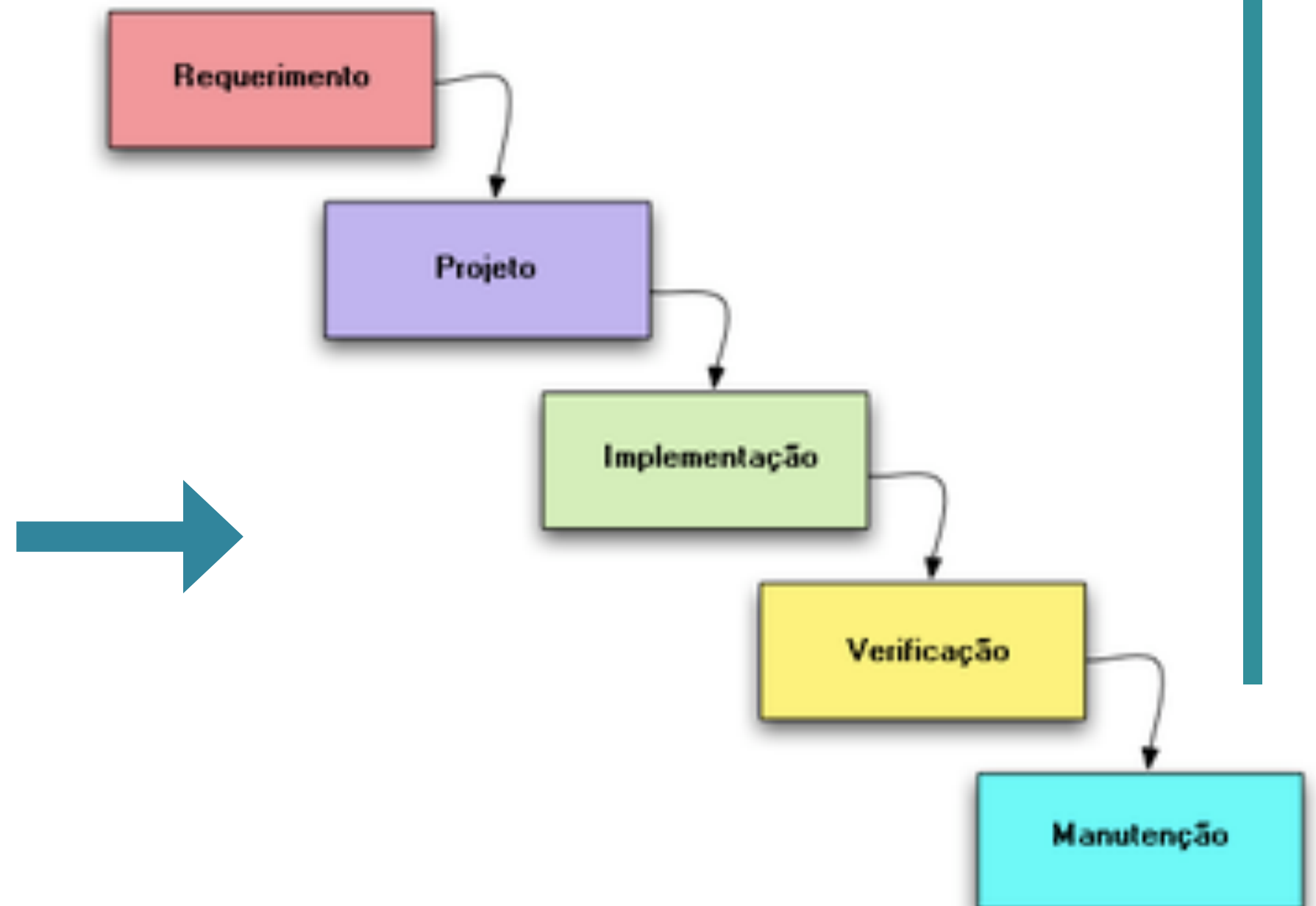
### Metodologias Tradicionais

**Características:** Divididas em etapas muito bem definidas e englobam atividades como Análise, Modelagem, Desenvolvimento e Testes.

Cada fase concluída gera um marco, que geralmente é algum documento, protótipo ou versão do sistema.

**Exemplo:** Modelo em **cascata**

**Foco principal:** Previsibilidade dos requisitos, projetos completamente planejados, mantendo sempre uma linha, caracterizando o processo como bastante **rigoroso**.



# METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

## Metodologias tradicionais **versus** Metodologias Ágeis

### Metodologias Ágeis

**Características:** Assume que mudanças no produto e documentação fazem parte do processo de desenvolvimento com foco no código e otimizados para alterações de requisitos, prezando, sempre, pela qualidade do software.

**Exemplo:** SCRUM

**Foco principal:** Pessoas, iterações curtas com entrega de produtos, leveza do processo e adaptação a mudanças nos requisitos.

# METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

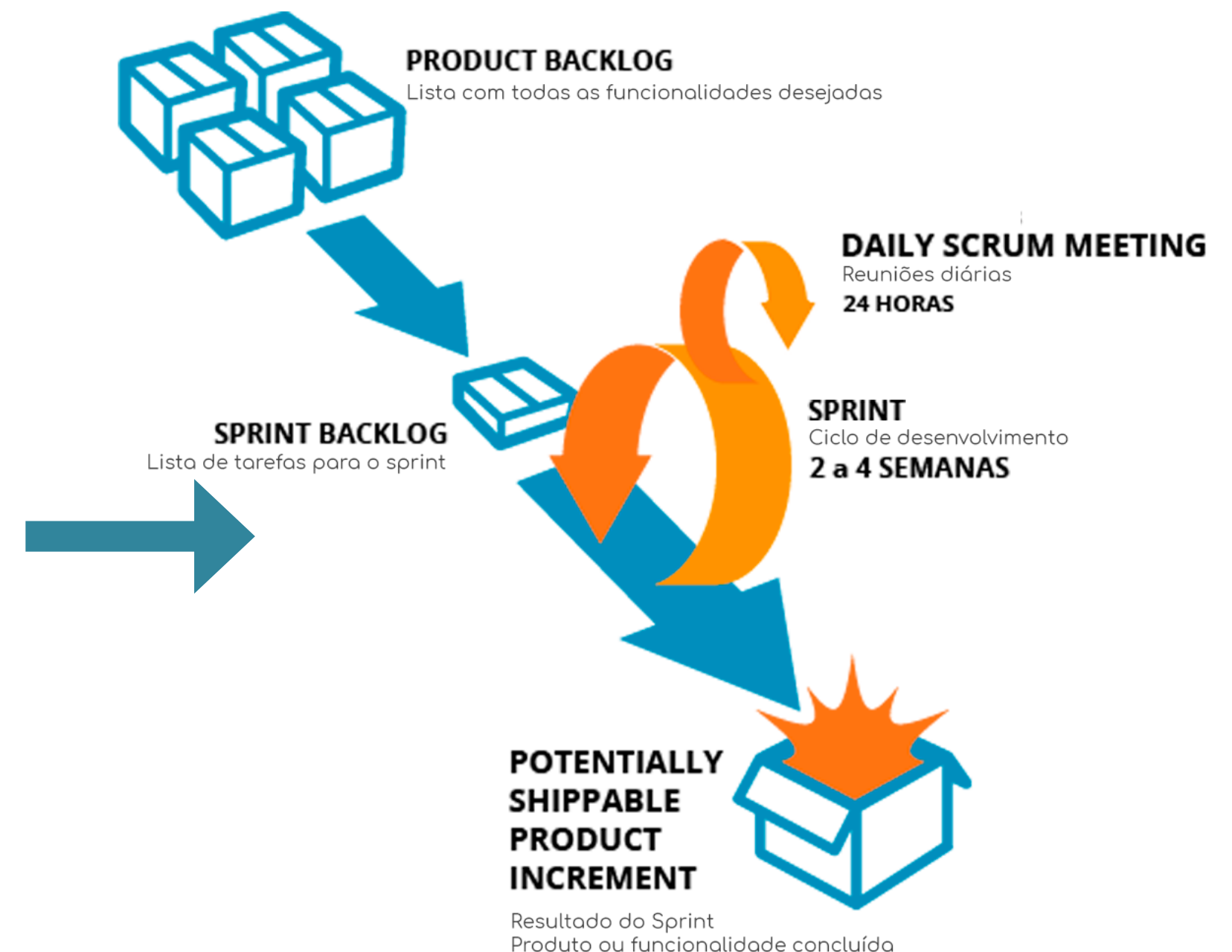
## Metodologias tradicionais **versus** Metodologias Ágeis

### Metodologias Ágeis

**Características:** Assume que mudanças no produto e documentação fazem parte do processo de desenvolvimento com foco no código e otimizados para alterações de requisitos, prezando, sempre, pela qualidade do software.

**Exemplo:** SCRUM

**Foco principal:** Pessoas, iterações curtas com entrega de produtos, leveza do processo e adaptação a mudanças nos requisitos.



# METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

## MÉTODOS ÁGEIS

Oferecem um conjunto de atividades que podem ser adotadas durante um processo de desenvolvimento de software.

- Redefinir continuamente as prioridades em um projeto de software.
- Uso de processos, ferramentas, negociações de contrato, documentos, porém de forma mais reduzida e objetiva.

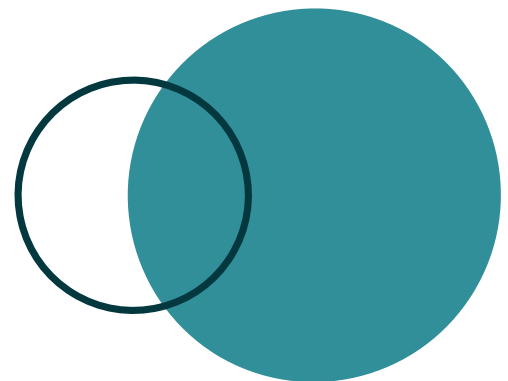
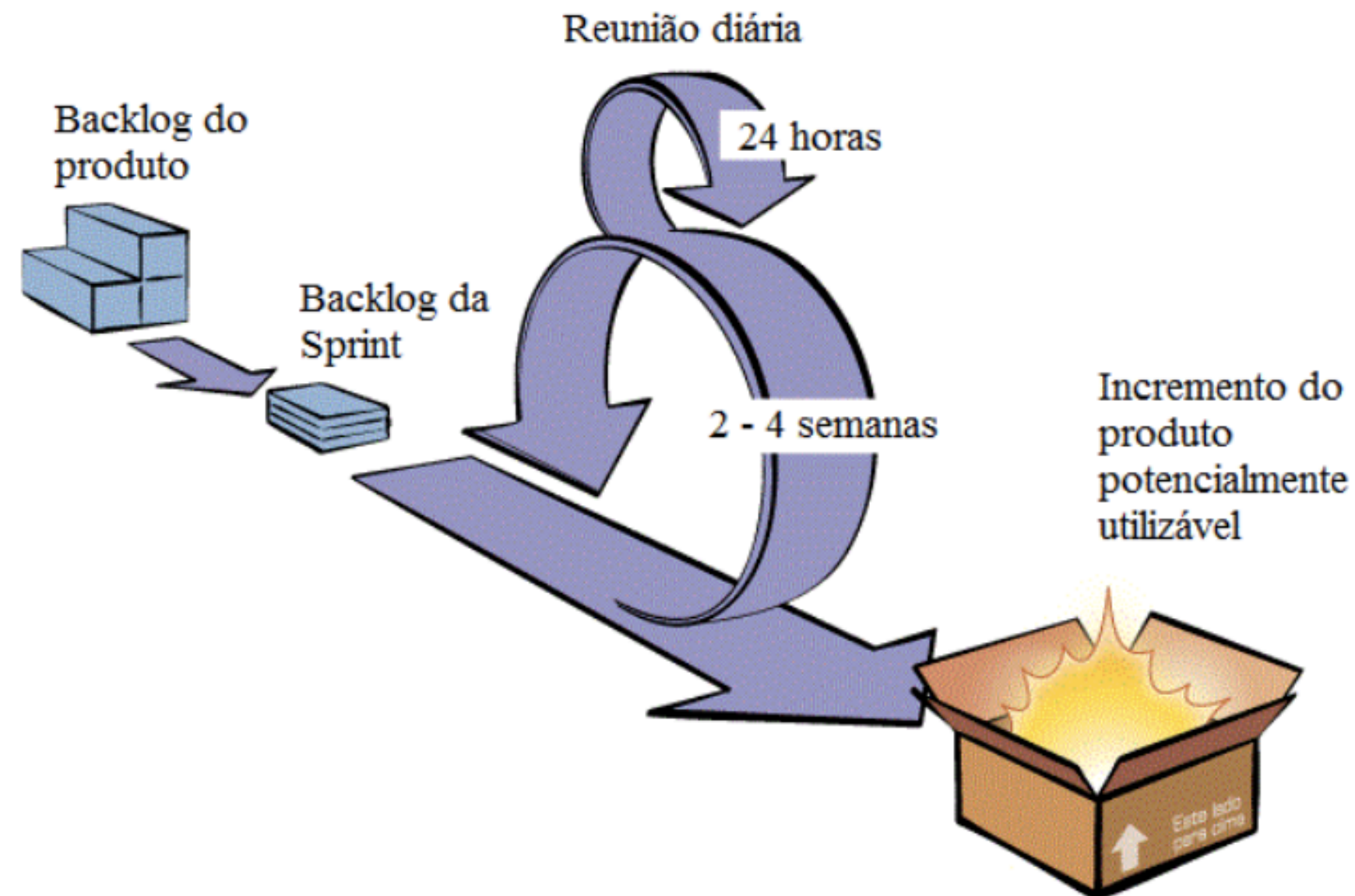
Informações complementares sobre a Aliança Ágil podem ser obtidas em <http://www.agilealliance.org/>

**Obs.:** O documento com os 12 princípios (comentados) do manifesto ágil está no *ulife*.

# MÉTODOS ÁGEIS

## METODOLOGIA SCRUM

O **SCRUM** utiliza um esqueleto de processo iterativo e incremental para aperfeiçoar a previsibilidade e melhorar o controle de riscos.



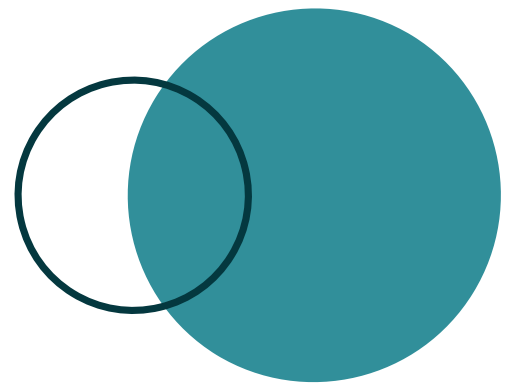
# METODOLOGIA SCRUM

## FASES

As **fases** do ciclo de vida do Scrum são:

### 1- Planejamento:

- Definição de novas funcionalidades baseado no conhecimento do sistema como um todo.
- Documento contendo a lista dos requisitos, em ordem de prioridade: Backlog do Produto (**Product Backlog**), que deve ser constantemente atualizado e priorizado.
- Técnicas para obter a lista de requisitos: entrevistas e questionários com o dono do projeto (**Product Owner**).
- Reunião de planejamento (**Sprint Planning Meeting**), visando definir a iteração (*Sprint*) inicial do projeto.
- **Cada *Sprint* tem duração de 2 a 4 semanas.**
- A equipe separa uma parte do topo do *Product Backlog* para o Sprint, formando o **Sprint Backlog**.





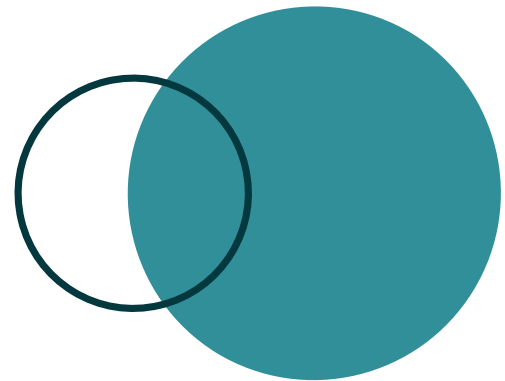
# MÉTODOS ÁGEIS

## METODOLOGIA SCRUM

As **fases** do ciclo de vida do Scrum são:

### 2- Desenvolvimento:

- As tarefas do **Sprint Backlog** são divididas entre os membros do time que são responsáveis por programar as novas funcionalidades e testá-las, respeitando o tempo previsto, requisitos exigidos e qualidade.
- Ao longo do Sprint reuniões são feitas diariamente (**Scrum Daily Meeting**), no mesmo horário, para acompanhar o progresso do trabalho.





# MÉTODOS ÁGEIS

## METODOLOGIA SCRUM

As **fases** do ciclo de vida do Scrum são:

### 3- Encerramento:

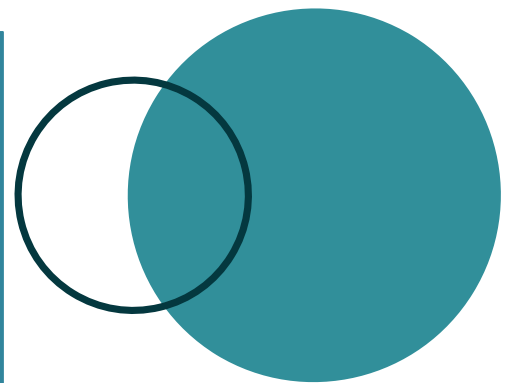
- A saída do Sprint é um conjunto de funcionalidades (incrementos) 100% desenvolvidas.

#### ***Sprint Review Meeting***

Reunião de revisão realizada para que seja apresentado o resultado alcançado. São validadas as funcionalidades e caso sejam necessárias, adaptações são realizadas.

#### ***Sprint Retrospective Meeting***

Reunião de retrospectiva que tem foco no processo, sendo uma oportunidade para a equipe se inspecionar, visando à criação de um plano de melhorias para os próximos trabalhos.



# MÉTODOS ÁGEIS

## PAPEIS

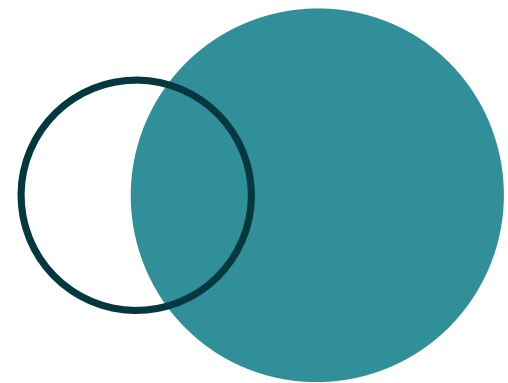
Os **papeis** do Scrum são:

### 1- Scrum Master:

- É responsável pelo processo SCRUM, por garantir que seja entendido e aplicado pela equipe, fazendo dele uma cultura na organização.
- É responsável também por proteger a equipe e assegurar que ela não se comprometa excessivamente com relação àquilo que é capaz de realizar durante um Sprint.
- Intermediador na **Daily Scrum Meeting** e torna-se responsável por remover quaisquer obstáculos que sejam levantados pela equipe.

### 2- Product Owner (Dono do Produto):

- Responsável por maximizar o valor do produto e por representar os interesses dos *stakeholders* no projeto.
- Gerencia todos os itens do **Backlog do Produto**.



# MÉTODOS ÁGEIS

## PAPEIS

Os **papeis** do Scrum são:

### 3- Team (Equipe):

- Grupo de pessoas (de 5 a 9 pessoas), que trabalha no desenvolvimento do produto. Deve ser multidisciplinar, auto gerenciável e trabalhar em conjunto.
- O cliente é parte da equipe, tendo participação ativa no processo de desenvolvimento.
- Não existe um líder de equipe.
- A equipe tem total autonomia para gerir seu trabalho, incluindo distribuir e decidir como serão realizadas as tarefas.



# MÉTODOS ÁGEIS

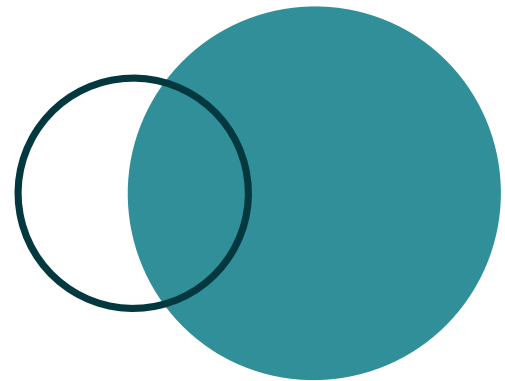
## ARTEFATOS

Os **artefatos** do Scrum são:

### 1- Backlog do Produto:

- Lista de funcionalidades que são necessárias no produto, sendo esses itens ordenados por prioridade pelo Product Owner.
- Pode ser descrito como uma lista de histórias do usuário (**user stories**).

**História de usuário** é um instrumento de escrita utilizado no processo de levantamento de requisitos para descrever a especificação de uma funcionalidade de software.



# MÉTODOS ÁGEIS

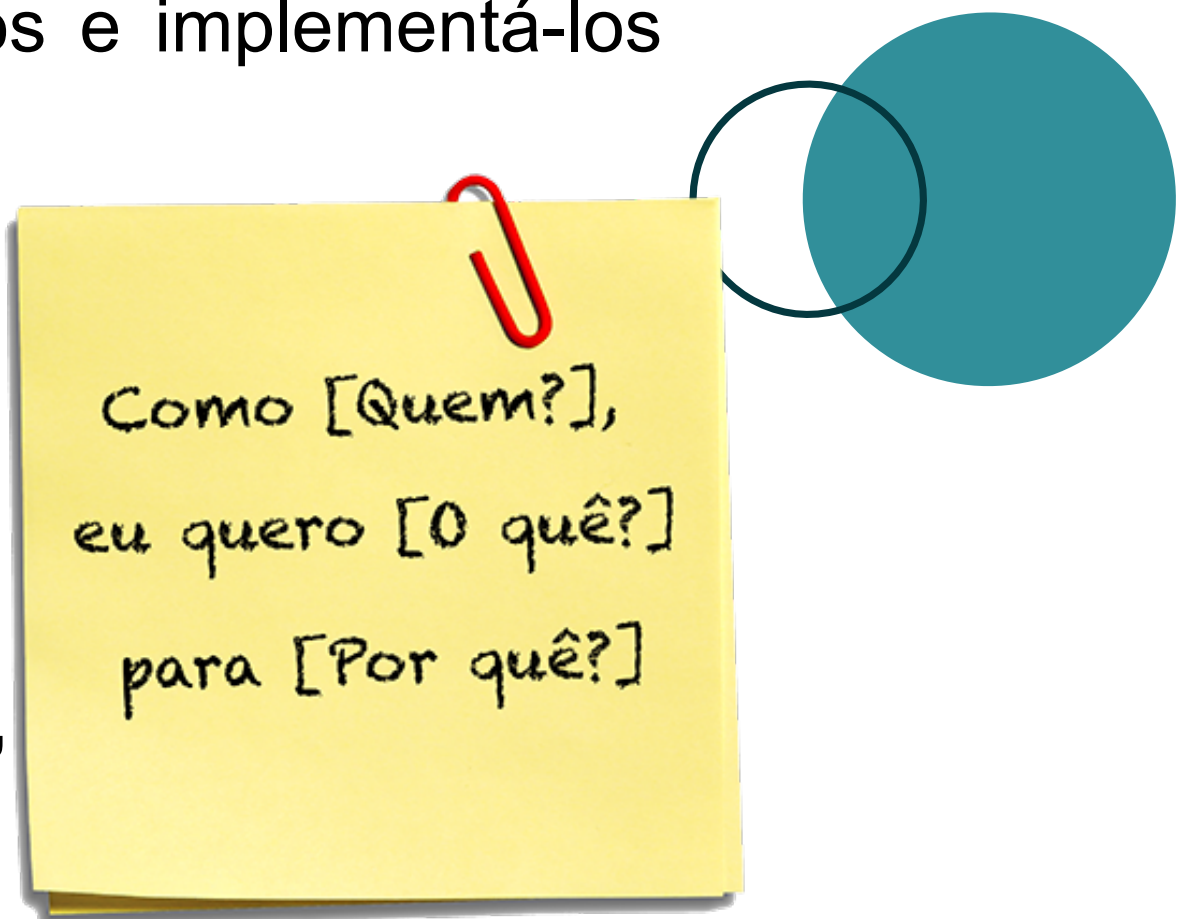
## ARTEFATOS

### História de usuário

- A escrita da história do usuário deve ocorrer sem o uso de termos técnicos.
- Elas devem ser compreensíveis pelos homens de negócios e seu conteúdo deve caber em um cartão de índice.
- Deve ser possível explicá-los em poucos segundos e implementá-los em menos de uma semana.

- A sintaxe de uma história de usuário pode ser:

**“Como um (usuário ou ator), eu quero (realizar determinada capacidade ou funcionalidade) para que eu (possa ter o seguinte valor de negócio ou benefício).”**



Como [Quem?],  
eu quero [O quê?]  
para [Por quê?]

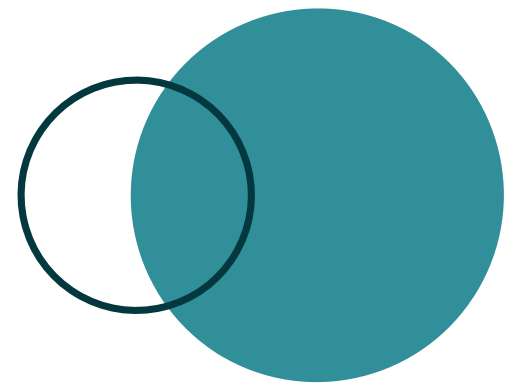
# MÉTODOS ÁGEIS

## ARTEFATOS

### História de usuário – Exemplo:

Formas de Pagamento

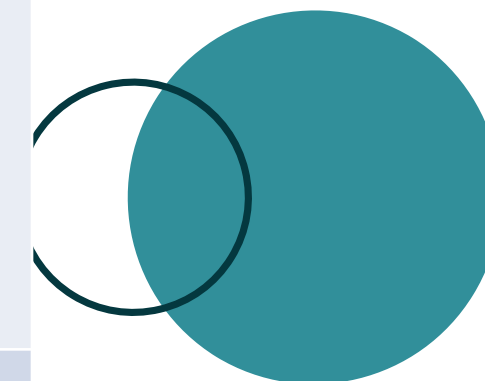
Como um Cliente,  
Eu quero que sejam disponibilizadas  
diversas formas de pagamento  
Para pagar meu pedido.





## Backlog do Produto – Exemplo:

| BP01 - Backlog do Produto |                                       |             |    |   |                                     |
|---------------------------|---------------------------------------|-------------|----|---|-------------------------------------|
| ID                        | Nome                                  | Importância | PH | Demonstrar  | Notas                               |
| HU1                       | Cadastrar as taxas do estacionamento. | 30          | 8  | Como um usuário do sistema eu quero cadastrar o valor das taxas de estacionamento (por hora/diária/mensalista e fração por 15min) para realizar o calculo do valor a ser pago posteriormente.   | Exigir senha de gerente do sistema. |
| HU2                       | Visualizar dados do Estacionamento.   | 30          | 8  | Como um usuário do sistema eu quero visualizar informações do estacionamento para ver número de vagas livres e ocupadas, taxas do estacionamento por hora/diária/mensalista e fração por 15min. |                                     |





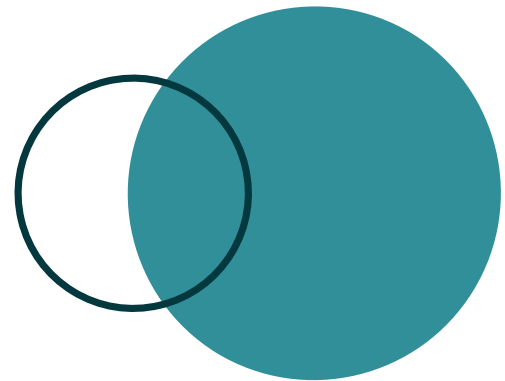
# MÉTODOS ÁGEIS

## ARTEFATOS

Os **artefatos** do Scrum são:

### 2- Testes de Aceitação:

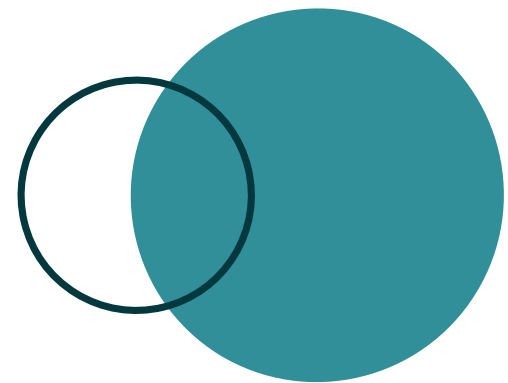
- Para validar se a história foi implementada de acordo com as especificações solicitadas pelo PO.
- Casos de teste da história podem ser escritos da seguinte forma:  
**“Dado (uma pré-condição), quando (o usuário realiza uma ação) então (esse é o resultado esperado).”**



Os **artefatos** do Scrum são:

## 2- Testes de Aceitação – Exemplo:

| TA01 - Testes de Aceitação |   |
|----------------------------|---|
| ID da História do Usuário  | História do Usuário   |
| HU1                        | Como um usuário do sistema eu quero cadastrar o valor das taxas de estacionamento (por hora/diária/mensalista e fração por 15min) para realizar o calculo do valor a ser pago posteriormente. |
| ID do Caso de Teste        | Caso de Teste   |
| CT01                       | O valor informado para as taxas não pode ser negativo.  |
| CT02                       | Somente usuário com perfil “gerente” pode realizar o cadastro de taxas.   |

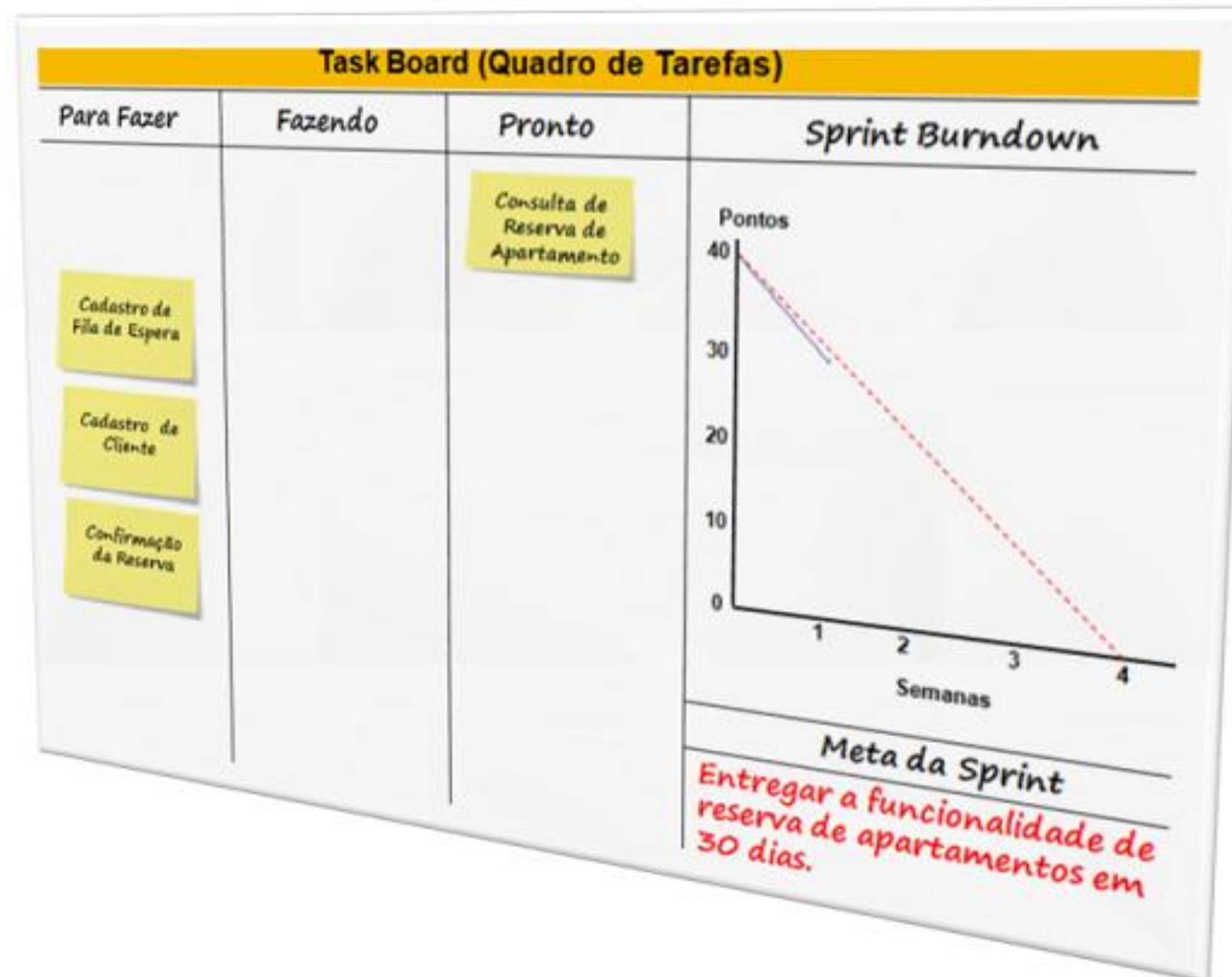


# MÉTODOS ÁGEIS

## ARTEFATOS

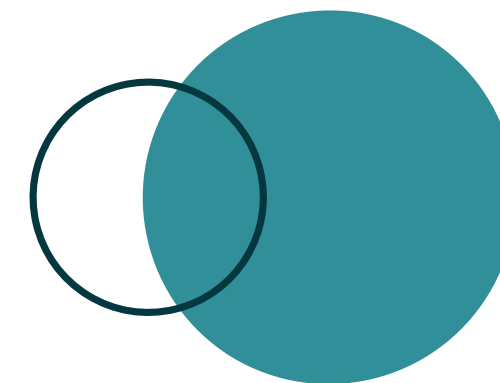
Os **artefatos** do Scrum são:

3- Quadro de tarefas:



# DÚVIDAS?

ENTRE EM CONTATO: [samara.leal@prof.unibh.br](mailto:samara.leal@prof.unibh.br)  
[otaviano.sousa@prof.unibh.br](mailto:otaviano.sousa@prof.unibh.br)



NÃO SE ESQUEÇA DE CONSULTAR O REFERENCIAL  
BIBLIOGRÁFICO e BUSCA ATIVA!