

# Métodos Ágeis

MAC0470/5856 - Desenvolvimento de Software Livre  
Professor Daniel Batista  
Anderson Andrei da Silva, 8944025

# Apêndice

- História / Conceitos
- O Manifesto Ágil
  - As 4 ideologias
  - Os 12 princípios
- Metodologias Ágeis
  - Scrum
  - Kanban
  - XP
  - Outros



# História / Conceitos

- Evolução a partir de metade de 1990
- Reação contra “métodos pesados”
  - Regulamentação e regimentos rigorosos
  - Métodos de gerenciamento (cascata)
- “Métodos leves”
- *Scrum*, 1986
- Kanban, 1960
- *Extreme Programming*, 1996
- *ASD, FDD, DSDM*, 1995



# O Manifesto Ágil

- Antigo conceito de “métodos leves”
- 2000, reunião entre líderes da comunidade de *Extreme Programming*
- Fevereiro de 2001, em Utah no resort de inverno Snowbird
- 17 participantes :

Kent Beck, Mike Beedle, Arie van 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




# O Manifesto Ágil - Os 4 valores

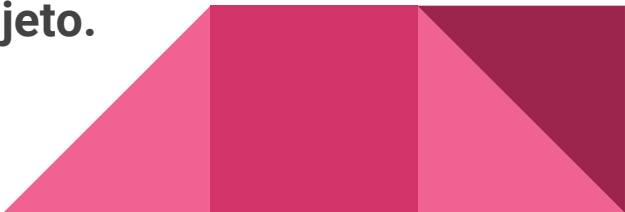
“Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar :

1. **Indivíduos e interações** mais que processos e ferramentas
2. **Software em funcionamento** mais que documentação abrangente
3. **Colaboração com o cliente** mais que negociação de contratos
4. **Responder a mudanças** mais que seguir um plano

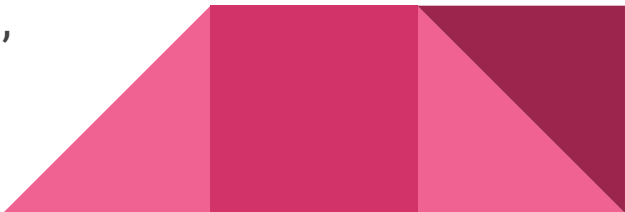
Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.”



# O Manifesto Ágil - Os 12 princípios

- **Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.**
  - Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento.  
Processos ágeis se adequam a mudanças, para que o cliente possa tirar **vantagens competitivas.**
  - **Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.**
  - **Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.**
- 

# O Manifesto Ágil - Os 12 princípios

- **Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.**
  - O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
  - Software funcional é a medida primária de progresso.
  - Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
  - Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
- 

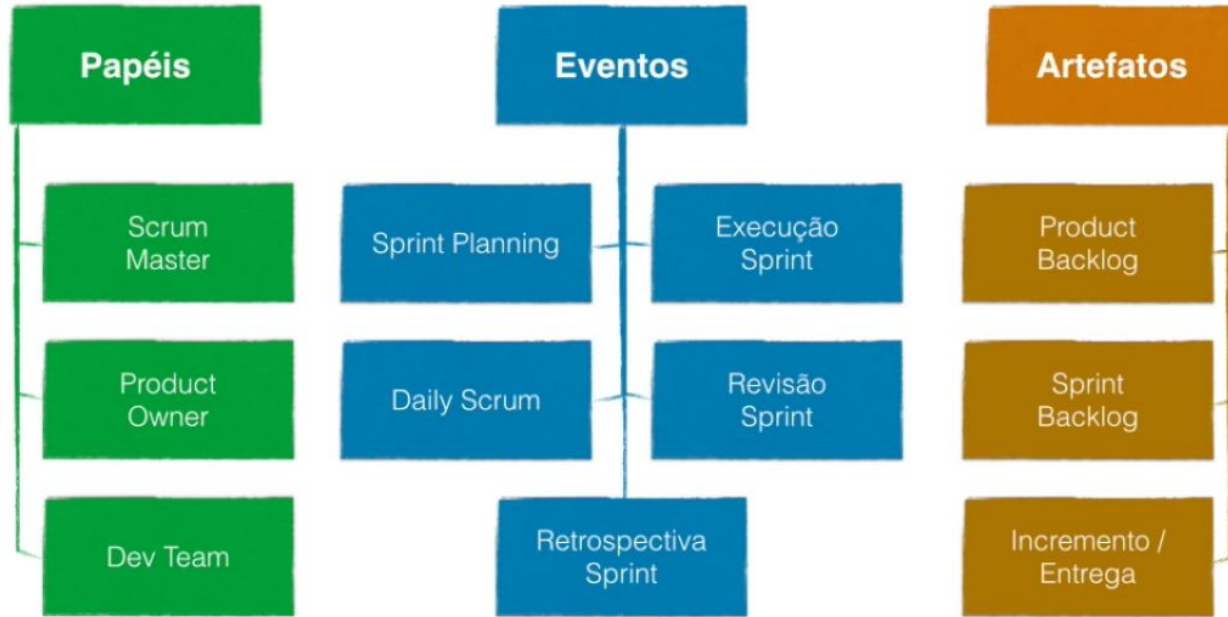
# O Manifesto Ágil - Os 12 princípios

- **Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.**
- As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
- Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

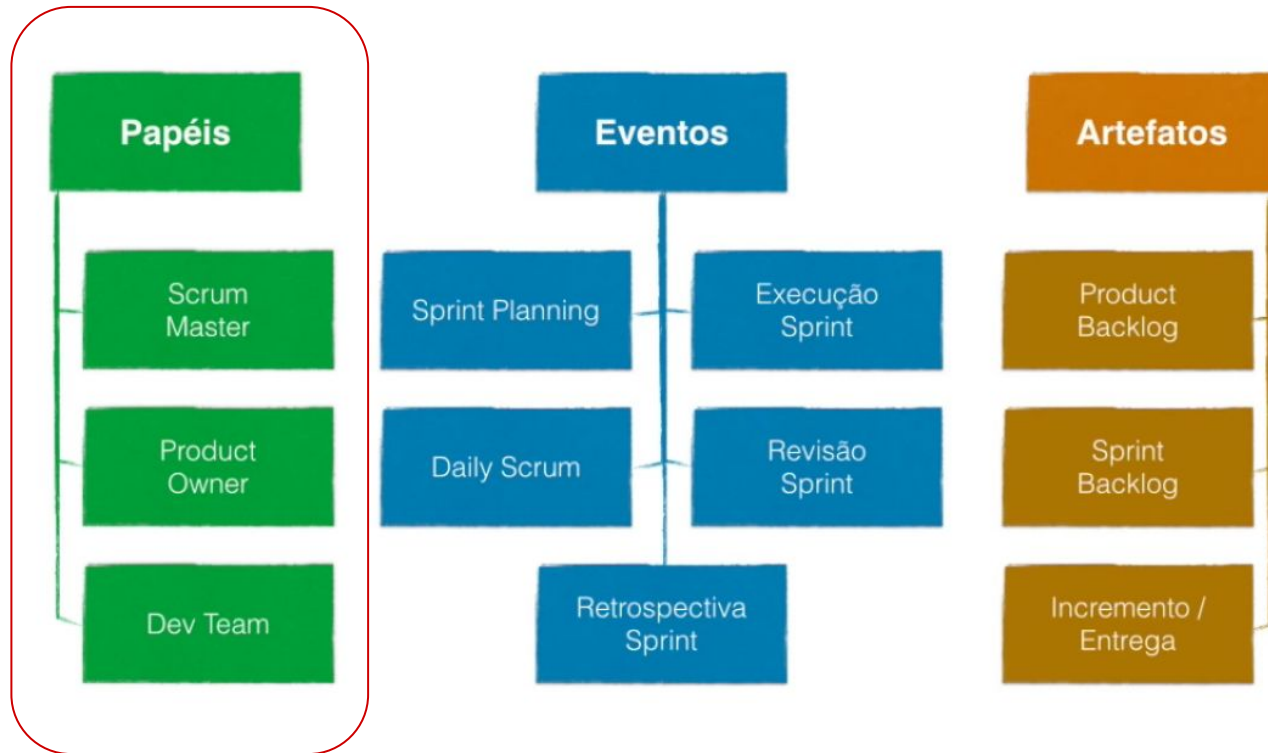




# Scrum - práticas fundamentais



# Scrum - práticas fundamentais



# Scrum - papéis



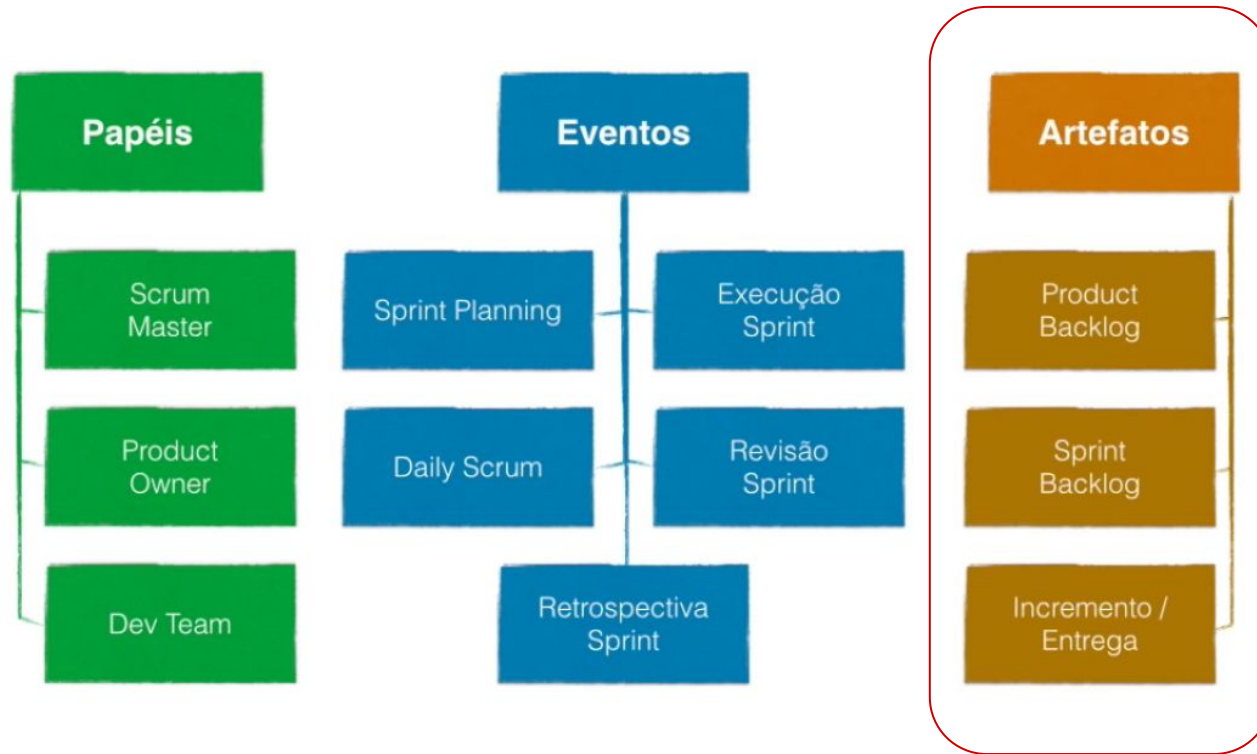
Dev Team

Product Owner

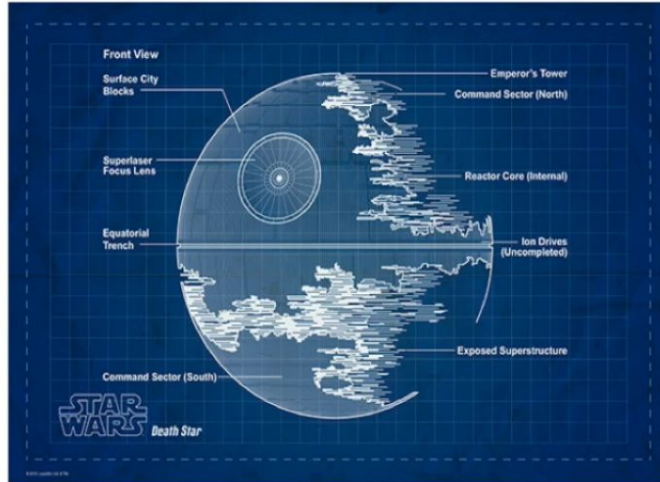


Scrum Master

# Scrum - práticas fundamentais



# Scrum - product backlog



Fonte : <http://www.mindmaster.com.br/scrum/>

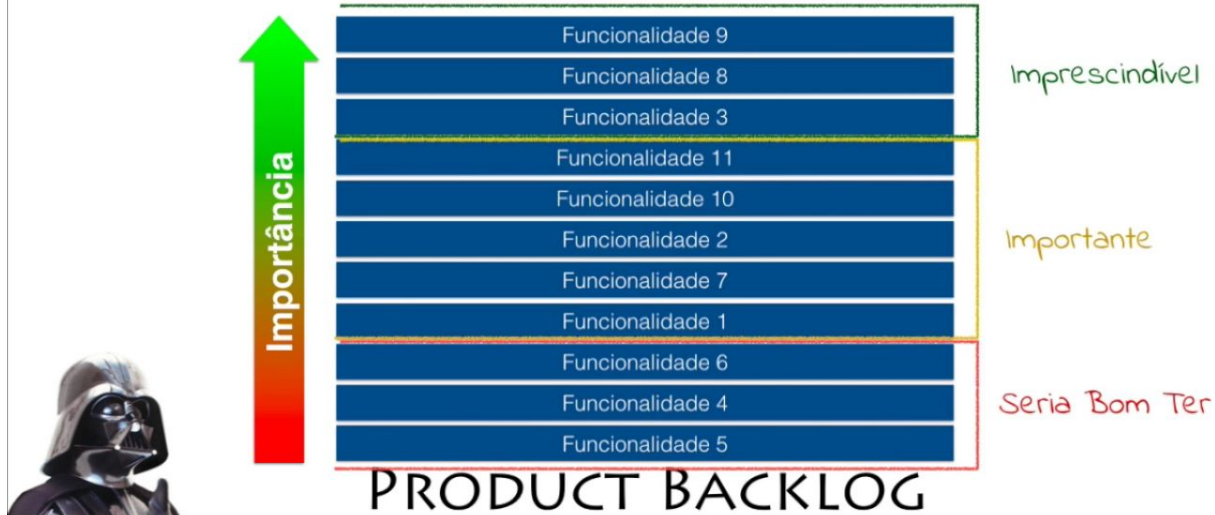
# Scrum - product backlog



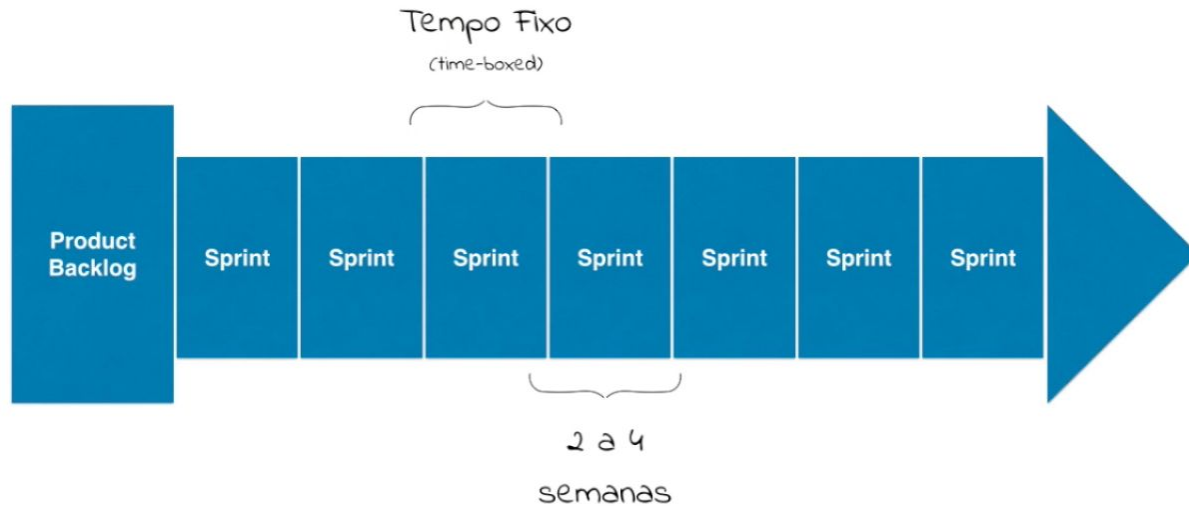
Funcionalidade 1
Funcionalidade 2
Funcionalidade 3
Funcionalidade 4
Funcionalidade 5
Funcionalidade 6
Funcionalidade 7
Funcionalidade 8
Funcionalidade 9
Funcionalidade 10
Funcionalidade 11

PRODUCT BACKLOG

# Scrum - sprint backlog

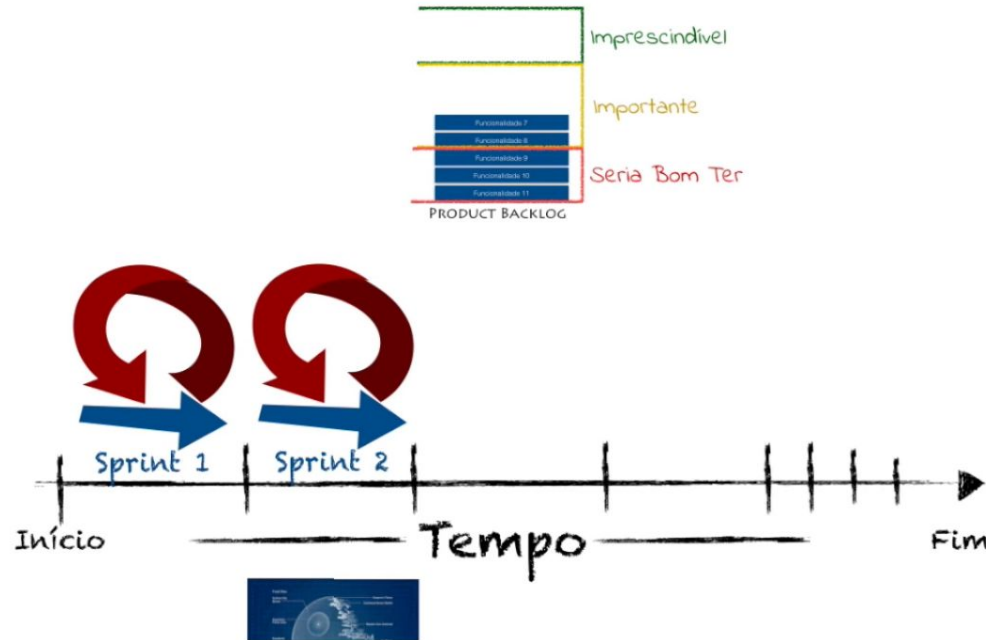


# Scrum - sprints

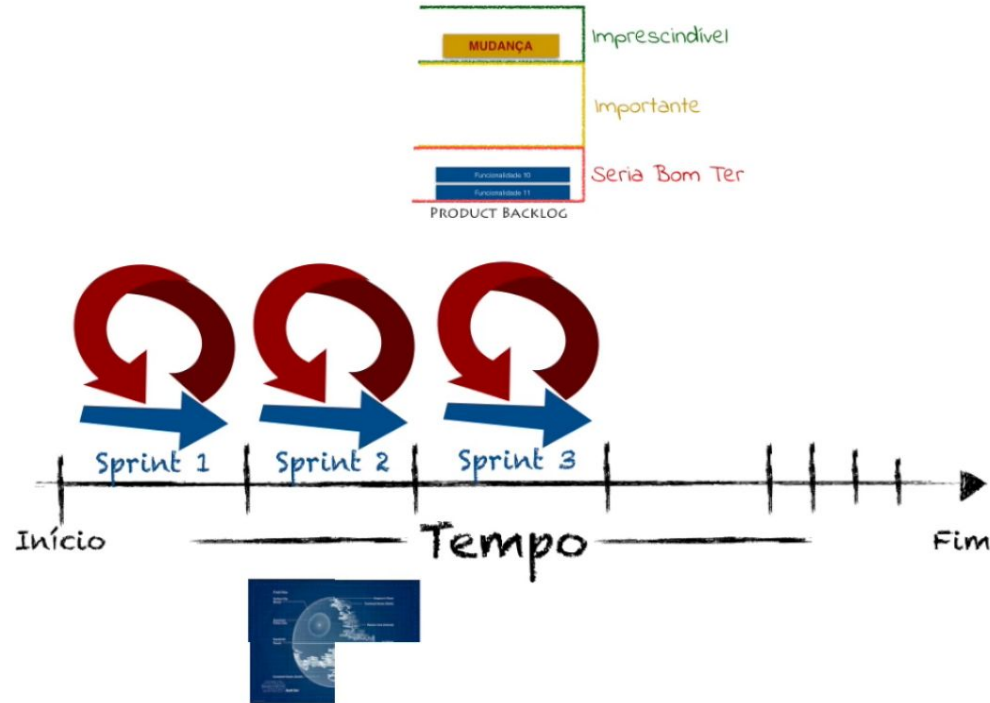




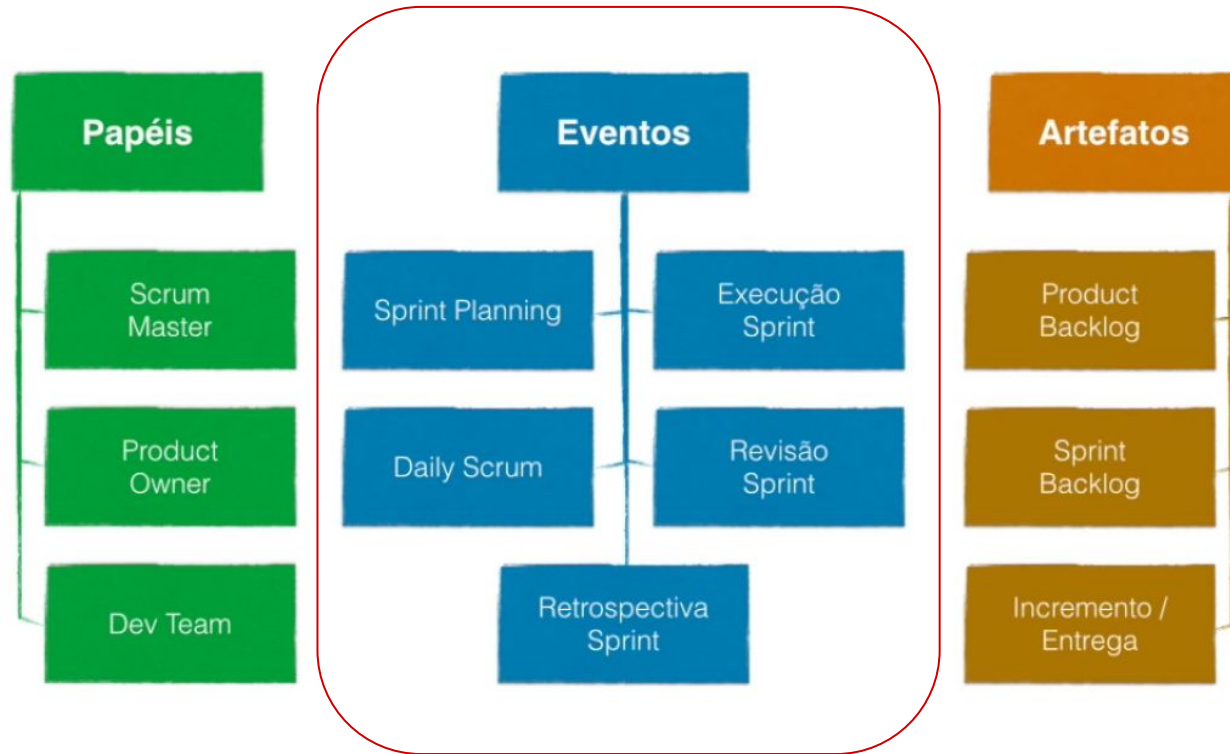
# Scrum - incremento / entrega



# Scrum - incremento / entrega



# Scrum - práticas fundamentais



# Scrum - eventos



Fonte : <http://www.mindmaster.com.br/scrum/>

# Scrum - *daily scrum*

**Daily Scrum** : é uma reunião diária, na qual define-se as tarefas do dia.

- Não deve haver atraso para iniciar esta reunião
- Geralmente poucas pessoas falam (somente quem precisa falar)
- Não deve durar mais que 15 minutos
- Deve haver constância de dia, horário e local
- Deve-se responder às três perguntas abaixo em toda reunião:
  - O que foi feito de ontem pra hoje?
  - O que será feito hoje?
  - Teve problemas para realizar suas tarefas?



# *Scrum - sprint review*

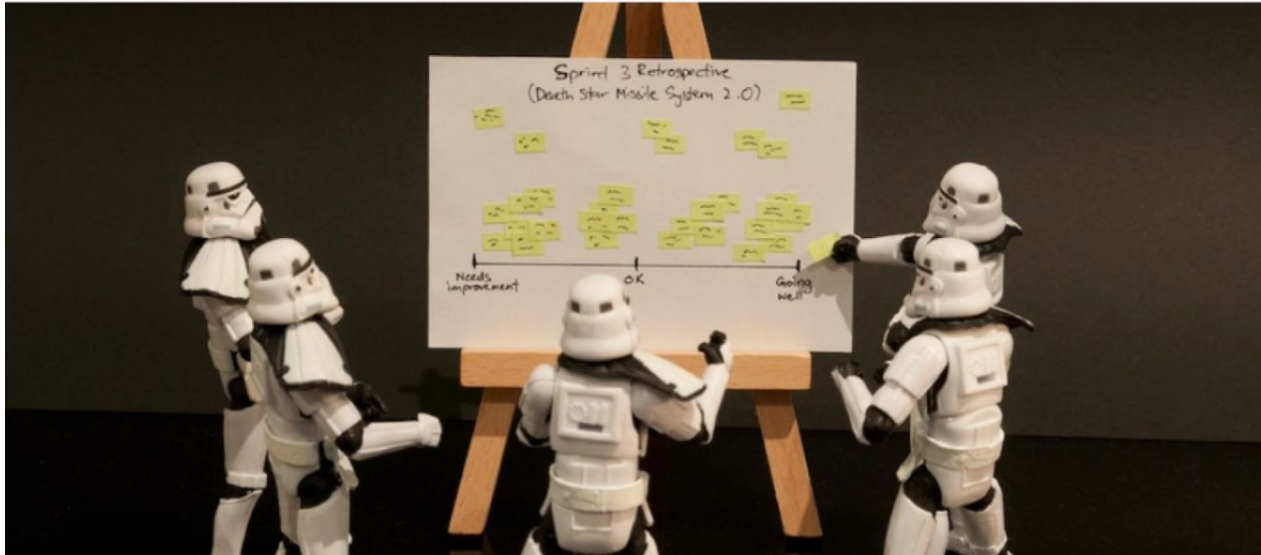
## Sprint Review



Fonte : <http://www.mindmaster.com.br/scrum/>

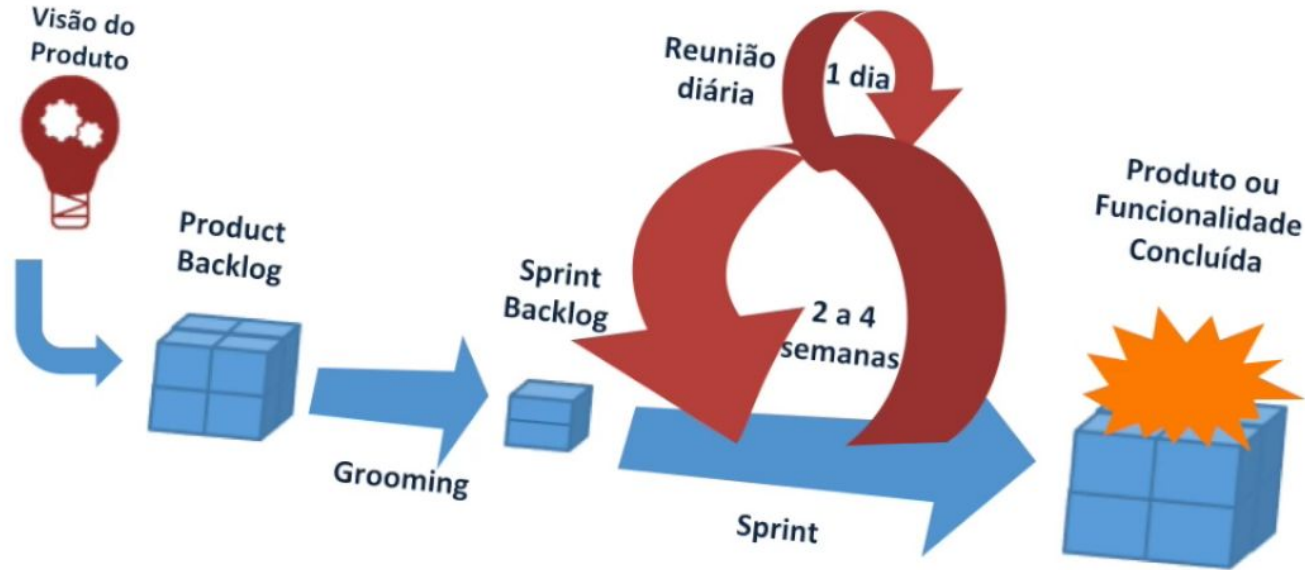
# Scrum - sprint retrospective

## Retrospectiva



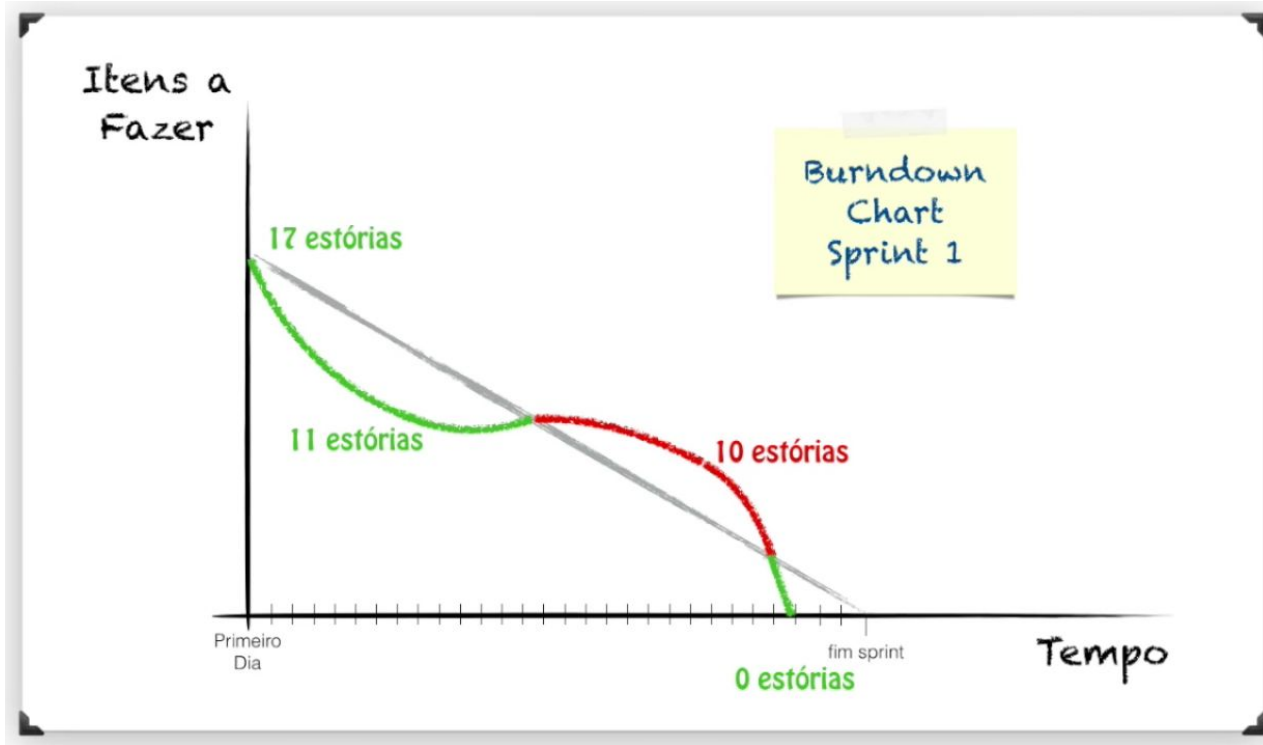
Fonte : <http://www.mindmaster.com.br/scrum/>

# Scrum - visão geral

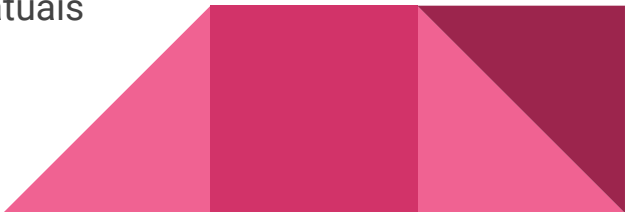




# Agregação - *Burndown charge*



# Kanban

- Palavra japonesa e seu significado literal é “cartão” ou “sinalização”
  - Toyota
    - Técnicas de “prateleira”
    - Reposição sob demanda
  - Implantação de mudanças que não prescreve papéis ou práticas específicas
  - Melhorar o desempenho e reduzir desperdício
  - 4 princípios fundamentais :
    - Comece com o que você faz agora
    - Concordar em buscar mudanças evolucionárias
    - Inicialmente, respeite os papéis, responsabilidades e cargos atuais
    - Incentivar atos de liderança em todos os níveis
- 

# Kanban

Composto por 6 práticas :

- Visualizar o fluxo de trabalho (workflow)
- Limitar a quantidade de trabalho em andamento (WIP)
- Gerenciar e medir o fluxo
- Tornar as políticas do processo explícitas
- Implementar loops de feedback
- Usar modelos para reconhecer oportunidades de melhoria



# Kanban - classificação de itens e hierarquia

- Épicos (**Epics**) - é uma grande história.
- História de usuário (**User Story**) - Uma história de usuário pode ser caracterizada como uma curta e simples descrição de um recurso.
  - " Como/Sendo <quem>, eu quero/gostaria/devo/posso <o que>, para que/de/para <porque/resultado>."
- Tarefas (**Tasks**) - As tarefas são os elementos de uma história.
  - Ex: "Como usuário comum, eu quero me cadastrar na landing page através do formulário no topo, por que desejo receber o ebook prometido"
    - uma tarefa dessa história poderia ser "Integrar API de e-mail ao formulário".

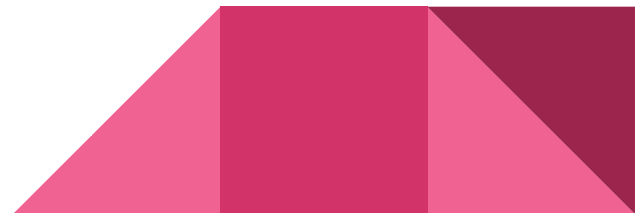
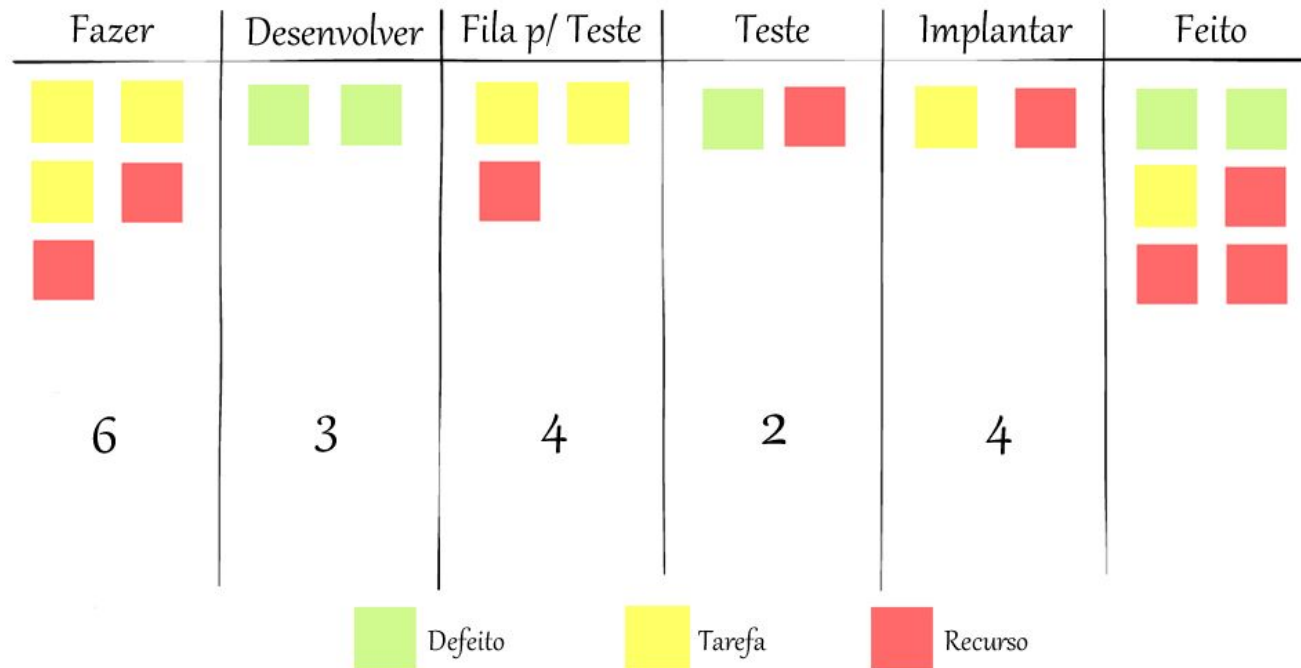
# Kanban

Podem haver :

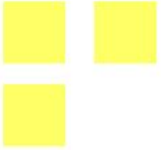





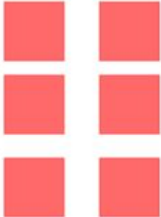




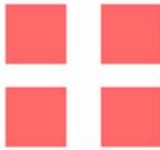
- Priorização de itens
  - Defeito
  - Tarefa
  - Recurso
- Raias



# Kanban - *board*



# Kanban - *board*

	Fazer	Desenvolver	Fila p/ Teste	Teste	Implantar	Feito
TAREFAS	 3	 3	 2	 2	 2	
BUGS	 6	 4	 4	 3	 3	

# “Scrumban”

- Sprints de extensão fixa e as funções a partir do Scrum
- Foco nos limites de trabalho em progresso e tempo de ciclo a partir do Kanban





# *Extreme Programming - XP*

- Equipes pequenas a médias
- Requisitos vagos e alteráveis
- Feedback constante
- Abordagem incremental
- Encorajamento da comunicação



# *Extreme Programming* - princípios básicos

- **Comunicação**

- Bom relacionamento cliente - desenvolvedor
- Conversas pessoais a outros meios de comunicação.

- **Simplicidade**

- Software com o menor número possível de classes e métodos
- Implementar apenas requisitos atuais



# *Extreme Programming* - princípios básicos

- **Feedback**

- Informações constantes do código e do cliente
  - Código - testes próprios
  - Cliente - teste dos clientes

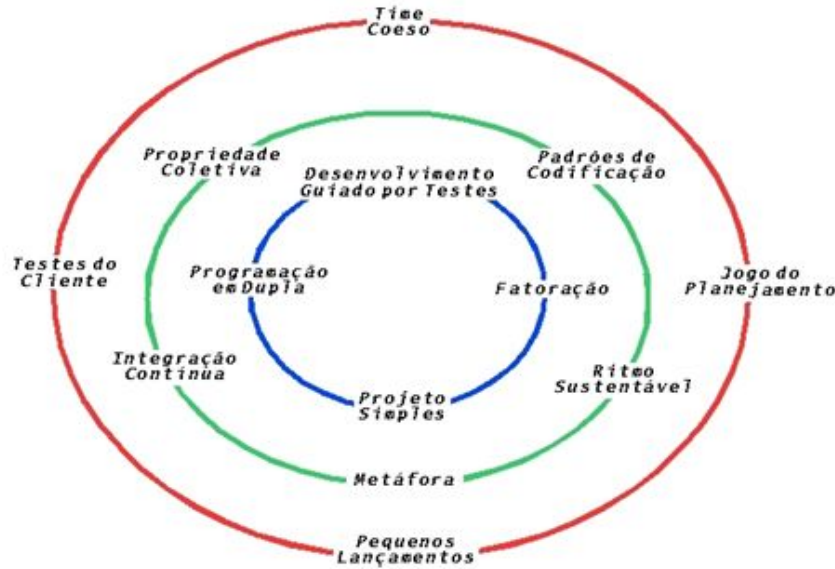
- **Coragem**

- Trocar de paradigma
- Aprender com os erros
- Dar e receber feedback sem medo das consequências
- Acreditar no feedback concreto (não na “teoria”)
- Fazer o que precisa ser feito
- Jogar fora código ruim
- Jogar fora protótipos criados para testar ideias



# Extreme Programming - principais práticas

## Práticas XP



# Extreme Programming - feedback

## Extreme Programming Planning/Feedback Loops



© J. Donovan Wells

**Fonte :** <https://enquantoisso.com/curso-gratis-extreme-programming-metodologia-agil-de-desenvolvimento-de-software-online-e-onde-fazer/>

# *Extreme Programming - papéis*

- Programador : “o coração do XP” ;
- Cliente ;
- Testador : testes funcionais com execução frequente;
- Rastreador : “a consciência da equipe”, responsável por finalizar a iteração ;
- Treinador : responsável por manter as práticas XP ;
- Consultor : (não obrigatório) ;
- Chefe .



# Metodologias ágeis - outros exemplos

- *Feature Driven Development (FDD)*
- *Dynamic System Development Model (DSDM)*
- *Adaptive Software Development (ASD)*
- *RUP*
- *OpenUp*
- *Lean Development*



# Referências

1. <https://www.projectbuilder.com.br/blog/quais-sao-os-principais-tipos-de-metodos-ageis/>
2. <https://www.culturaagil.com.br/o-que-sao-metodos-ageis/>
3. <https://targetteal.com/pt/blog/o-que-sao-metodologias-ageis/>
4. <https://becode.com.br/manifesto-agil-e-top-metodologias-ageis/>
5. <https://becode.com.br/scrum-para-iniciantes/>
6. <http://www.desenvolvimentoagil.com.br/scrum/>
7. <http://www.mindmaster.com.br/scrum/>
8. <https://www.culturaagil.com.br/kanban-do-inicio-ao-fim/>
9. <https://br.atlassian.com/agile/kanban>
10. <http://www.manifestoagil.com.br/>
11. <https://www.devmedia.com.br/introducao-ao-extreme-programming-xp/29249>
12. <https://www.devmedia.com.br/integrando-xp-as-principais-metodologias-ageis/30989>
13. <https://enquantoisso.com/curso-gratis-extreme-programming-metodologia-agil-de-desenvolvimento-de-software-online-e-onde-fazer/>





# Be Agile !

## Dúvidas ?

MAC0470/5856 - Desenvolvimento de Software Livre

Professor Daniel Batista

Anderson Andrei da Silva, 8944025

<https://github.com/andersonandrei/mac0470>