Scrum

Engenharia de Software II

Profa. Andréa Sabedra Bordin

História

- Em meados dos anos 80, Hirotaka Takeuchi e Ikujiro Nonaka definiram uma estratégia de desenvolvimento de produto flexível e completa onde a equipe de desenvolvimento **trabalha como uma unidade para alcançar um objetivo comum.**
- Eles descreveram uma abordagem inovadora para o desenvolvimento de produtos que eles chamaram de abordagem holística ou "*rugby*", onde uma equipe tenta alcançar a linha final como uma unidade.
- Eles basearam sua abordagem em estudos de casos de manufatura de várias indústrias.
- O desenvolvimento de um produto não deve ser como uma corrida de revezamento sequencial, mas sim deve ser análogo ao jogo de *rugby* onde a equipe trabalha em conjunto.

O **Scrum** é um método de reinício de jogada no rugby, onde os jogadores dos dois times se juntam com a cabeça abaixada e se empurram com o objetivo de ganhar a posse de bola.

O que é o Scrum?

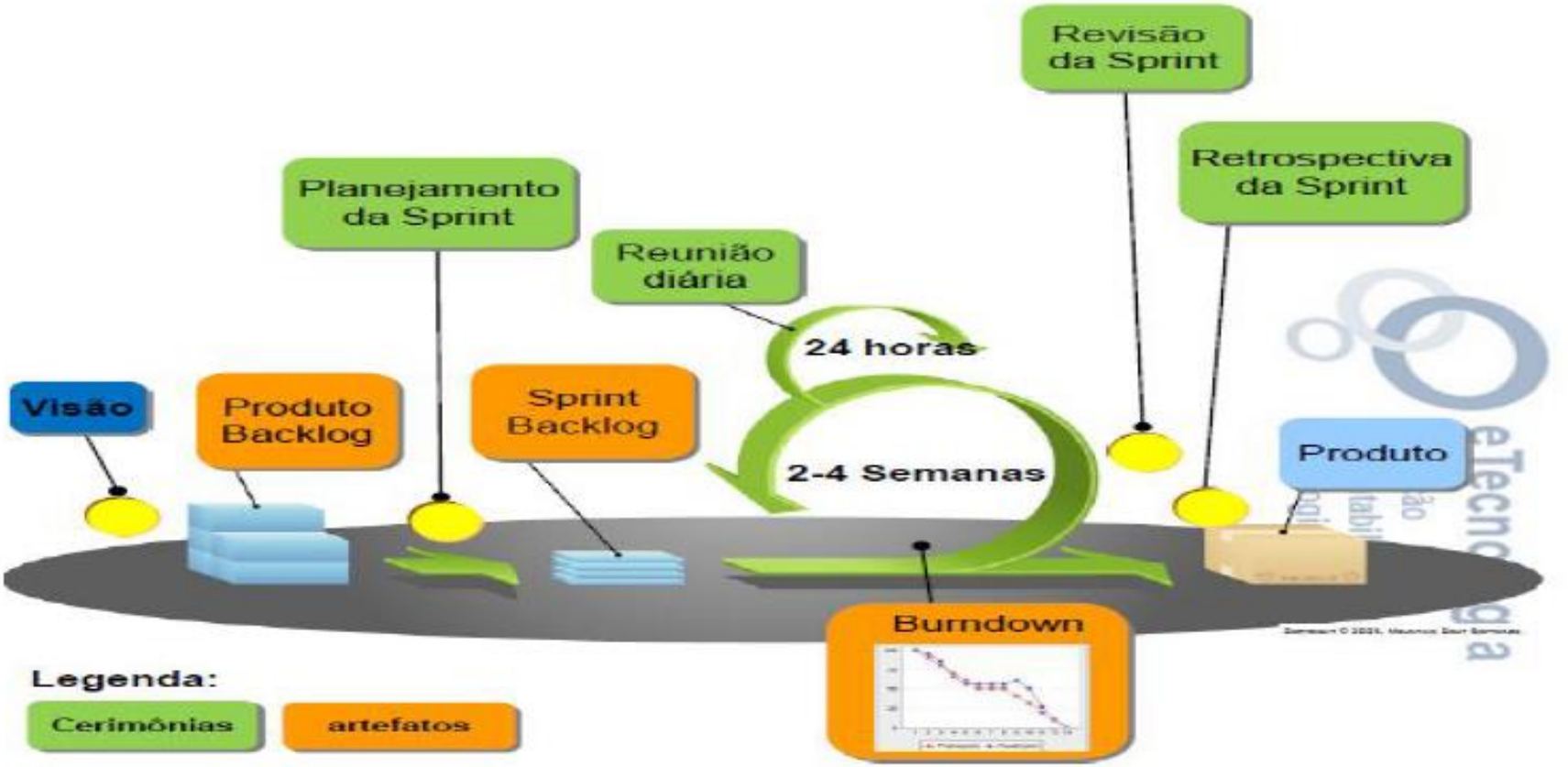
Scrum é um *framework* para organizar e **gerenciar** trabalhos complexos, tais como projetos de desenvolvimento de software.

Scrum é um processo **iterativo** e **incremental** para desenvolvimento de qualquer produto ou gerenciamento de qualquer trabalho.

O que é o Scrum?

- Uma alternativa de utilização de métodos ágeis na gerência de projetos.
- Pode ser aplicável a qualquer tipo de projeto.
- É simples.
- Processos, artefatos e regras são poucos e fáceis de entender.
- A simplicidade pode impressionar os acostumados com metodologias clássicas.

Framework Scrum



Papéis

- Product Owner (PO)
- ScrumMaster (SM)
- Equipe Scrum

Cerimônias

- Planejamento da Sprint
- Reunião Diária
- Revisão da Sprint
- Retrospectiva da Sprint

Artefatos

- Product Backlog
- Sprint Backlog
- Burndown (gráfico)

Fases

- Planejamento (pré-*game*)
- Desenvolvimento das *sprints* (*game*)
  - Reuniões Diárias
  - Revisão
  - Retrospectiva
- Encerramento (pós-*game*)

Fase de Planejamento

- **Criação do *backlog* do produto (*product backlog*).**
  - Lista de todas as funcionalidades desejadas.
    - Funcionalidade = história de usuário = *user story*
    - Escritas em linguagem natural.
  - Como no XP, cabe ao Dono do Produto (***Product Backlog - PO***) escrever as histórias dos usuários e, por isso, ele deve estar sempre disponível para tirar dúvidas do time.
- Atribuição de prioridades das histórias.
- Definição de equipes e seus líderes.
- Projeto da arquitetura do sistema.
- Estimativas de datas e custos.

História de usuário

É uma pequena descrição que detalha um item do *backlog* do produto.

<<Título da História>>

Como um <<ator>>
Eu <<quero/preciso/desejo>> <<objetivo da história>>
Para <<razão da história ser necessária>>

QUEM
O QUÊ
POR QUÊ

Como cliente, eu quero consultar os pagamentos realizados no Portal da Operadora para que possa controlar as minhas contas.

Como cliente, eu quero o imprimir a segunda via do boleto de pagamento pelo Portal da Operadora para que não tenha que ir a Operadora.

Como cliente, eu quero imprimir o relatório de comprovante de pagamentos pelo Portal da Operadora para que possa controlar as minhas contas.

Exemplo de Histórias de Usuário de um sistema do domínio Hoteleiro

Como **recepcionista** eu gostaria de **registrar a entrada de um hóspede** para **controlar a lotação do hotel**.
Como recepcionista eu gostaria de registrar a saída de um hóspede para controlar a lotação do hotel.
Como gerente eu preciso ter acesso ao mapa de reservas do hotel.
Como gerente eu preciso cadastrar os quartos disponíveis para reserva.
Como recepcionista eu gostaria de registrar o uso de um serviço na conta de um hóspede.

PRiORiDaDeS

1.

2.

3.

Product Backlog
Histórias de usuário com **prioridade**

prioridade	Descrição da história
alta	Como gerente eu preciso cadastrar os quartos disponíveis para reserva.
alta	Como recepcionista eu gostaria de registrar a entrada de um hóspede para controlar a lotação do hotel.
alta	Como recepcionista eu gostaria de registrar a saída de um hóspede para controlar a lotação do hotel.
média	Como recepcionista eu gostaria de registrar o uso de um serviço na conta de um hóspede.
baixa	Como gerente eu preciso ter acesso ao mapa de reservas do hotel.

O backlog do Produto é um artefato dinâmico, isto é, ele deve ser continuamente atualizado, de forma a refletir mudanças nos requisitos e na visão do produto

Fase de Desenvolvimento (*Sprints*)

- **Sprint** é o nome dado por Scrum para uma iteração.
- Como todo método ágil, Scrum é um método iterativo, no qual o desenvolvimento é dividido em sprints, de até um mês.
- Ao final de um sprint, deve-se entregar um produto com **valor tangível** para o cliente.
- O resultado de um sprint é chamado de um produto potencialmente pronto para entrar em produção (*potentially shippable product*).
    - O adjetivo potencial não torna a entrada em produção obrigatória.

Fase de Desenvolvimento (*Sprints*)

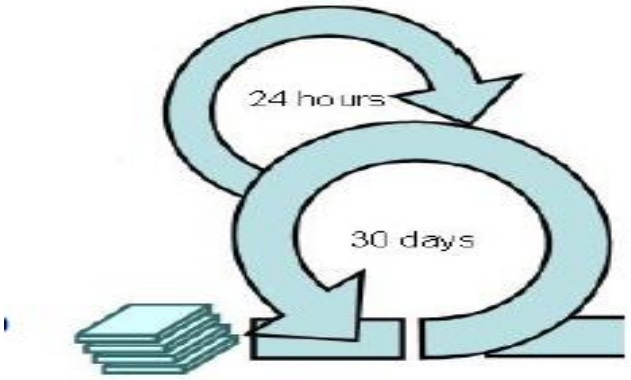
- Começa com o planejamento do *sprint*.
  - Reunião na qual todo o time se reúne para decidir as histórias que serão implementadas no *sprint* que vai se iniciar.
  - **É o evento que marca o início de um sprint.**
  - Essa reunião é dividida em duas partes:
  - **A primeira parte** é comandada pelo Dono do Produto.
    - Ele propõe histórias para o sprint e o restante do time decide se tem **velocidade** para implementá-las.
    - Neste momento a equipe discute os requisitos específicos que cada história de usuário requer.
  - **A segunda parte** é comandada pelos desenvolvedores.
    - Nela, eles **quebram as histórias em tarefas e estimam a duração delas.**
    - No entanto, o Dono do Produto deve continuar presente nessa parte final, para tirar dúvidas sobre as histórias selecionadas para o *sprint*.
      - Por exemplo, pode-se decidir cancelar uma história, pois ela se revelou mais complexa ao ser quebrada em tarefas.

Fase de Desenvolvimento (*Sprints*)

- O resultado do planejamento do *sprint* é chamado de ***sprint backlog*** (uma parte do *backlog*).
  - Lista com as tarefas do sprint, que inclui o esforço de desenvolvimento das mesmas.
- O *backlog* do *sprint* pode ser dinâmico.
  - Por exemplo, tarefas podem se mostrar desnecessárias e outras podem surgir, ao longo do *sprint*.
- O que não pode ser alterado é o objetivo do sprint (*sprint goal*)
  - A lista de histórias que o dono do produto selecionou para o sprint e que o time de desenvolvimento se comprometeu a implementar na duração do mesmo.

Fase de Desenvolvimento (*Sprints*)

- Scrum é um método adaptável a mudanças, mas desde que elas ocorram **entre** os *sprints*.
- Duração de 1 a 4 semanas (30 dias).
  - Este é o tempo necessário para produzir algo de interesse para o *Product Owner* e os *stakeholders*.
- Sempre apresenta um executável ao final.



Falando em estimativa....

Todas as histórias possuem o mesmo nível de complexidade ?

Estimativa de história de usuário

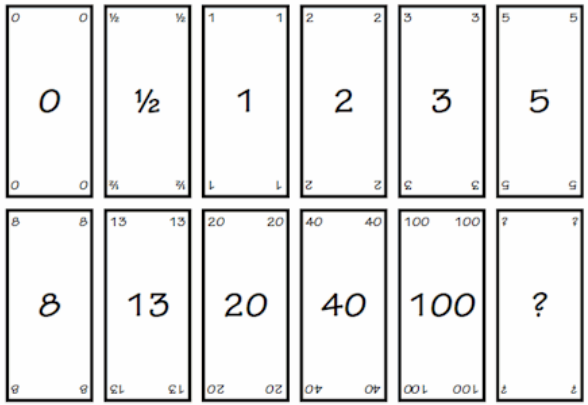
- **Pontos da história (*story points*)** são unidades de medida para expressar uma **estimativa do esforço** geral necessário para implementar as **histórias de usuário**.
- Equipes atribuem pontos de história em relação à complexidade de trabalho, à quantidade de trabalho e ao risco ou à incerteza.

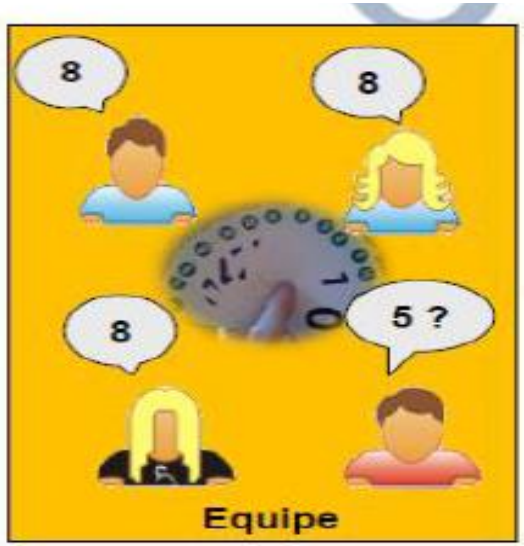
Estimativa de história de usuário
Técnica de *planning poker*

- Consiste-se na obtenção de estimativa através de um **jogo de cartas**.
- A ideia principal é permitir que todos os membros da equipe de desenvolvimento (programadores, testadores, design, analistas, etc.) participem colocando a sua visão de complexidade, levando em consideração o fator tempo e esforço para pontuar a estória e após juntos chegar a um denominador comum na equipe.
  - O *Product Owner* participa no planejamento, mas não estima.
- O *planning poker* pode ser considerado a combinação de três técnicas menos comuns de métodos de estimativas ágeis: opinião de especialista, analogia e desagregação.

Estimativa de história de usuário
Técnica de *planning poker*

- No *planning poker*, cada integrante tem a sua disposição um baralho de 12 cartas, numeradas numa sequência similar a encontrada nos números de **Fibonacci**.





Nota 1 – Estimativa*

https://www.mountaingoatsoftware.com/blog/why-the-fibonacci-sequence-works-well-for-estimating

Product Backlog
Histórias de usuário com **estimativa**

prioridade	descrição	Estimativa
alta	Como gerente eu preciso cadastrar os quartos disponíveis para reserva.	5
alta	Como recepcionista eu gostaria de registrar a entrada de um hóspede para controlar a lotação do hotel.	10
alta	Como recepcionista eu gostaria de registrar a saída de um hóspede para controlar a lotação do hotel.	10
média	Como recepcionista eu gostaria de registrar o uso de um serviço na conta de um hóspede.	7
baixa	Como gerente eu preciso ter acesso ao mapa de reservas do hotel.	4

Sprint Backlog

alta	Como gerente eu preciso cadastrar os quartos disponíveis para reserva.	5

Decompor a história em TAREFAS com estimação de horas

Tarefa	Estimativa de horas
Gerar protótipo	3
Criar/Alterar modelo de dados	4
Criar/alterar diagrama de classes	2
Desenvolver *front-end*	4
Desenvolver *back-end*	4

Não estimar muito tempo para uma tarefa facilita o acompanhamento do desenvolvimento.

Orientação: 1 a 16 horas

Sprint Backlog

Quantas histórias cabem em um *sprint backlog*?

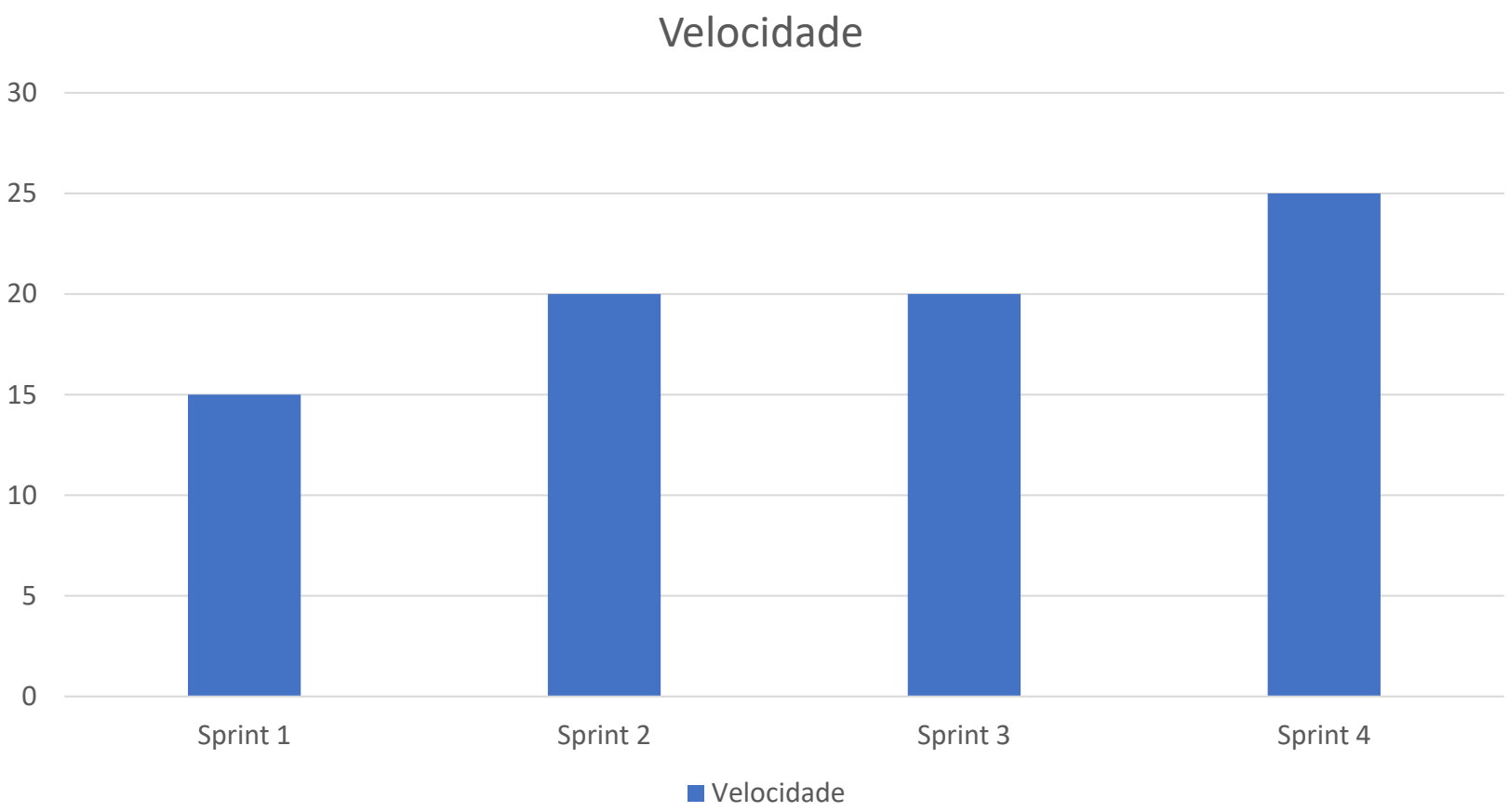
Sprint Backlog
Velocidade

prior	descrição	Estim
alta	Como gerente eu preciso cadastrar os quartos disponíveis para reserva.	5
alta	Como recepcionista eu gostaria de registrar a entrada de um hóspede para controlar a lotação do hotel.	10
alta	Como recepcionista eu gostaria de registrar a saída de um hóspede para controlar a lotação do hotel.	10
média	Como recepcionista eu gostaria de registrar o uso de um serviço na conta de um hóspede.	7
baixa	Como gerente eu preciso ter acesso ao mapa de reservas do hotel.	4

Sprint backlog inicial
5
10
10
Total de pontos: 25

Sprint backlog final
5 concluído
10 concluído
10 não iniciada
Total de pontos: 15

Sprint Backlog
Velocidade



Sprint Backlog
Estimativa pela capacidade da equipe

Tarefa	Estimativa de horas
Gerar protótipo	3
Criar/Alterar modelo de dados	4
Criar/alterar diagrama de classes	2
Desenvolver front-end	4
Desenvolver back-end	4

4 membros * 8 horas diárias

Sprint de 2 semanas = 10 dias

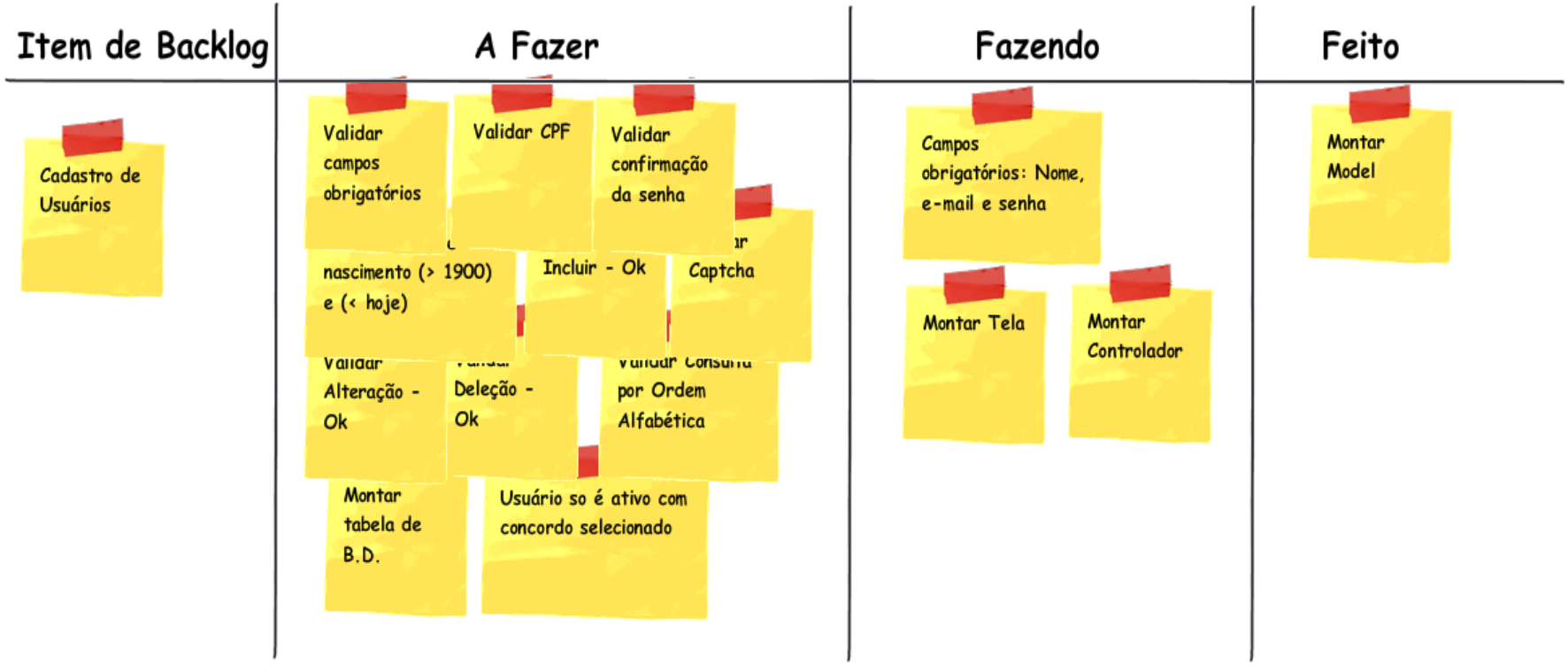
4 * 8 * 10 = 320 horas por *sprint*

Estime quantas tarefas (horas) cabem na capacidade de execução de um *sprint.*
80% é o recomendado

Sprint
REUNIÃO DIÁRIA (*SCRUM DAILY MEETING*)

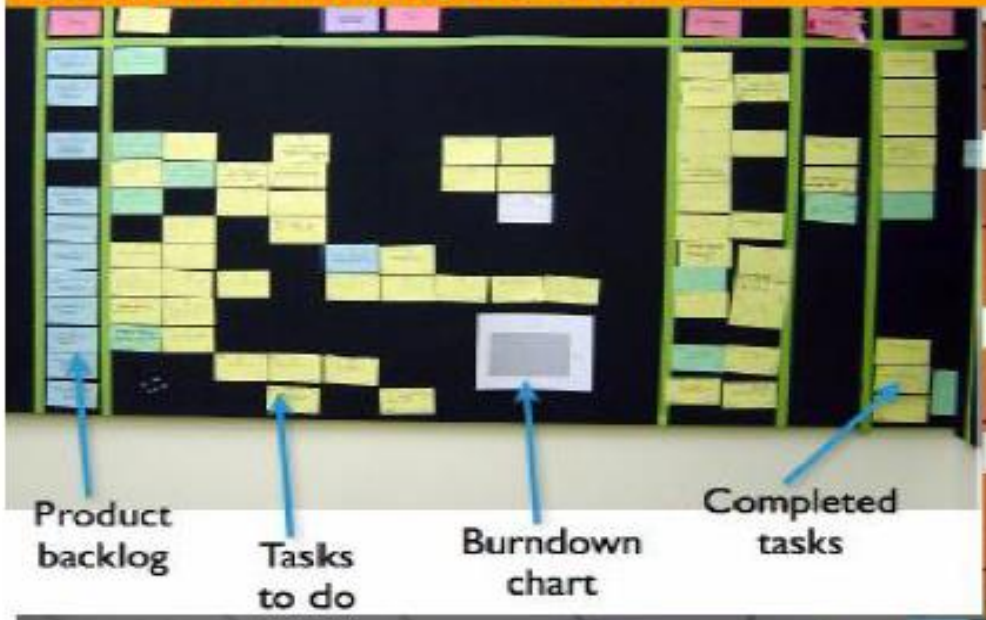
- Cerca de 15 minutos de duração.
- Todos respondem às perguntas:
  - O que você realizou desde a última reunião?
  - Quais problemas você enfrentou?
  - Em que você trabalhará até a próxima reunião?
- •Benefícios:
  - Maior integração entre os membros da equipe.
  - Rápida solução de problemas.
  - Promove o compartilhamento de conhecimentos.
  - Progresso medido continuamente.
  - Minimização de riscos.

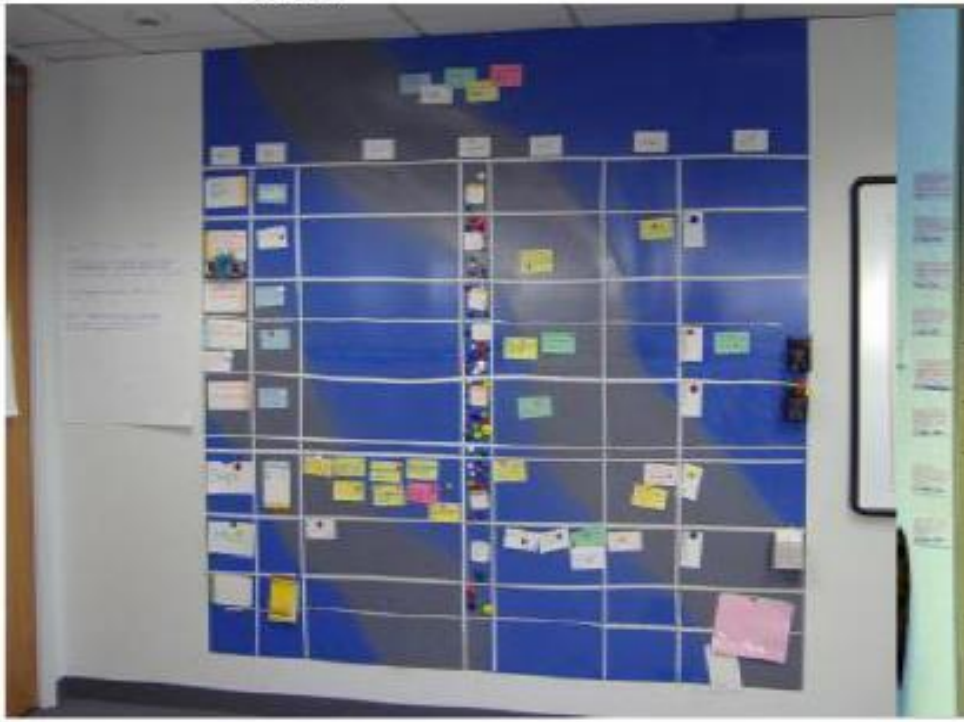
Acompanhamento do *sprint*

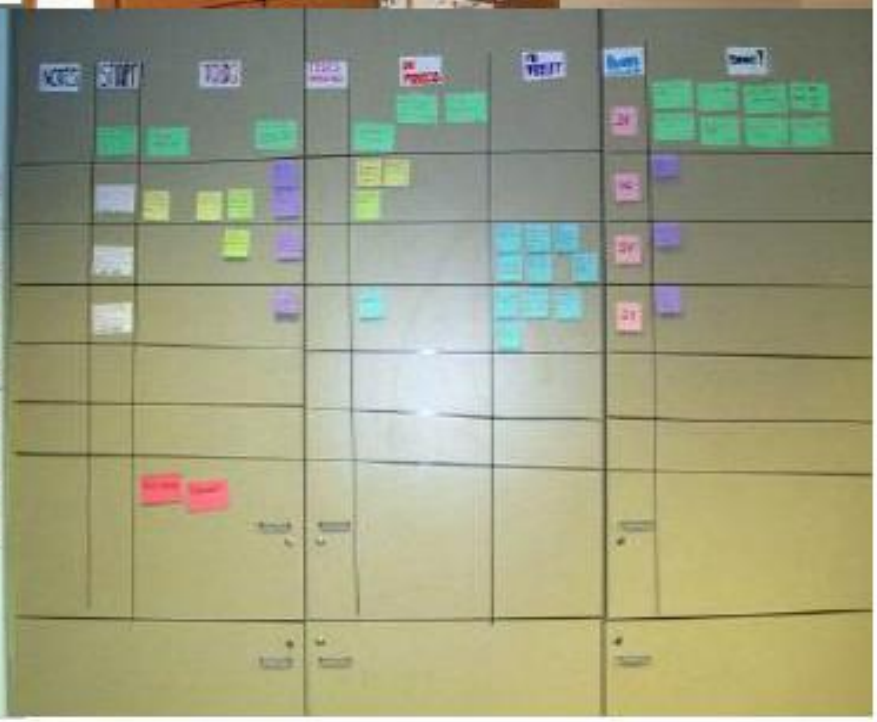


- Ao lado do Backlog do Sprint, costuma-se anexar um quadro com tarefas *a fazer*, *em andamento* e *finalizadas*.
- Esse quadro — também chamado de **Quadro Scrum** (*Scrum Board*) — pode ser fixado nas paredes do ambiente de trabalho, permitindo que o time tenha diariamente uma sensação visual sobre o andamento do sprint.

Gestão à Vista: Dá visibilidade e transparência ao desenvolvimento de software







Critérios para tarefa concluída

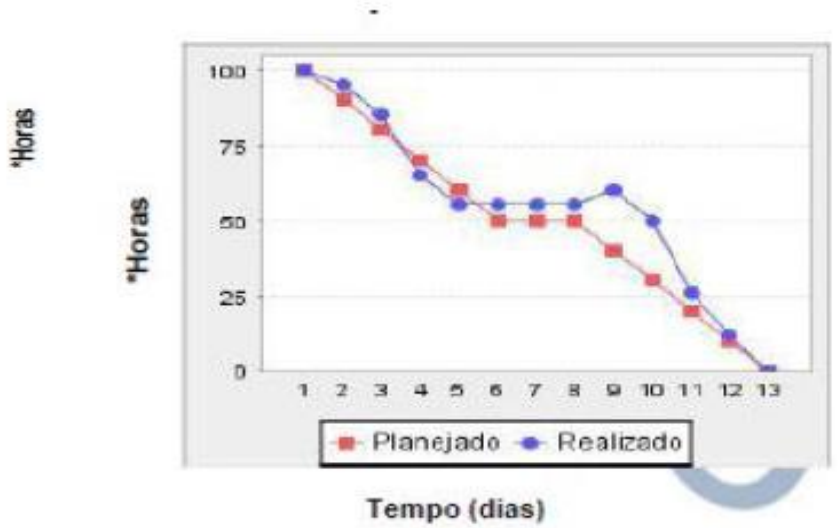
- Uma decisão importante em projetos Scrum envolve os critérios para considerar uma história ou tarefa como concluídas (*done*).
- Esses critérios devem ser combinados com o time e ser do conhecimento de todos os seus membros.
  - Por exemplo, em um projeto de desenvolvimento de software, para que uma história seja marcada como concluída pode-se exigir a implementação de testes de unidade, que devem estar todos passando, bem como a revisão do código por um outro membro do time.
  - Além disso, o código deve ter sido integrado com sucesso no repositório do projeto.
- O objetivo desses critérios é evitar que os membros — de forma apressada e valendo-se de código de baixa qualidade — consigam mover suas tarefas para a coluna concluído.

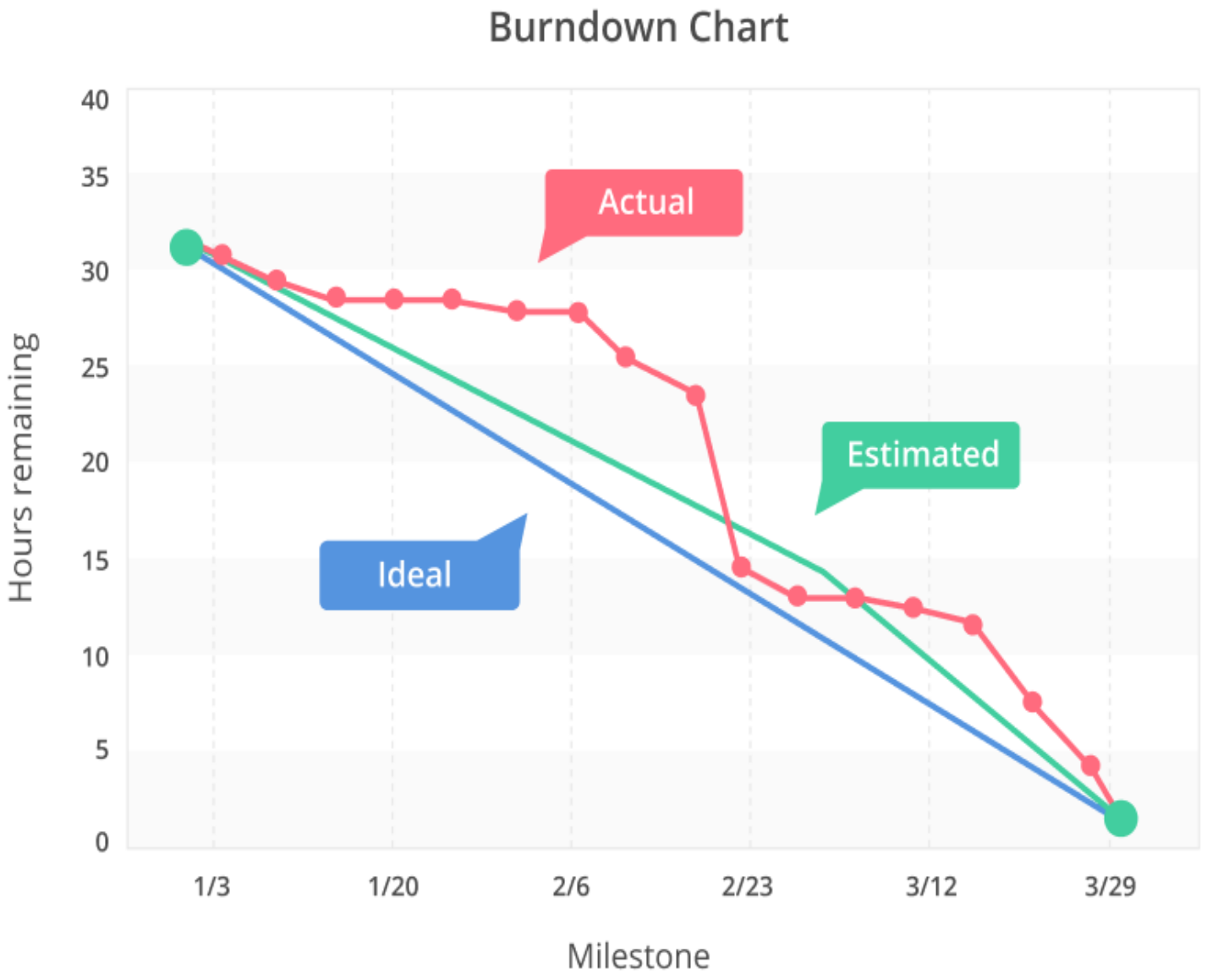
Acompanhamento do *sprint* Gráfico *Burndown*

O gráfico **Burndown** é a principal ferramenta de **gerenciamento do processo de desenvolvimento de software.**

Pois, ele representa o **trabalho restante sobre tempo**, ou seja, ele permite visualizar o progresso e/ou a evolução do trabalho executado pela a equipe e a quantidade trabalho x tempo (pontos) que ainda faltam para completar a Sprint.
Atualização do Burndown é diária, isto facilita a tomada de decisão, podemos decidir em melhorar a produtividade da equipe e/ou para mitigar risco da Sprint.

Exemplo:
Através da leitura do Burndown podemos decidir, que devemos adicionar novas tarefas na Sprint (velocidade da equipe está acima do planejado, melhorando sua produtividade) ou retirar tarefas (a velocidade da equipe está abaixo do planejado, caso não seja feita redução de tarefas a meta da Sprint estará comprometida).
O ideal, neste caso, é retirar as tarefas que não afetem a meta da Sprint.
Se a meta for afetada pode-se também decidir pelo cancelamento da Sprint.




Burndown Chart
Hours remaining
40
35
30
25
20
15
10
5
0
Actual
Estimated
Ideal
1/3
1/20
2/6
2/23
3/12
3/29
Milestone

Sprint
REUNIÃO DE RETROSPECTIVA

- Não deve levar mais do que 3 horas.
- Participam desta reunião:
  - Time, *Scrum Master* e, opcionalmente, *Product Owner*.
- Os membros do time devem responder a duas questões:
  - O que aconteceu de bom durante o último *Sprint*?
  - O que pode ser melhorado para o próximo *Sprint*?
- *Scrum Master* escreve as respostas e prioriza na ordem que deseja discutir as potenciais melhorias.
- *Scrum Master* nesta reunião tem o papel de fazer com que o time encontre melhores formas de aplicar o Scrum.

Fase de Encerramento

- Finalização do projeto
- Atividades:
  - Testes de integração
  - Testes de sistema
  - Documentação do usuário
  - Preparação de material de treinamento
  - Preparação de material de marketing

Papéis do *Scrum*

- Todas as responsabilidades de gerenciamento são divididas entre **três papéis**:
  - *Product Owner*
  - *Scrum Master*
  - Time

Papéis do *Scrum*
PRODUCT OWNER

- Responsável por apresentar os interesses de todos os *stakeholders.*
- Define fundamentos iniciais do projeto, objetivos e planos de *release.*
- Responsável pela lista de requisitos (*Product Backlog).*
- Certifica se as atividades com maior valor para o negócio são desenvolvidas primeiro.
- Priorização frequente das funcionalidades antes de cada iteração.

Papéis do Scrum
SCRUM MASTER

- Responsável pelo sucesso do Scrum.
- Ensina o Scrum para os envolvidos com o projeto.
- Implementa o Scrum na empresa de forma adaptada a sua cultura, para continuamente gerar benefícios.
- Certifica se cada pessoa envolvida está seguindo seus papéis e as regras do Scrum.

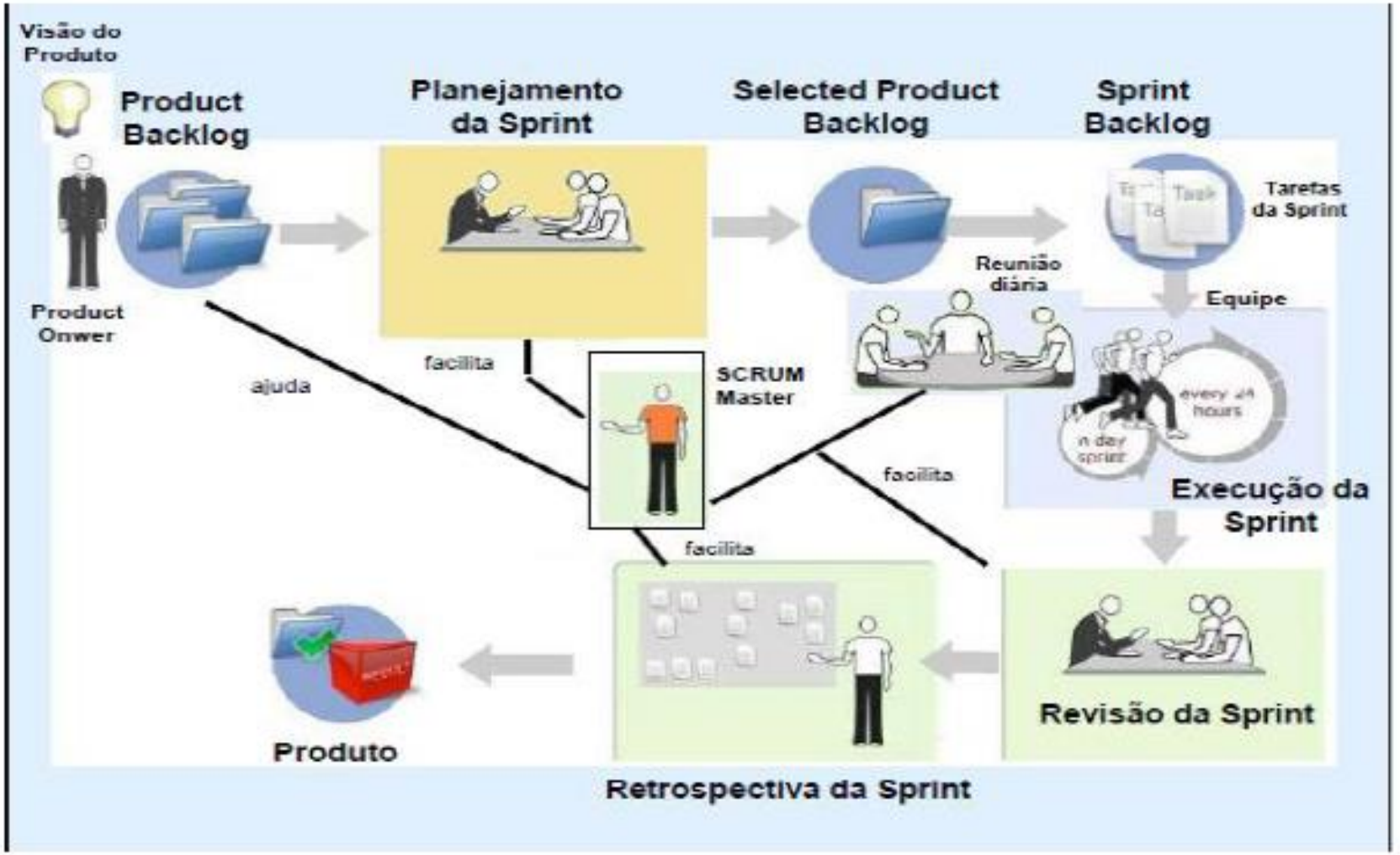
Papéis do Scrum
TIME OU EQUIPE

- 6 a 10 membros.
- múltiplos times podem ser formados para trabalhar em projetos maiores.
- Responsável por escolher as funcionalidades a serem desenvolvidas em cada interação e desenvolvê-las.
- O time se auto gerencia e auto organiza.
- Todos os membros do time são coletivamente responsáveis pelo sucesso de cada iteração.
- Responsabilidades do time durante o *Sprint*:
  - Participar das reuniões diárias do Scrum
  - Manter o *Sprint Backlog* atualizado
  - Disponibilizar o *Sprint Backlog* publicamente

Metáfora
Comprometidos x Envolvidos

A equipe Scrum é formado por pessoas "comprometidas" em realizar as tarefas da Sprint Backlog.

Roadmap Scrum



CERTIFICAÇÃO

- *Certified Scrum Master (CSM)*
- *Certified Scrum Product Owner (CSPO)*
- *....*

- https://www.scrumalliance.org/

Diferenças entre Scrum e XP

- XP é um método ágil voltado exclusivamente para projetos de desenvolvimento de software. Para isso, XP inclui um conjunto de práticas de programação, como testes de unidade, programação em pares, integração contínua e design incremental, que foram estudadas na seção anterior, dedicada a XP.
- Scrum é um método ágil para gerenciamento de projetos, que não necessariamente precisam ser projetos de desenvolvimento de software. Por exemplo, a escrita deste livro — como comentaremos daqui a pouco — é um projeto que está sendo realizado usando conceitos de Scrum. Tendo um foco mais amplo que XP, Scrum não propõe nenhuma prática de programação.

Referências

- https://www.scrumalliance.org/whyscrum/scrum-resources
- http://www.mindmaster.com.br/scrum
- A Guide to the SCRUM BODY OF KNOWLEDGE (SBOK™ Guide)
- SCRUM Experience. Tutorial disponível em:
- http://pt.slideshare.net/Ridlo/scrumexperience-o-tutorial-scrum
- Scrum – exemplo prático. Disponível em: https://www.youtube.com/watch?v=vpKlvPGaReI
- Aula da Profa. Fabiane Benitti
  - https://www.youtube.com/watch?v=vpKlvPGaReI
  - Minuto inicial: 6:50