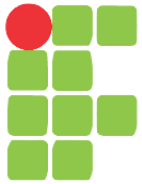


INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SERGIPE

Gestão Ágil de Projetos de Software

Prof. Me. Christiano Lima Santos

Sumário

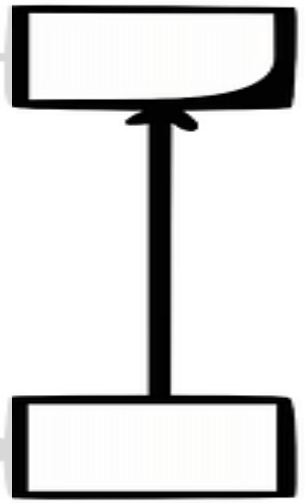
- Cenário atual da indústria de software
- Modelos de processo prescritivo
- Métodos ágeis
- Programação Extrema
- Scrum

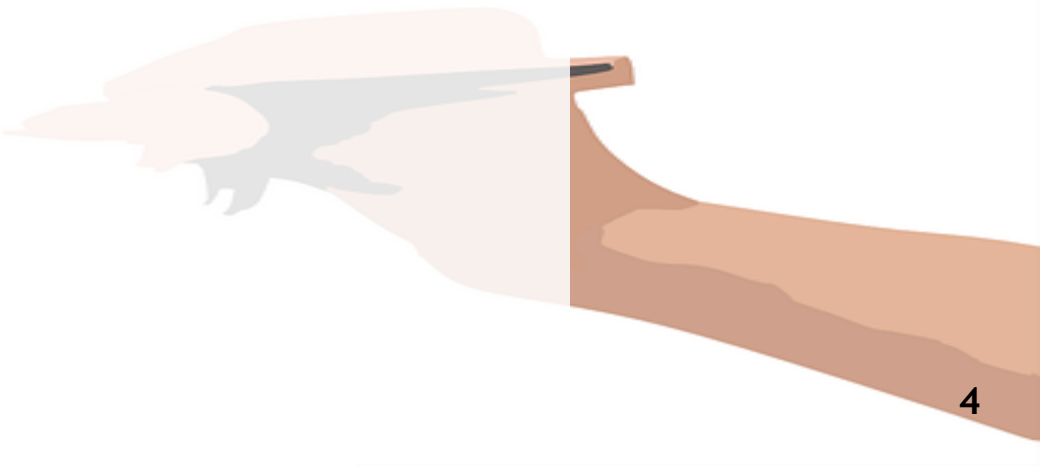
Cenário Atual da Indústria de Software

- Software afeta todos os aspectos de nossas vidas:
  - Financeiro – sistemas bancários *online*;
  - Educacional – ambientes virtuais de aprendizagem;
  - Social – redes sociais.
- A Engenharia de Software nos capacita para o desenvolvimento de softwares complexos dentro do prazo e com alta qualidade;
  - É aqui onde entram os processos e modelos de processo!

Processo de Software

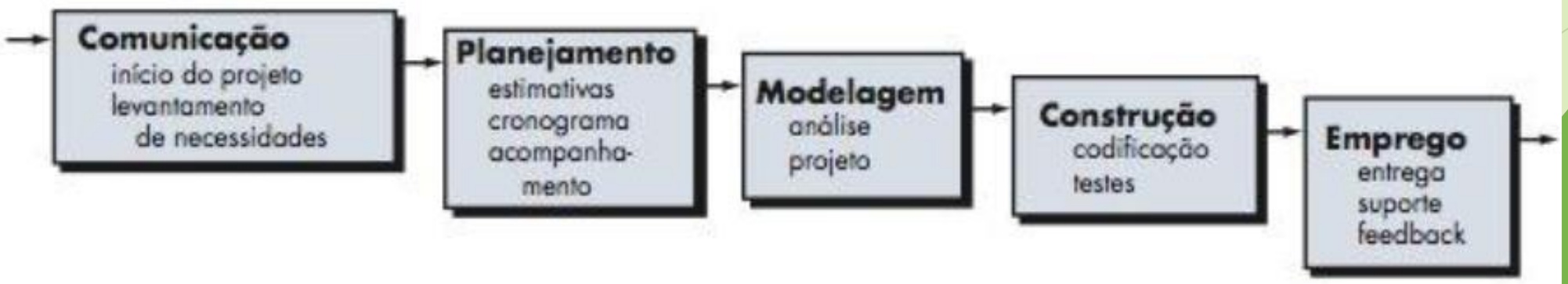
- **Processo** é um conjunto de atividades, ações e tarefas para a criação de algum produto de trabalho;
- Nos livros podemos encontrar diversas categorias de modelos de processo de software:
  - Prescritivos;
  - Iterativos;
  - Ágeis.





Modelos de Processo Prescritivo

- Prescrevem um conjunto de elementos de processo – atividades metodológicas, tarefas, produtos de trabalho etc. – e um fluxo de processo (fluxo de trabalho);
- Exemplo: Modelo Cascata, cujo fluxo de processo é linear.



Vamos praticar!

Quais são os possíveis problemas na adoção do modelo cascata?

Problemas do Modelo Cascata

Problema 1

Projetos reais raramente seguem o fluxo sequencial que o modelo propõe. Embora o modelo linear possa conter iterações, ele o faz indiretamente, podendo provocar confusão à medida que a equipe de projeto prossegue;

Problemas do Modelo Cascata

Problema 2

Frequentemente, é difícil para o cliente estabelecer explicitamente todas as necessidades no início do projeto, algo essencial no modelo cascata;

Problemas do Modelo Cascata

Problema 3

Uma versão operacional do(s) programa(s) não estará disponível antes de estarmos próximos do final do projeto. Um erro grave, se não detectado até o programa operacional ser revisto, pode ser desastroso.

Apelo da Indústria de Software

- Modelos prescritivos requerem muita disciplina para o sucesso em seu uso;
  - Mas não temos toda essa disciplina!
- Além disso, vivemos em um mundo repleto de mudanças:
  - No software em desenvolvimento;
  - Na equipe;
  - Nas tecnologias utilizadas.
- É necessária uma **abordagem mais ágil**!

Filosofia do Desenvolvimento Ágil

- Satisfação do cliente por meio de **entregas incrementais**;
- **Equipes pequenas** e bem motivadas;
- **Simplicidade no desenvolvimento** reduzindo a quantidade de artefatos produzidos (subprodutos);
- **Comunicação contínua** entre todos os envolvidos.

Abordagens ágeis

- Há diversas abordagens:
  1. Programação Extrema;
  2. Scrum;
  3. Desenvolvimento de Software Adaptativo;
  4. Método de Desenvolvimento de Sistemas Dinâmicos;
  5. Crystal;
  6. Desenvolvimento Dirigido a Funcionalidades;
  7. Desenvolvimento de Software Enxuto;
  8. Modelagem Ágil;
  9. Processo Unificado Ágil.

Abordagens ágeis

- Conversaremos um pouco sobre duas delas:
  - Programação Extrema;
  - Scrum.

Programação Extrema - XP

- Envolve quatro atividades metodológicas:
  - Planejamento;
  - Projeto;
  - Codificação;
  - Testes.

Processo XP

1. **Planejamento**
   - Inicia com a identificação das funcionalidades por meio de histórias de usuário (*user stories*);
   - Para cada história de usuário, cliente atribui um valor (prioridade) e a equipe de desenvolvimento, um custo (tempo para implementar);
     - Histórias que demorem mais que uma a três semanas devem ser subdivididas.

História de usuário

- Definição de alto nível de um requisito, contendo apenas informação suficiente para que desenvolvedores possam definir uma estimativa do esforço para implementá-la;

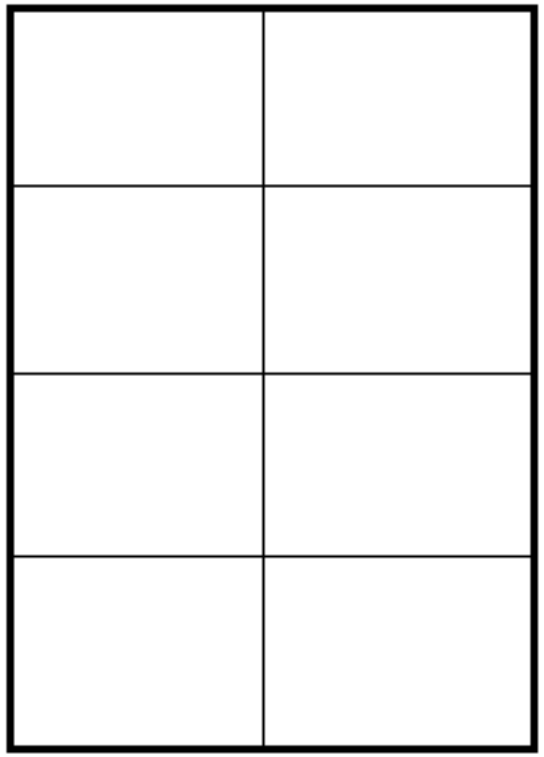
- Geralmente escritas em cartões 8cm x 13cm com o formato:

  Como um(a) <persona>, [quando <evento>] eu quero <ação> [tal que <resultado esperado>].

- Exemplos:
  - Como um administrador, eu quero cadastrar novos funcionários;
  - Como um leitor, quando eu atrasar a devolução de um livro, eu quero receber um e-mail notificando isso.

Vamos praticar!

Agora, elabore 6 histórias de usuário para um sistema de seu interesse!



Exemplos:

- ✓ Como um administrador, eu quero cadastrar novos funcionários;
- ✓ Como um leitor, quando eu atrasar a devolução de um livro, eu quero receber um e-mail notificando isso.

Processo XP

2. **Projeto**
   - Orienta a implementação das histórias de usuário na medida em que são escritas;
   - Encoraja-se uso de cartões CRC (classe-responsabilidade-colaborador);
   - Caso um difícil problema de projeto seja encontrado, recomenda-se implementar e avaliar um protótipo.

Cartão CRC

- O modelo CRC é uma coleção de cartões (8 cm x 13 cm) divididos em três seções (Classe, Responsabilidades e Colaboradores);
  - Uma **classe** é uma coleção de objetos similares (ex: coleção de livros, usuários, empréstimos etc.);
  - Uma **responsabilidade** é algo que uma classe sabe ou faz;
  - Um **colaborador** é outra classe com quem nossa classe interage a fim de cumprir suas responsabilidades.

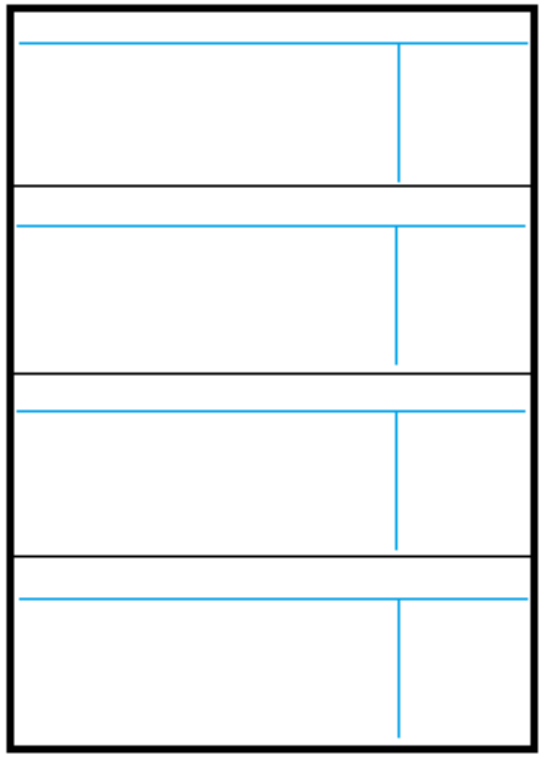
Cartão CRC

- Exemplo de Cartão CRC:

Classe Consumidor	
Responsabilidades Faz pedidos Sabe nome Sabe endereço Sabe telefone Sabe histórico de pedidos	**Colaboradores** Pedido

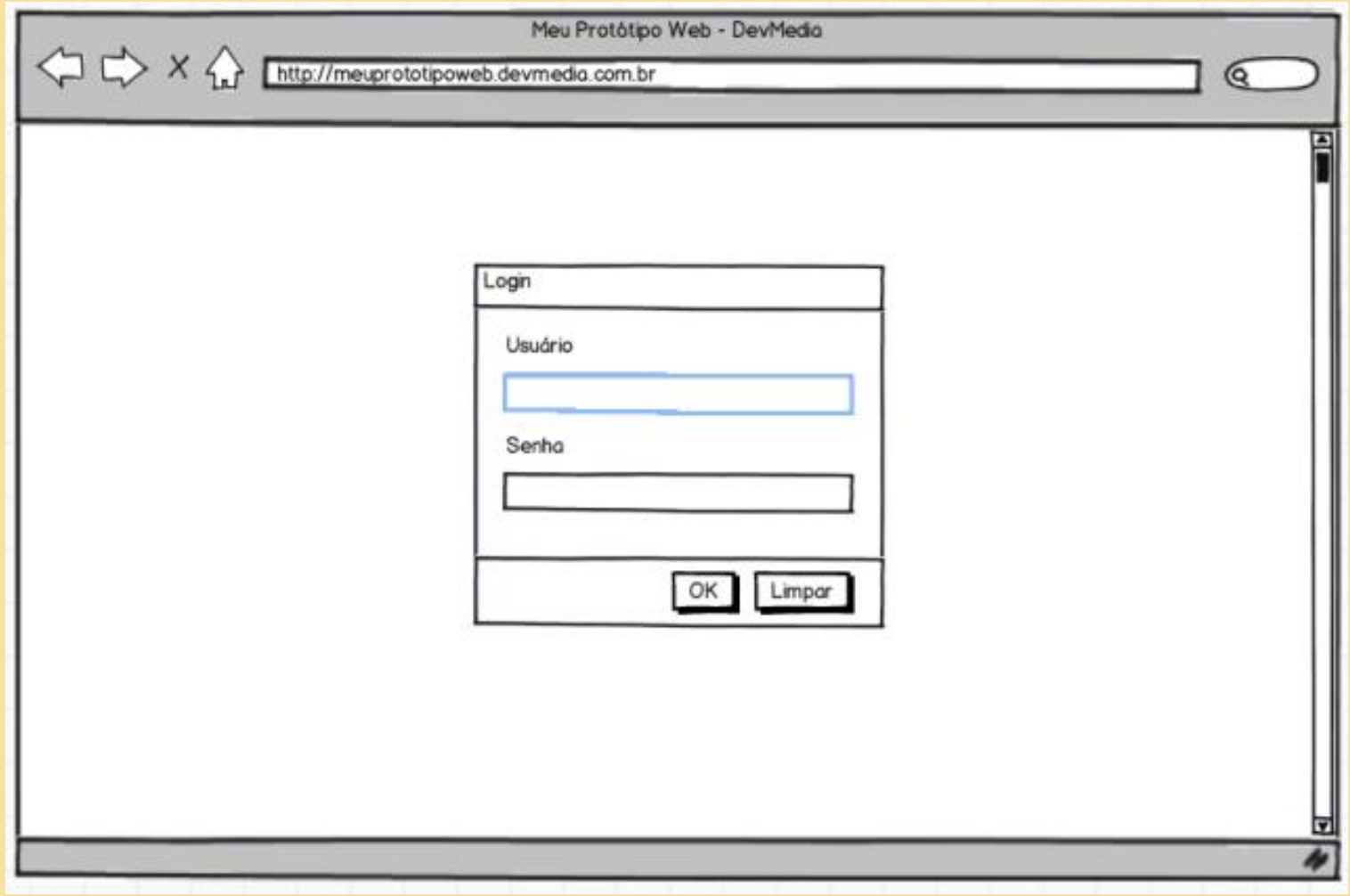
Vamos praticar!

Agora, elabore 3 cartões CRC!

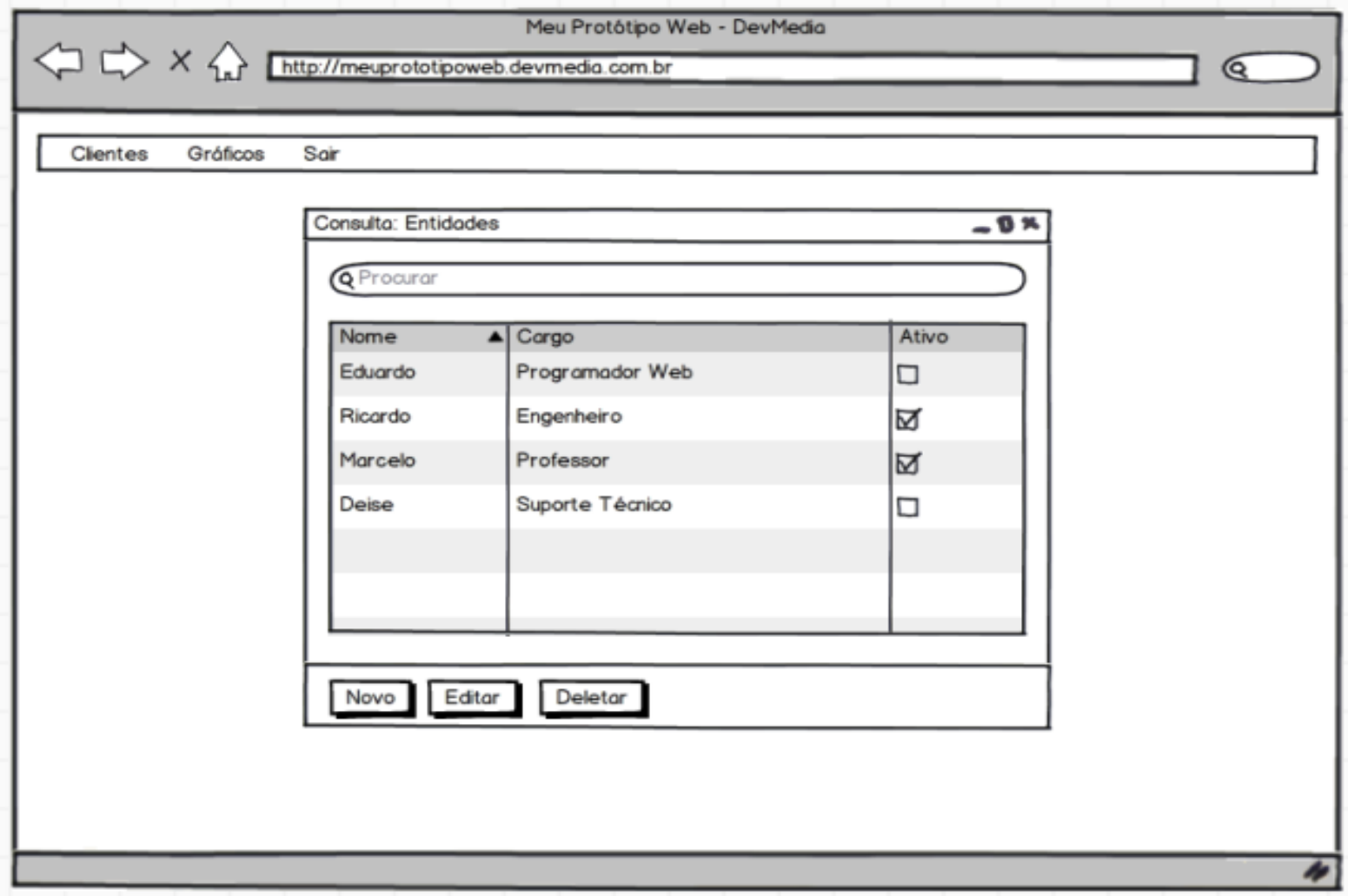


Classe Consumidor	
Responsabilidades Faz pedidos Sabe nome Sabe endereço Sabe telefone Sabe histórico de pedidos	**Colaboradores** Pedido

Protótipo de tela



Protótipo de tela



Processo XP

3. **Codificação**
   - Inicia com desenvolvimento de uma série de testes de unidade para cada história que será implementada no incremento atual;
   - Emprega programação em pares. Assim, o código é revisto na medida em que é criado;
   - Após implementação, o código é testado;
   - Integração contínua do código produzido.

Processo XP

4. **Testes**
   - Testes de unidade são definidos antes da codificação;
   - Foco na automatização dos testes de unidade;
   - Encorajamento de testes de regressão;
   - Testes de aceitação (testes de cliente) são obtidos a partir as histórias de usuário.

Exemplo de teste de unidade automatizado

```
public class MathExt {
    public static int factorial(int x) {
        if (x < 0) {
            return 0;
        } else if (x == 0) {
            return 1;
        } else {
            return x * factorial(x-1);
        }
    }
}
```

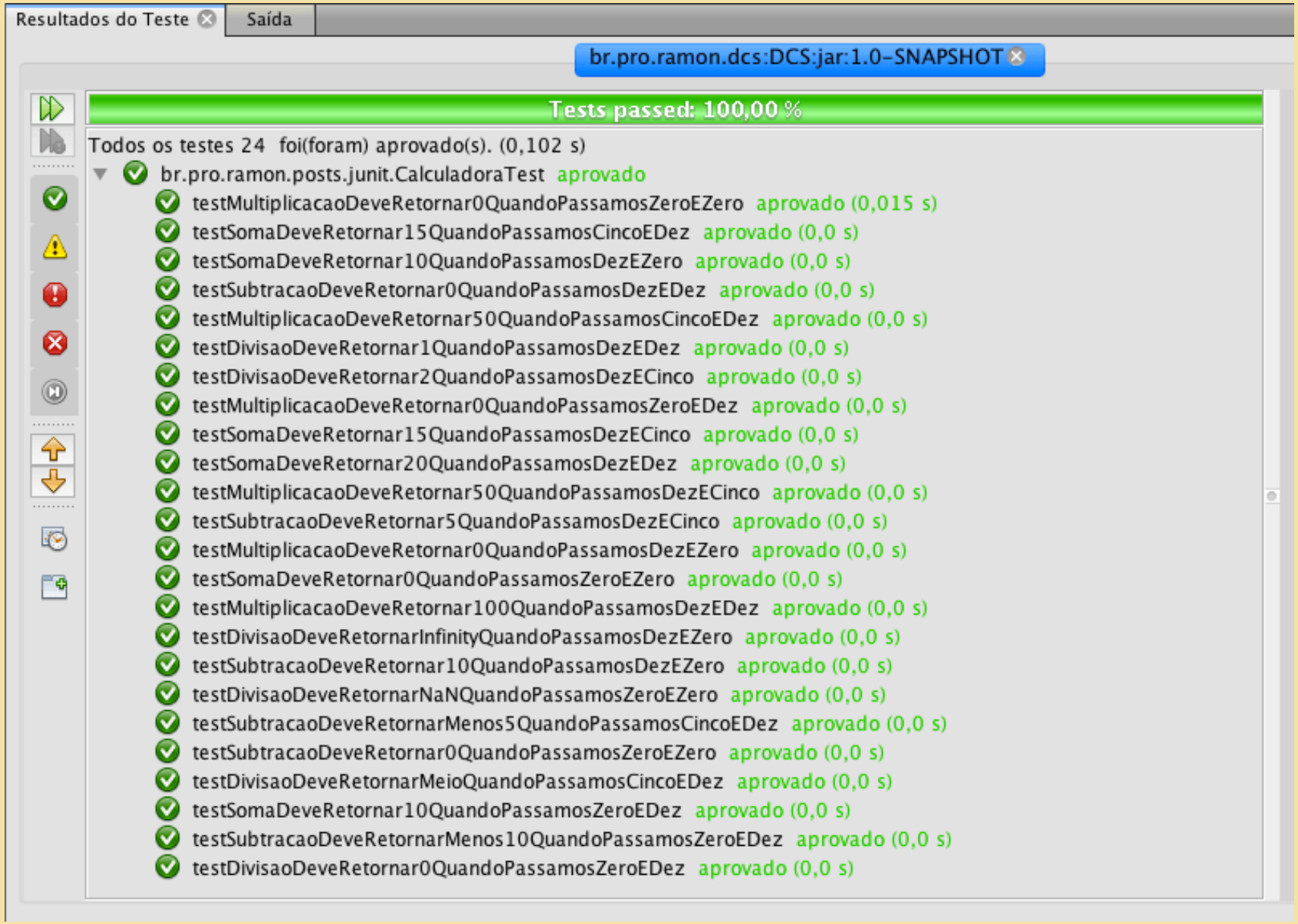
Classe a ser testada

Exemplo de teste de unidade automatizado

```
public class Tester {
    public static void main(String[] args) {
        if (MathExt.factorial(-1) != 0) {
            Logger.log("MathExt.factorial(-1) falhou!");
        }
        if (MathExt.factorial(0) != 1) {
            Logger.log("MathExt.factorial(0) falhou!");
        }
        ...
        Logger.log("Teste está concluído.");
    }
}
```

Classe testadora

Exemplo de teste de unidade em JUnit



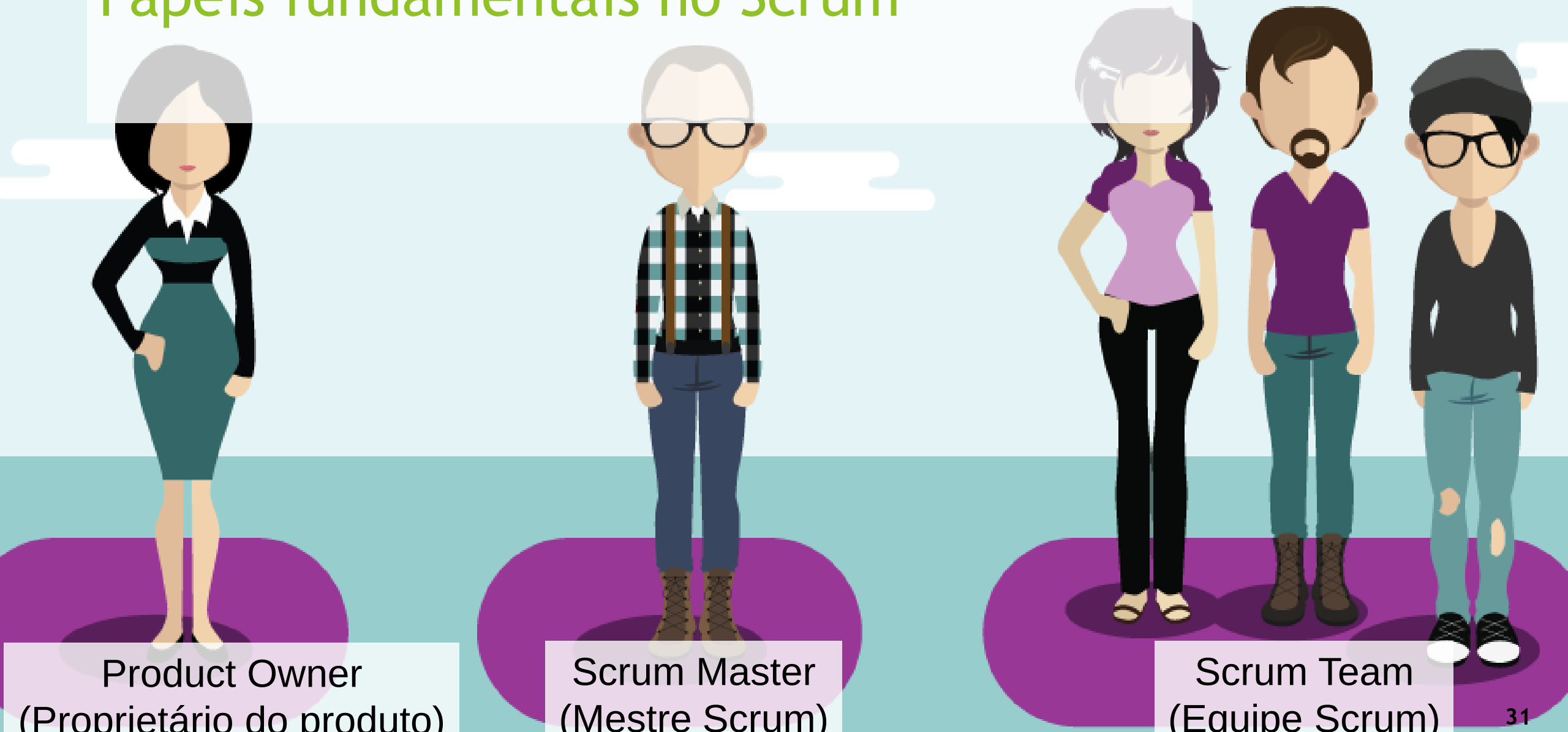
Resumindo...


projeto simples
cartões CRC
soluções pontuais
protótipos
valores das histórias
de usuários
critérios de teste de aceitação
plano de iteração
projeto
planejamento
codificação
refabricação
teste
programação em dupla
Versão
incremento de software
velocidade de projeto registrado
(computada)
teste de unidades
integração contínua
teste de aceitação

E onde entra o Scrum nisso tudo?

- Criado por Ken Schwaber e Jeff Shutterland no início da década de 1990 como **framework** para desenvolvimento de software **ágil**;
- Ele não diz como executar cada tarefa, mas esboça como gerir tais tarefas;
- Introduz novos artefatos e reuniões para controle do progresso do projeto.

Papéis fundamentais no Scrum



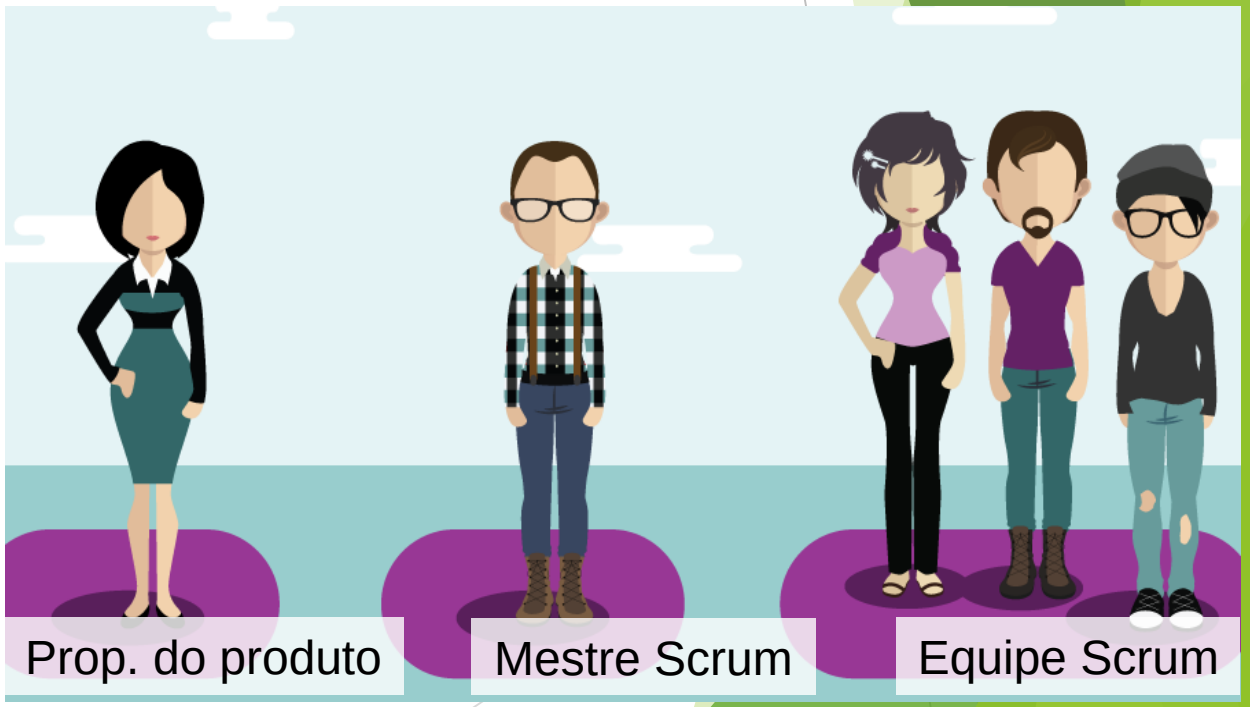
Product Owner
(Proprietário do produto)

Scrum Master
(Mestre Scrum)

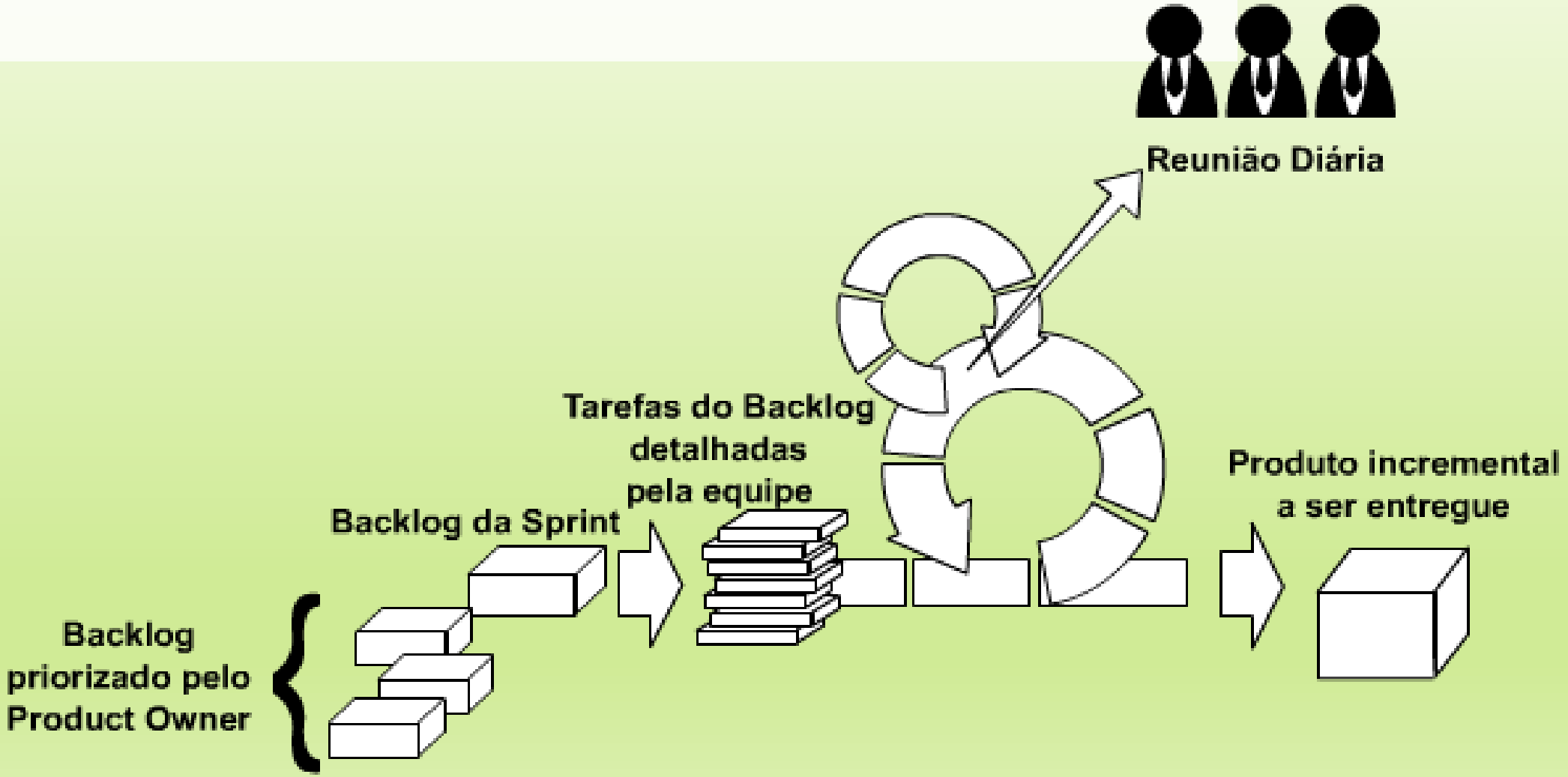
Scrum Team
(Equipe Scrum)

Vamos praticar!

Defina o papel de cada um em sua equipe!



Visão geral do Scrum



No início do projeto...

Precisamos conhecer várias coisas (exemplo de um sistema espacial)

- Tripulação;
- Alimentação;
- Duração das viagens;
- Sistema de navegação;
- Sistema de comunicação;
- Recursos oferecidos aos tripulantes;
- Sistema de segurança;
- Sistema gestor de oxigênio;
- Sistema de diagnóstico e tratamento de tripulantes.

Elicitação de requisitos → Elaboração do backlog do produto

Poker do Planejamento

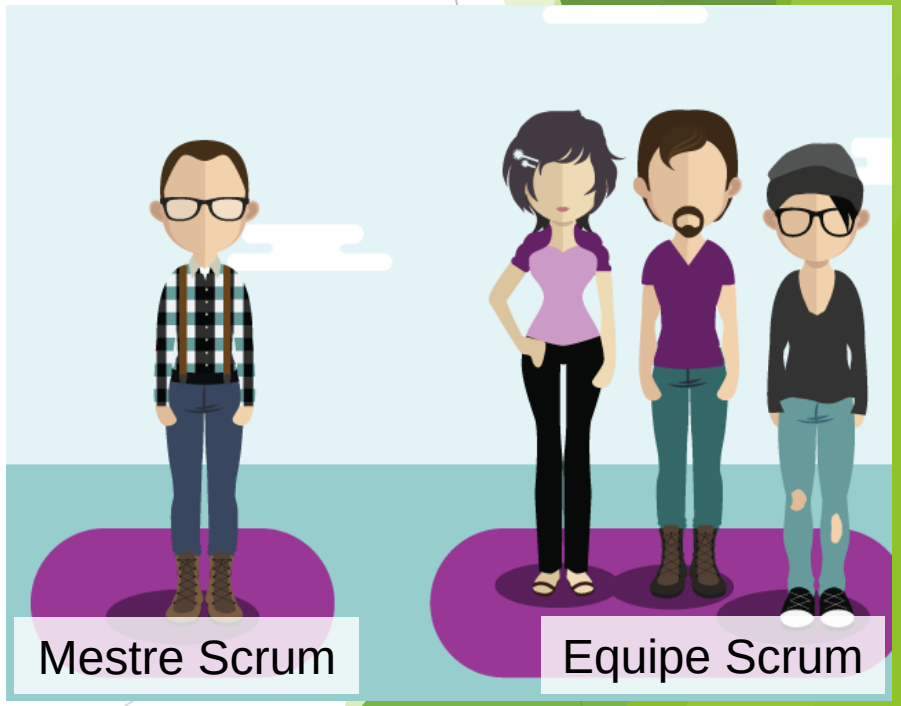
- Emprega um “jogo de cartas” na estimativa de tempo/esforço para implementação de cada item do backlog;
- Cada membro possui cartas com os números 2, 3, 5, 8 e 13;
- Para cada item, cada membro escolhe uma carta para estimar o tempo da tarefa (2 – muito rápida, 13 – muito demorada). Todos revelam suas cartas ao mesmo tempo;
- Caso dois membros tenham escolhido cartas muito distantes (2 e 8, 2 e 13 ou 3 e 13), eles devem explicar por que estimaram daquela forma e todos “jogam” suas cartas novamente;
- A estimativa para aquele item será a média dos valores.

Estimativa do tempo/esforço do projeto

Vamos praticar!

Equipe de desenvolvimento, já temos nosso backlog de produto (histórias de usuário), que tal estimarmos o tempo com o “poker do planejamento”?

	3
5	4,5
7,5	13
10	3



No início de cada ciclo...

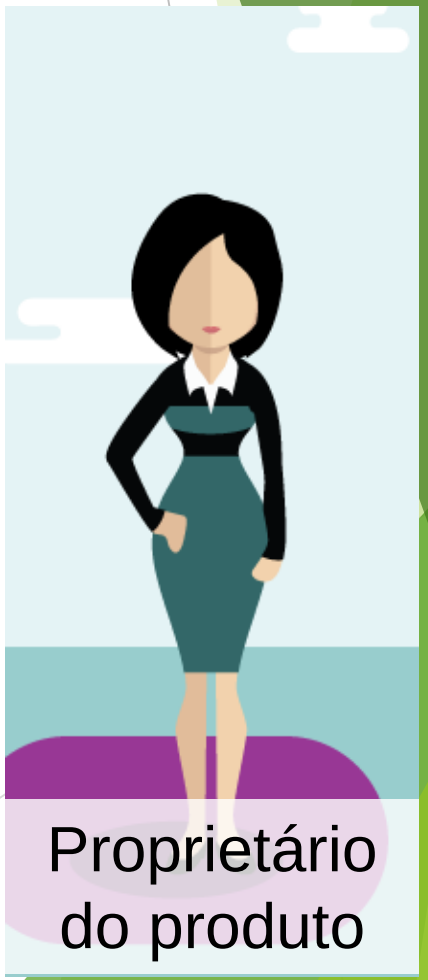
- Reunião de planejamento de ciclo (parte 1):
  - Envolve Proprietário do Produto, Mestre Scrum e Equipe;
  - Proprietário do Produto redefine a prioridade de cada item do backlog de produto;
  - Neste momento, o Proprietário do Produto pode incluir, alterar ou remover itens do backlog do produto.

Repriorização do backlog do produto

Vamos praticar!

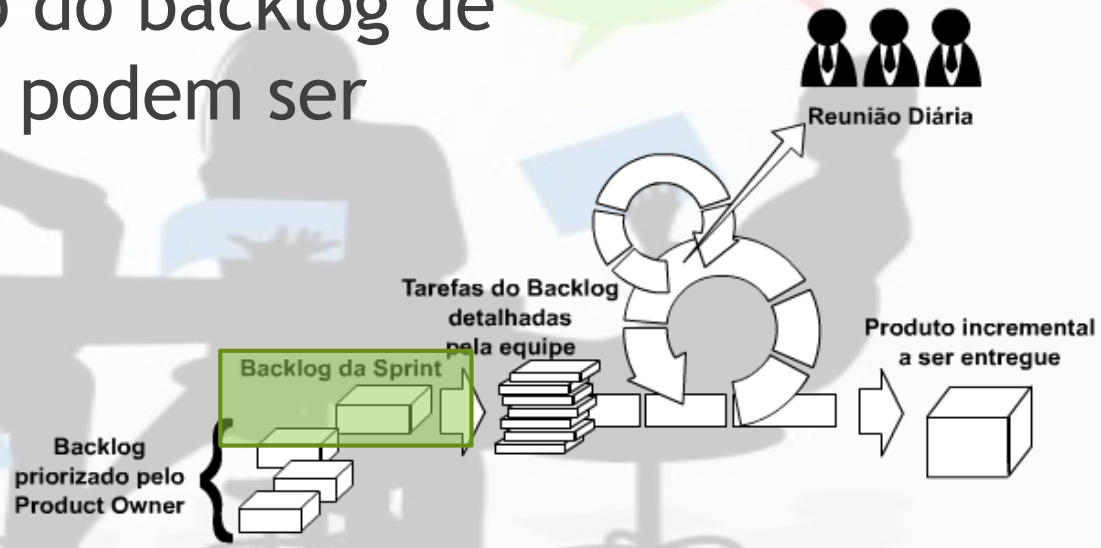
Proprietário do produto, priorize as 6 histórias de usuário de seu sistema.

	2
1	5
3	6
7	4



No início de cada ciclo...

- Reunião de planejamento de ciclo (parte 2):
  - Envolve Mestre Scrum e Equipe;
  - Equipe analisa os itens do topo do backlog de produto e escolhe aqueles que podem ser cumpridos no ciclo atual.

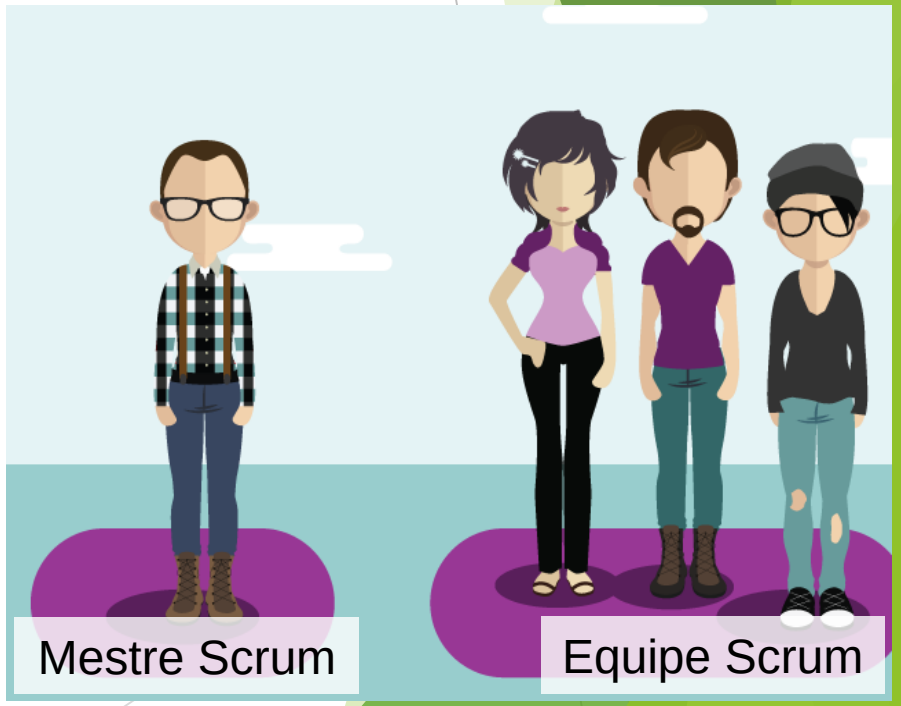


Elaboração do backlog do ciclo

Vamos praticar!

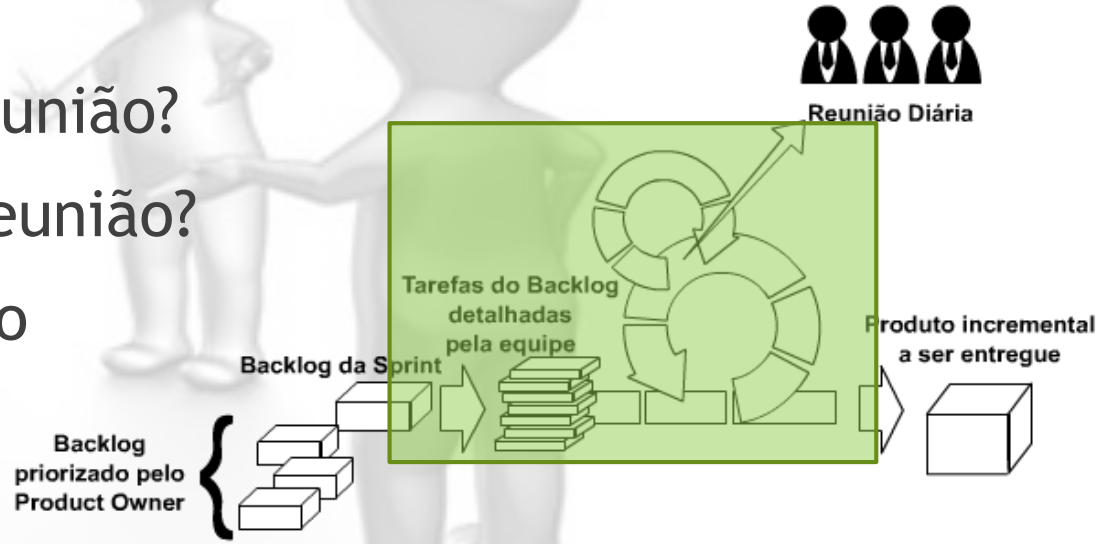
Equipe de desenvolvimento, escolha os itens que irão compor o backlog do ciclo.

	② 3 X
① 5 X	⑤ 4,5
③ 7,5	⑥ 13
⑦ 10	④ 3 X



Durante cada ciclo...

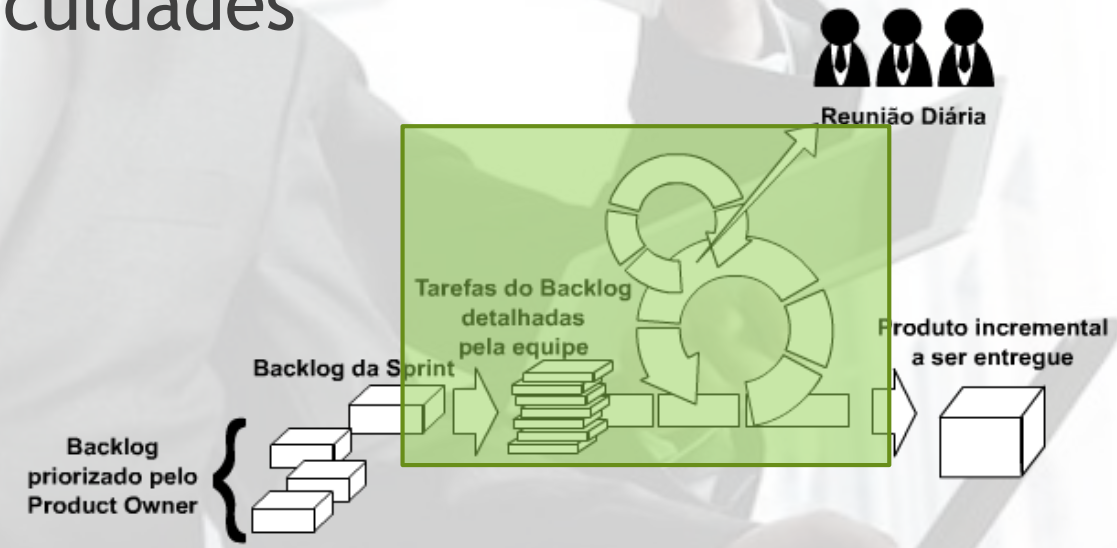
- Reuniões diárias:
  - Envolvem Mestre Scrum e Equipe;
  - Foco em três perguntas:
    - O que foi feito desde a última reunião?
    - O que será feito até a próxima reunião?
    - Quais obstáculos estão impedindo o seu avanço?



Acompanhamento do progresso do projeto

Durante cada ciclo...

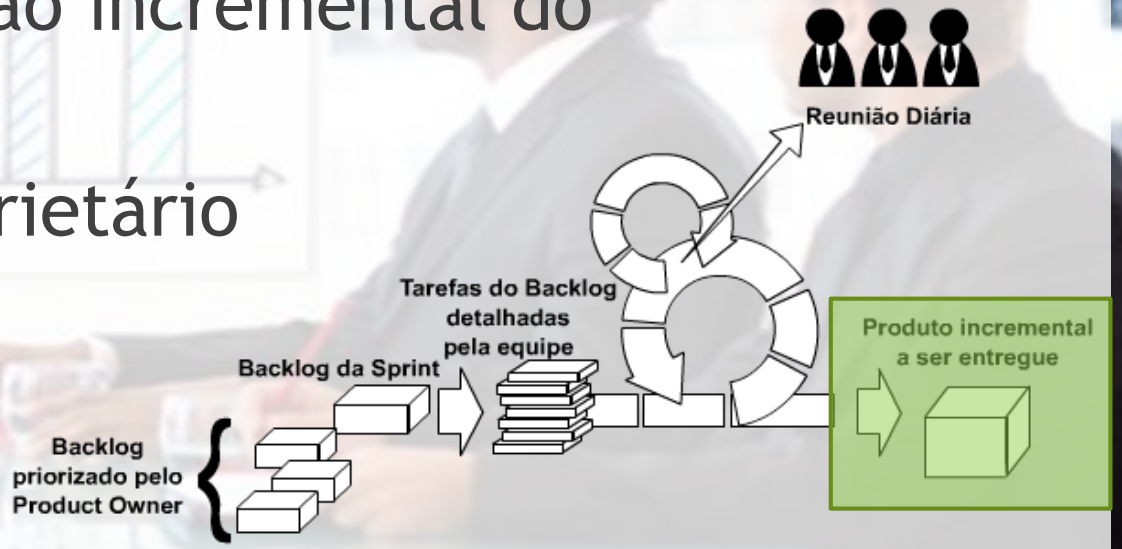
- Trabalha-trabalha-trabalha!
  - Equipe – execução do projeto;
  - Mestre Scrum – Solução de dificuldades e obstáculos.



Agora sim, a coisa está andando!

No final de cada ciclo...

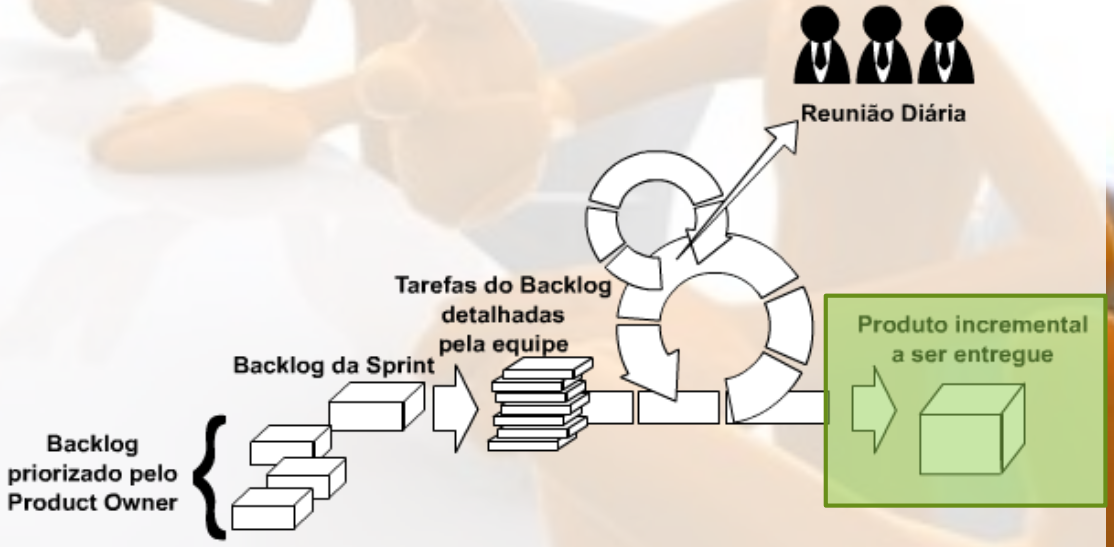
- Revisão do ciclo:
  - Envolve Proprietário do Produto, Mestre Scrum e Equipe;
  - Apresentação/entrega de versão incremental do produto;
  - Validação do mesmo pelo Proprietário do Produto.



Validação do produto

No final de cada ciclo...

- Retrospectiva do ciclo:
  - Envolve Mestre Scrum e Equipe;
  - O que ocorreu bem?
  - O que pode ser melhorado?



Compartilhamento do conhecimento

Ao final do projeto...

- Entrega final do produto gerado!

Resumindo...

Papéis

- Proprietário do produto;
- Mestre Scrum;
- Equipe Scrum.

Artefatos

- Backlog do produto;
- Backlog do ciclo;
- Versões incrementais do produto.

Reuniões

- Planejamento do ciclo;
- Diárias;
- Revisão do ciclo;
- Retrospectiva do ciclo.

Desafios ao Scrum

O que fazer quando o Proprietário do Produto solicita mudanças durante o ciclo?

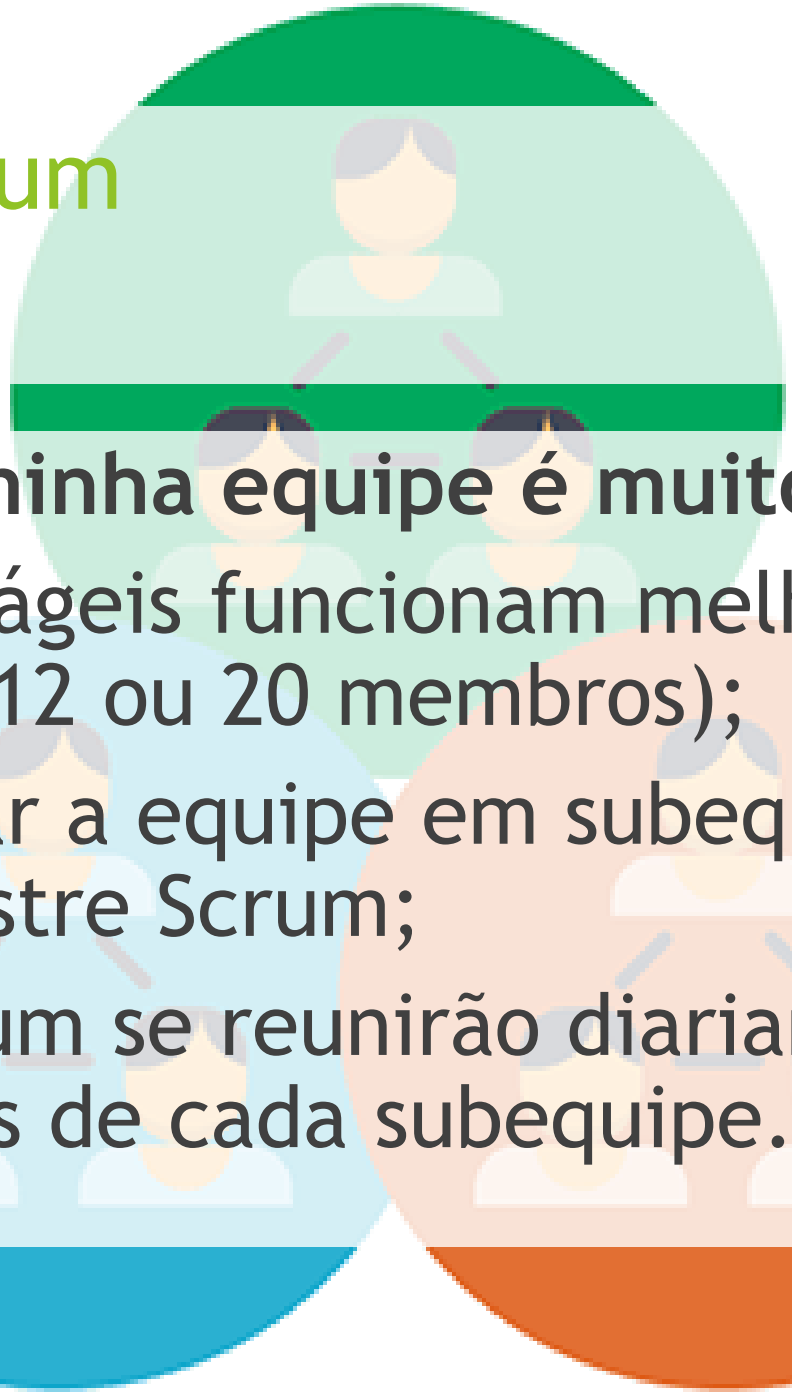
- Ele somente pode solicitar no início de cada ciclo;
- O ciclo atual pode ser cancelado, voltando-se para a reunião de planejamento de ciclo e recomeçando toda a contagem do ciclo.

Desafios ao Scrum

O que fazer se o Proprietário do Produto solicita custos e prazos "exatos" no início do projeto?

- Por seguir abordagem iterativa também no planejamento e modelagem da solução, geralmente não se conhece no início do projeto todos os custos e prazos envolvidos;
- Entretanto, podem-se adotar métodos e artefatos de outras abordagens de projeto para definição de custos e prazos – por exemplo, do PMBOK.

Desafios ao Scrum



O que fazer se minha equipe é muito grande?

- As abordagens ágeis funcionam melhor com equipes pequenas (até 12 ou 20 membros);
- Pode-se quebrar a equipe em subequipes, cada qual com seu próprio Mestre Scrum;
- Os Mestres Scrum se reunirão diariamente após as reuniões diárias de cada subequipe.

Desafios ao Scrum

O que fazer se o projeto for complexo demais?

- Empregue métodos e ferramentas de outras abordagens de gestão de projetos para:
  - Gestão de riscos;
  - Gestão da comunicação;
  - Gestão da qualidade etc.

Desafios ao Scrum

O que fazer se o nível de falhas exigido do produto final é muito pequeno (exemplo, nosso ônibus espacial)?

- Adote métodos rígidos para validação e verificação de todo o sistema ou de suas partes críticas (exemplo, métodos formais).

Dicas

- Baixe esta apresentação (e outras) em meu website:
  - http://christianosantos.com
- Procure aprender mais:
  - Livros (ex: Scrum – A arte de fazer o dobro do trabalho na metade do tempo);
  - Websites (ex: http://www.desenvolvimentoagil.com.br/scrum).
- Conheça outras ferramentas úteis (Kanban, PMBoK).
- **Ponha em prática.**

INSTITUTO FEDERAL DE SERGIPE
Campus Tobias Barreto

OBRIGADO

https://christianosantos.com

PELA ATENÇÃO