

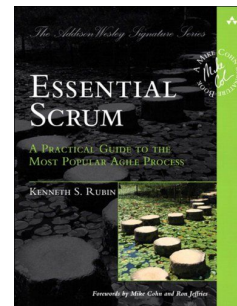
Scrum (+ Kanban e Lean)

Prof. Marco Tulio Valente

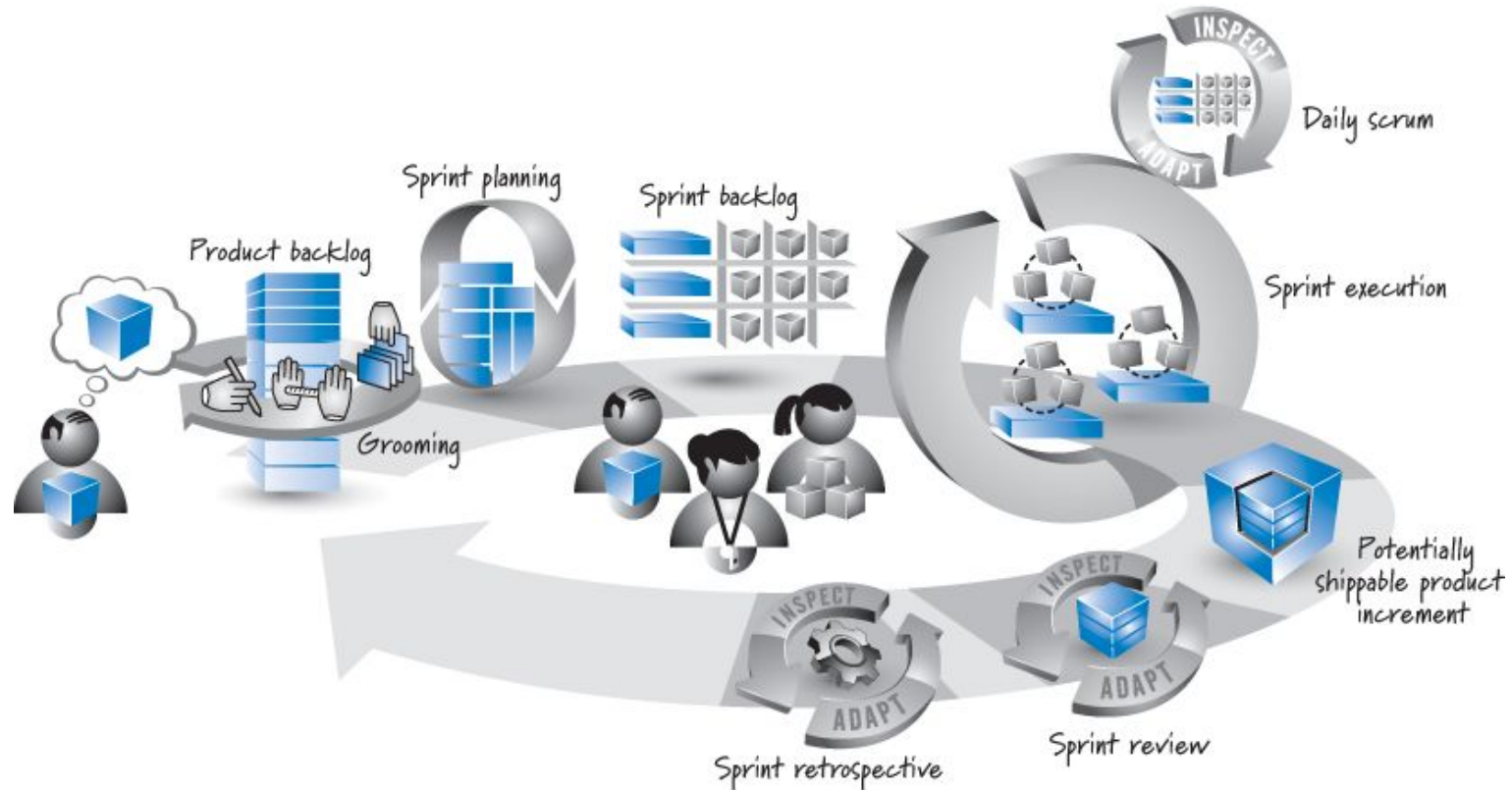
mtov@dcc.ufmg.br

Scrum

- Método ágil, iterativo e incremental para gerenciamento de projetos
- Não necessariamente apenas projetos de desenvolvimento de software
- Proposto por Jeffrey Sutherland e Ken Schwaber (em paper OOPSLA 1995)
- Hoje, Scrum é uma "indústria", com livros, cursos, consultores, certificações
- Dentre os métodos ágeis é o mais bem definido (e talvez mais "rígido")
- Neste curso, vamos ver um resumo de Scrum, usando o seguinte livro:
 - Essential Scrum, Kenneth Rubin, Addison-Wesley, 2012.
 - Todas as figuras usadas nos slides a seguir são deste livro

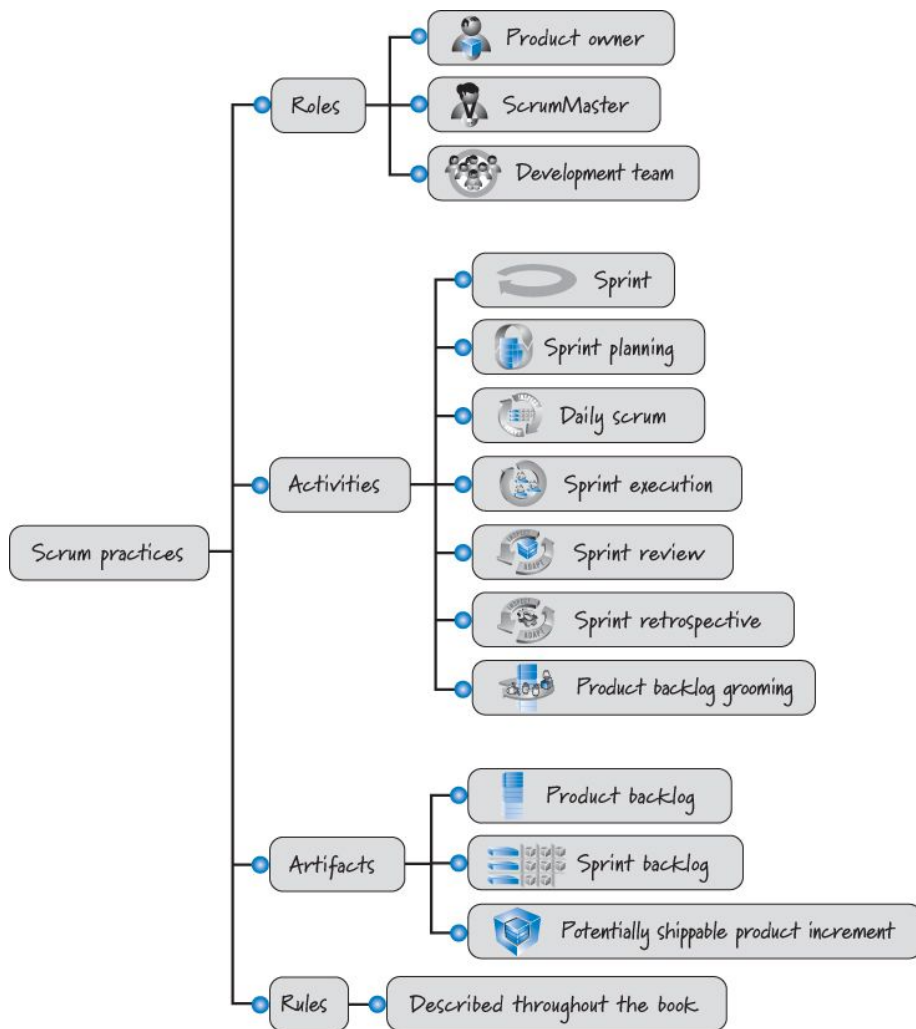


Scrum em 1 slide



Resumo das "Práticas Scrum"

Vamos estudar cada uma delas
no restante destes slides

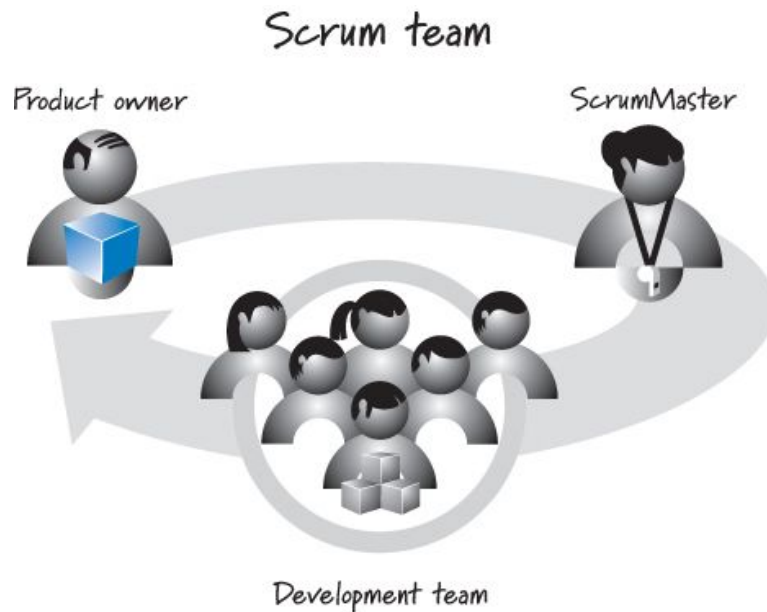


Duração das atividades é definida por um "time box"

- Todas as atividades em Scrum, incluindo sprints, reuniões etc, têm uma duração bem definida, a qual é chamada de **time-box** da atividade
- Por isso, o termo "time box" aparece sempre em documentos Scrum.
- Por exemplo, veja essa frase do Scrum Guide:
 - "The heart of Scrum is a Sprint, a time-box of one month or less during which a “Done”, useable, and potentially releasable product Increment is created."
- O objetivo da fixação de "time boxes" é criar um fluxo de trabalho, fomentar o compromisso da equipe e evitar a perda de foco

Times Scrum

- 1 Product Owner, 1 Scrum Master e de 5 a 9 desenvolvedores
- Isto é, "two pizza" teams, que possam ser alimentados por duas pizzas (que foi uma metáfora criada por Jeff Bezos, da Amazon)



Product Owner (PO)

- Representante dos clientes; possui a visão do produto
- Sua missão é maximizar o retorno do investimento feito no projeto
- Responsável por viabilizar que o time entenda os requisitos do projeto
- Deve estar sempre disponível para tirar dúvidas do time
- Responsável por escrever as histórias, normalmente no seguinte formato:

As a <type of user>

I want to <do_something>

so that <some value is created>

Como escolher um Product Owner?

- Caso #1: Desenvolvimento interno de um sistema para o depto X:
 - Product Owner deve ser um funcionário deste departamento, com grande conhecimento de seu funcionamento
- Caso #2: Desenvolvimento de um sistema para clientes externos
 - Product Owner pode ser um cliente externo (por exemplo, quando foi o cliente externo que contratou o desenvolvimento do sistema)
 - Ou Product Owner pode ser alguém da área de Marketing ou Vendas da organização que está desenvolvendo o software (isto é, alguém que vai representar os clientes reais).
- Em qualquer caso, não deve ser alguém da área de TI

Scrum Master

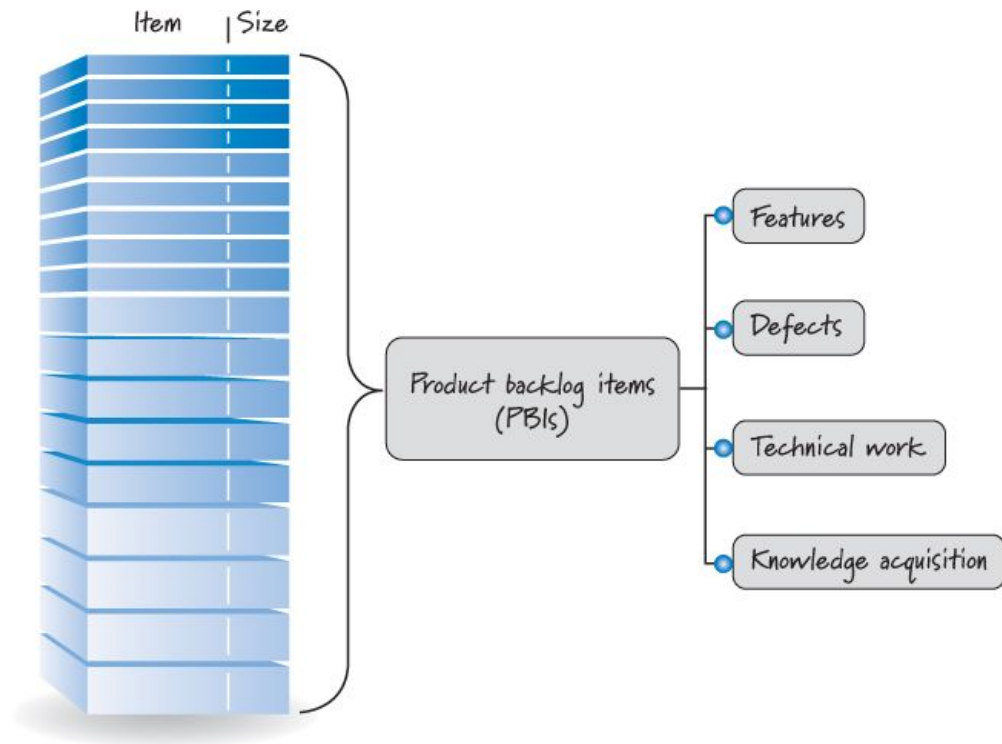
- É o "guia", "guardião" e "coach" do time
- É o facilitador dos trabalhos e removedor de "impedimentos"
- Deve ser um especialista em Scrum
- Mas não é um gerente de projeto tradicional; ou o "chefe" do time, pois todos em um time Scrum tem o mesmo nível hierárquico
- Por exemplo, ele não realiza tarefas de RH (autoriza folgas, idas ao médico)
- Mas deve garantir que as regras de processos Scrum estão sendo seguidas
- Se "sobrar tempo", pode ser o scrum master de mais de um time Scrum (mas nunca deve ser também Product Owner, junto com Scrum Master)

Membros do Time

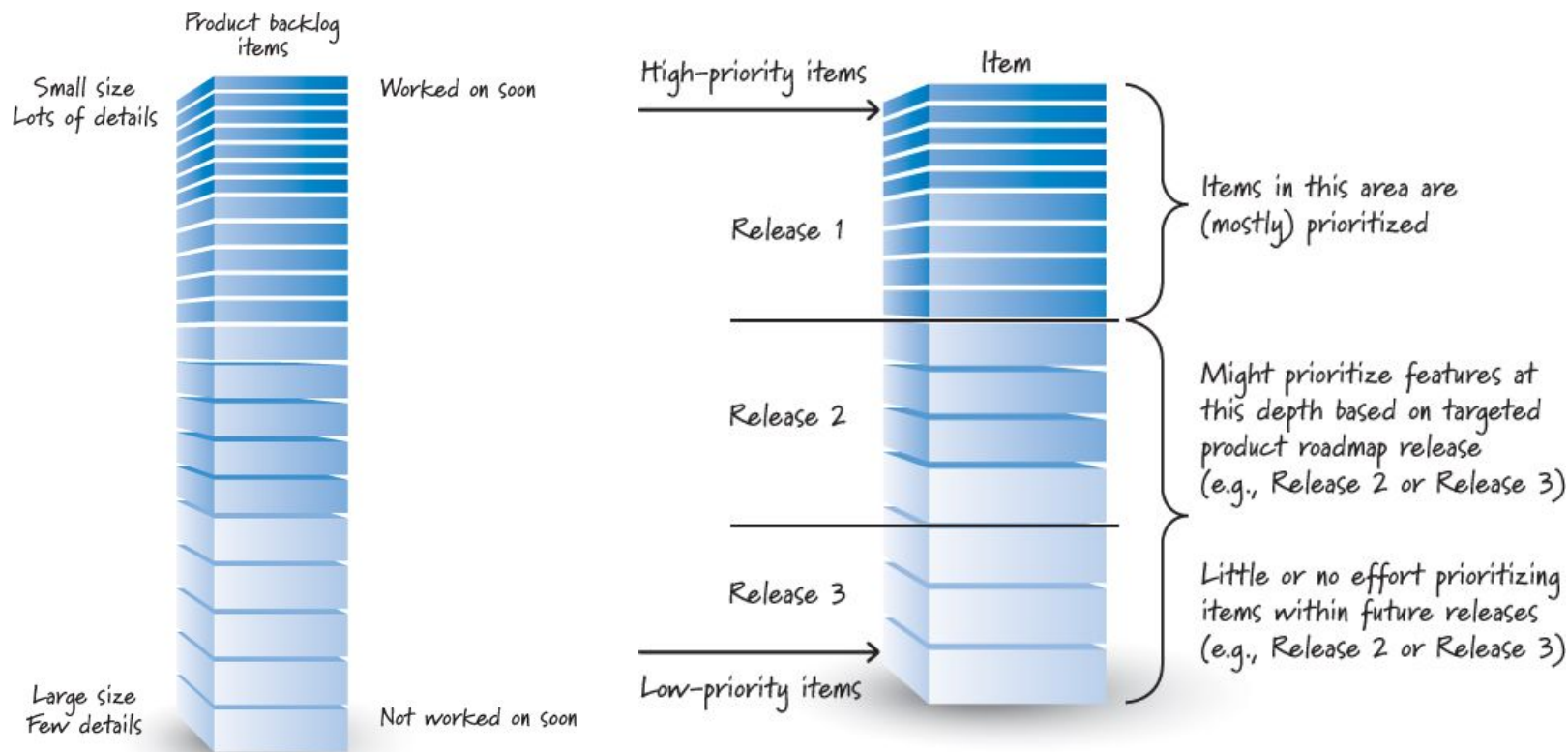
- No caso de projetos de software, são os programadores
- Times devem ser cross-funcionais, isto é, incluir todos os profissionais necessários para implementar o produto que está sendo definido pelo "product owner", por meio de histórias
- No caso de software: backend e/ou frontend developers, UI designers, etc
- Tomam todas as decisões técnicas (sobre linguagens de programação, arquitetura, frameworks etc que serão usados no sistema)
- Realizam as estimativas de tempo para implementação das user stories

Product Backlog

- Lista priorizada de requisitos que o sistema deve oferecer, incluindo features que restam para implementar, bem como defeitos que já devem ser corrigidos, tarefas técnicas (exemplo: atualizar uma API) e de aquisição de conhecimento (exemplo: estudar uma tecnologia)

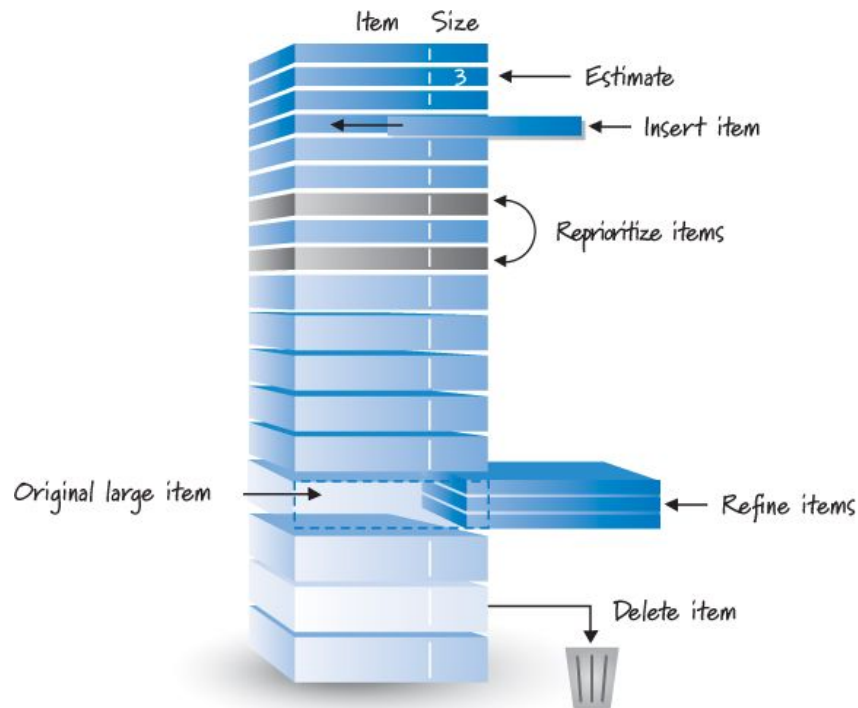


Organização do Product Backlog



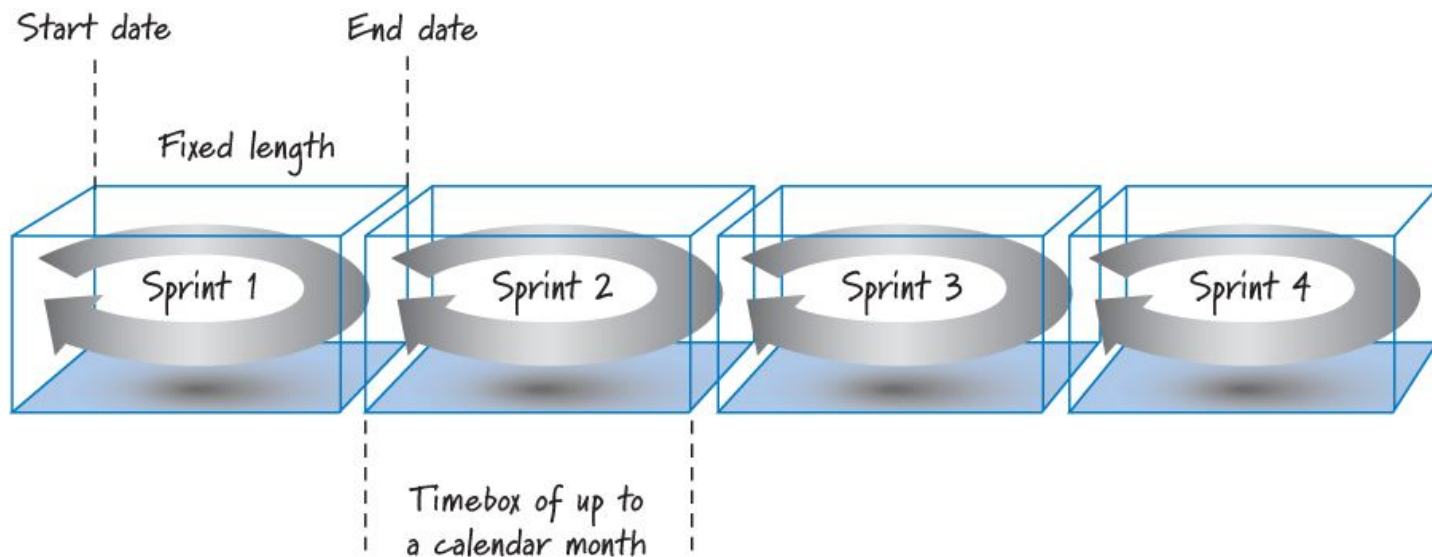
Grooming

- Tarefas de "cuidar" do product backlog, incluindo: (1) criar e refinar estórias; (2) estimar estórias; (3) priorizar estórias
- Quem realiza? Todos membros do time
- Quem "chefia"? O product owner
- Quando acontece? Scrum não especifica quando; pode ser semanalmente, diariamente etc



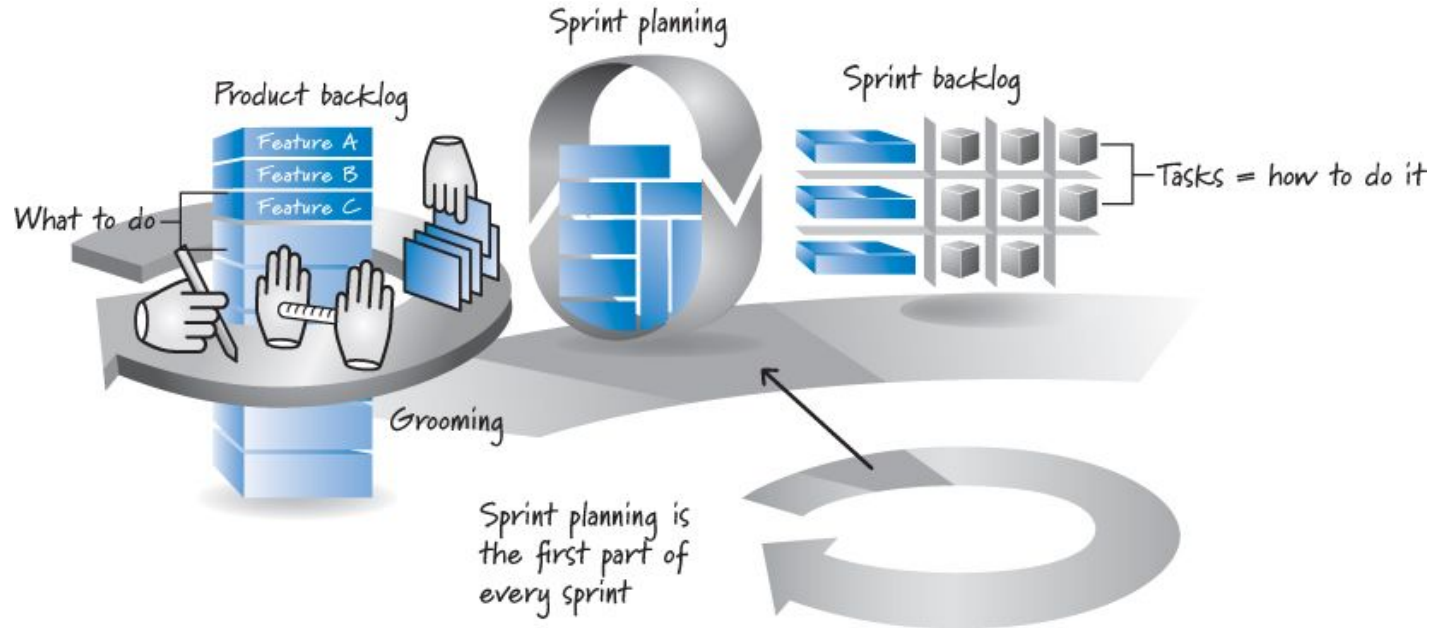
Sprint

- Scrum é um método iterativo; desenvolvimento é dividido em sprints, de até um mês; ao fim de um sprint, deve-se gerar um valor tangível para o cliente



Sprint Planning

- Planejamento realizado por todos os membros de um time para definir quais itens do "Product Backlog" serão implementados no próximo sprint



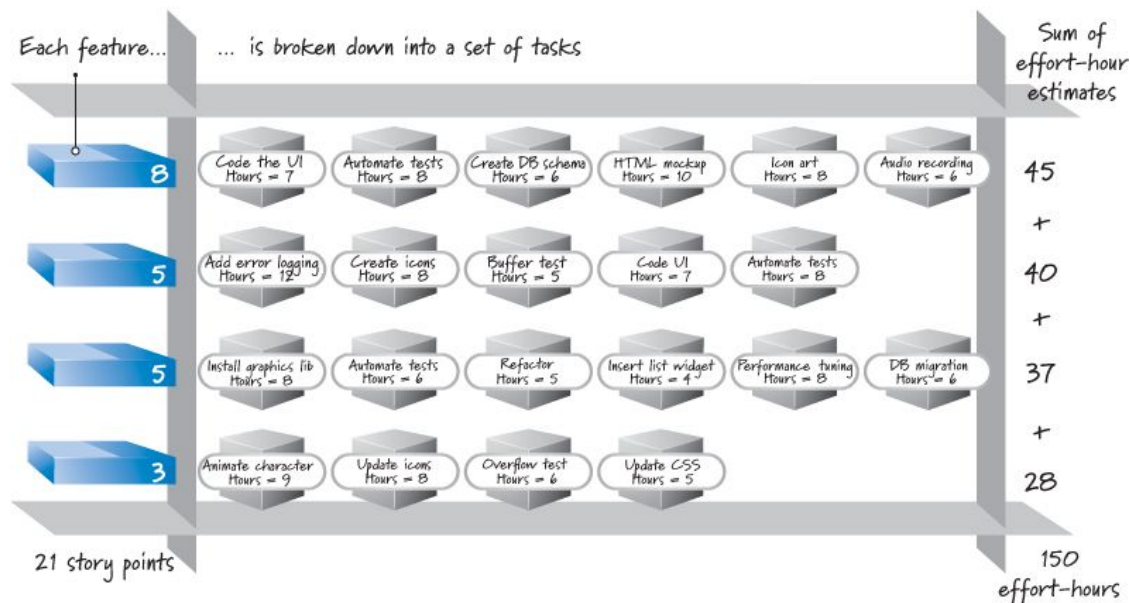
Com funciona um Sprint Planning?

- 1a parte da reunião: comandada pelo Product Owner
 - Product Owner: propõe histórias para o sprint
 - Time: decide se consegue implementá-las no intervalo do sprint
- 2a parte da reunião: comandada pelos membros do time
 - Estórias do sprint são quebradas em tarefas (pelos membros do time) e tem sua duração estimada
 - Product owner deve estar disponível para tirar dúvidas; talvez uma história possa ser até cancelada (pois ela se revelou mais complexa ao ser quebrada em tarefas)
- Resultado de um sprint planning é o **sprint backlog**

Sprint Backlog

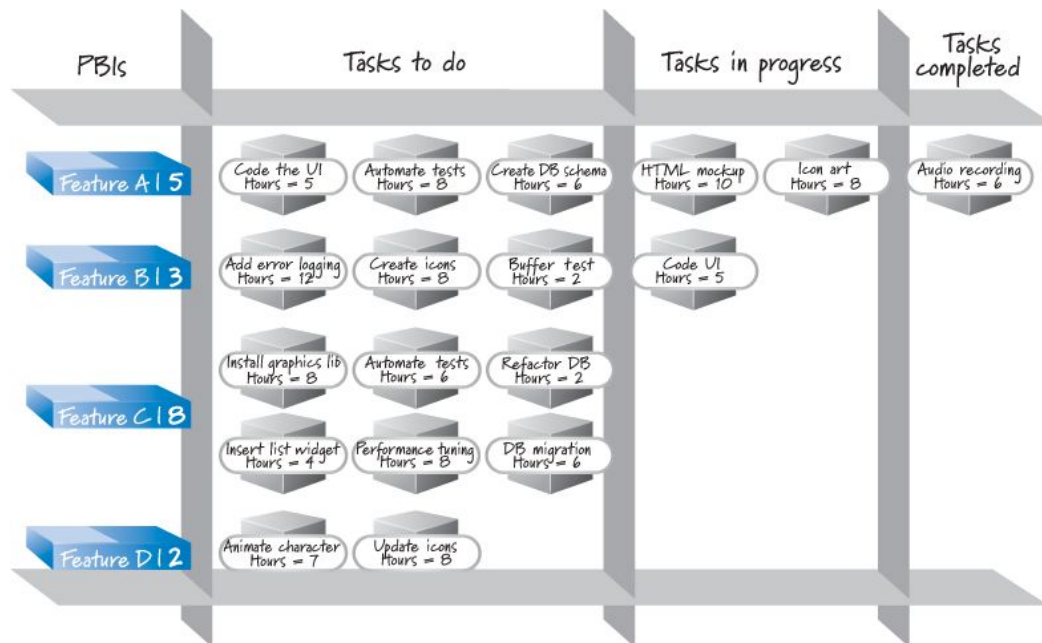
- Lista as tarefas de um sprint (cada item do "product backlog" é quebrado em algumas tarefas; essas tarefas vão para o "sprint backlog")

Sprint backlog



Task Board

- Quadro com tarefas "a fazer", "em andamento" e "finalizadas"
- Definição de "done" deve ser bem clara; às vezes, até anexada no task board



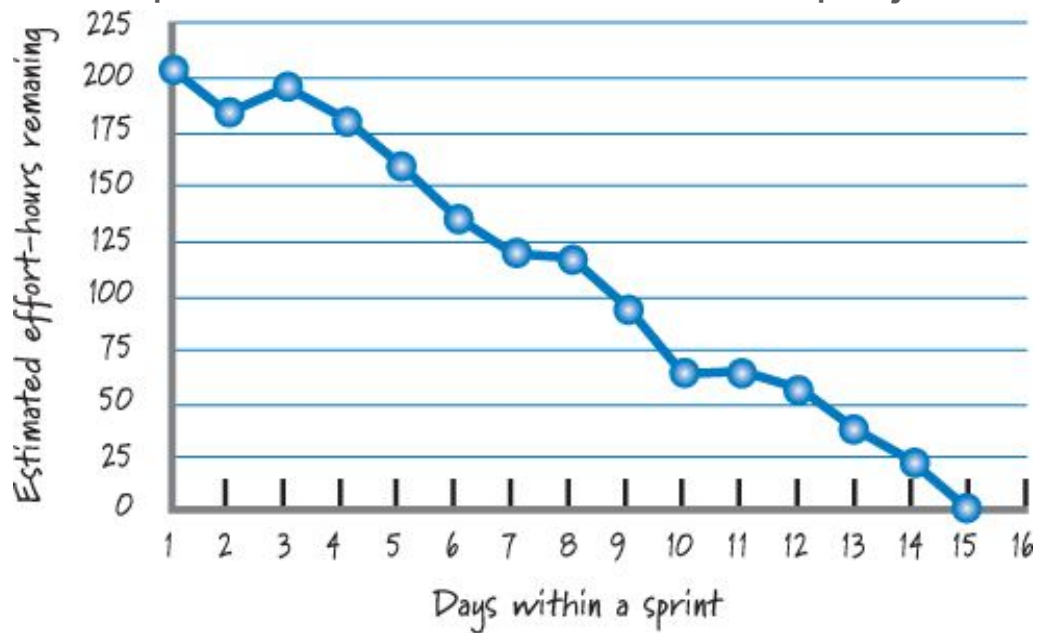
Exemplo de Task Board



Source: <https://hygger.io/blog/how-to-use-scrum-task-board>

Burndown Chart

- A cada dia do sprint, mostra qual o esforço (em horas) das tarefas "to do"; logo, o ideal é que zere no final do sprint (por exemplo, no dia 15)
- Isto é, no dia X do sprint ainda nos restam tarefas que juntas somam Y horas



Daily Scrum

- Reunião de 15 min, talvez em pé; uma das práticas mais discutidas de Scrum
- A sugestão é que cada participante responda: (1) o que fez ontem; (2) o que pretendo fazer hoje; (3) se for o caso, liste obstáculos
- Objetivos:
 - Sincronizar (e socializar) todos com o andamento do projeto
 - Antecipar problemas e "interferências" entre as tarefas
 - Antecipar "trabalho perdido" que está para acontecer
 - Criar um senso comum de responsabilidade com o projeto
- Problemas detectados na daily scrum devem ser discutidos e resolvidos fora dela, em reuniões menores

Exemplo de Daily Scrum

Mary: I fixed the missing-messages bug, I'm about to refactor the protocol to support faster logins

Jon: We should talk afterwards, I'm also touching the protocol to edit the login screen.

Dyneshwari: I did an unrelated UI change that doesn't interfere with anyone.

Shruthi: I just started laying the groundwork for ads in the stream.

Manager: We should talk afterwards, legal has a bunch of new requirements on .. ads.

Su: Nothing to report.

Grant: Oh, we moved all the test databases. You need to get the latest base data in your environment or you won't be able to work.

Joe: Jon, loop me in when you talk to Mary, I just had to change the protocol for UI widget X and I've got some tips for working with it.

Outro exemplo de Daily Scrum

- Veja também esse vídeo da IBM Agile Academy:



Sprint Review

- Reunião que marca o fim de um sprint, com participação inclusive de stakeholders envolvidos com o resultado do sprint
- Time demonstra os resultados do sprint para os clientes
- Deve durar de 30 a 60 minutos (dependendo da duração do sprint)
- Oportunidade para comunicar histórias que não foram implementadas
- Oportunidade também para receber feedback (mas ela não é uma reunião para tomada de decisões; ou para analisar "fracassos").

Retrospectiva

- Última atividade do sprint
- Reunião interna do time scrum, com duração de 1-2 horas
- Objetivo: refletir sobre o sprint, com o objetivo de melhorar o processo
- Não é uma reunião para "lavar a roupa suja"
- Resultado: 1-2 pontos de melhoria do processo, que idealmente possam ser aplicados a partir do próximo sprint
 - Exemplo 1: os membros não podem se atrasar, para a daily scrum
 - Exemplo 2: temos que respeitar as regras de layout do código
- Depois desta reunião, o ciclo se repete!

Exercício - ENADE 2017 (Ciência da Computação)

Os métodos ágeis são fundamentados no desenvolvimento e entrega incremental tendo em vista atender aos requisitos dos clientes. Eles agregam um conjunto de princípios provenientes do manifesto ágil, tais como:

- envolvimento do cliente;
- entrega incremental;
- pessoas, não processos;
- aceitação das mudanças;
- manutenção da simplicidade.

O *Scrum* é um exemplo de método ágil de gerenciamento de projetos. Avalie as afirmações a seguir sobre a relação do *Scrum* com os princípios do manifesto ágil.

- I. O *Scrum* adota a entrega incremental por meio de *Sprints*.
- II. O *Scrum* adota a simplicidade por meio do uso da programação em pares.
- III. O *Scrum* adota o envolvimento do cliente com a priorização e a negociação dos requisitos na concepção de *Sprints*.

É correto o que se afirma em

- A** II, apenas.
- B** III, apenas.
- C** I e II, apenas.
- D** I e III, apenas.
- E** I, II e III.

ENADE 2017 - ENADE 2017 (Ciência da Computação)

Os métodos ágeis são fundamentados no desenvolvimento e entrega incremental tendo em vista atender aos requisitos dos clientes. Eles agregam um conjunto de princípios provenientes do manifesto ágil, tais como:

- envolvimento do cliente;
- entrega incremental;
- pessoas, não processos;
- aceitação das mudanças;
- manutenção da simplicidade.

O *Scrum* é um exemplo de método ágil de gerenciamento de projetos. Avalie as afirmações a seguir sobre a relação do *Scrum* com os princípios do manifesto ágil.

- I. O *Scrum* adota a entrega incremental por meio de *Sprints*.
- II. O *Scrum* adota a simplicidade por meio do uso da programação em pares.
- III. O *Scrum* adota o envolvimento do cliente com a priorização e a negociação dos requisitos na concepção de *Sprints*.

É correto o que se afirma em

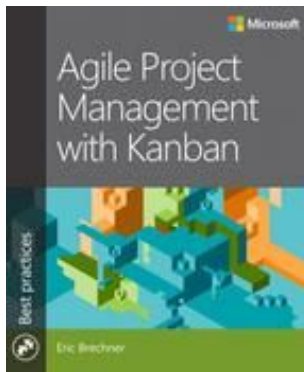
- ☐ A II, apenas.
- ☐ B III, apenas.
- ☐ C I e II, apenas.
- ☒ D I e III, apenas.
- ☐ E I, II e III.

Outros Métodos Ágeis

Kanban

Kanban

- Método antigo para controle de produção, proposto pela Toyota
- Mais recentemente, foi adaptado para desenvolvimento de software
- Neste curso, vamos ver a adaptação de Kanban proposta no seguinte livro:
 - Agile Project Management with Kanban, Eric Brechner
 - Engenheiro da Microsoft, com experiência em times do Xbox etc



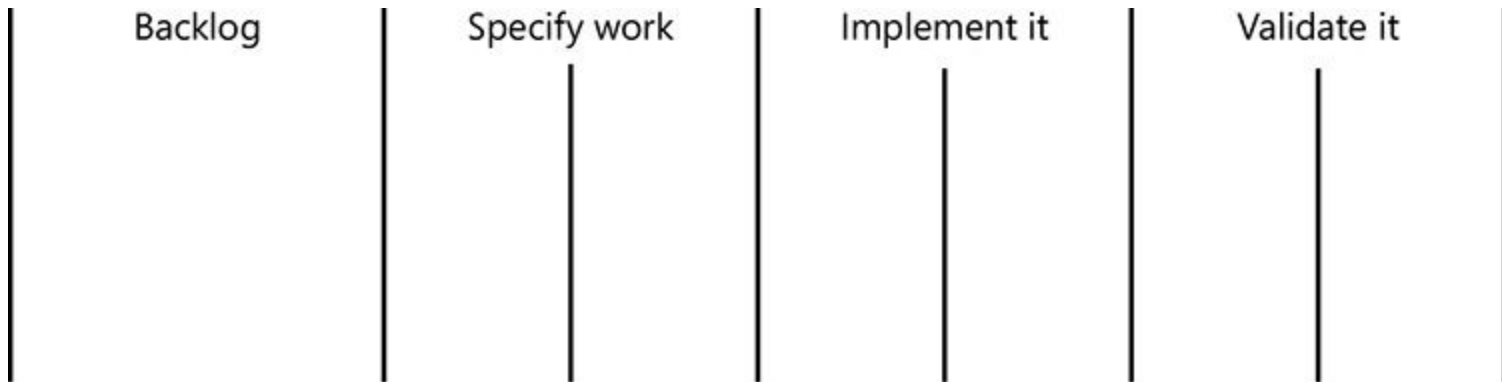
Kanban

- Kanban = Scrum - sprints + "quota" de tarefas em cada estágio
- Também não tem scrum master, product owner, pelo menos explicitamente
- Componente mais importante: kanban board (que inclui backlog)
- É como se existisse um único sprint, que nunca termina.



Kanban Board (ou Sign Board)

- Com exceção do Backlog, demais fases possuem duas colunas:
 - 1a coluna: itens em andamento na fase (ex.: em implementação)
 - 2a coluna: itens que já terminaram essa fase, mas que ainda não foram movidos para a fase seguinte (ex.: implementados e aguardando entrarem em validação; é uma espécie de buffer para a fase seguinte)
- Kanban é um "pull system"



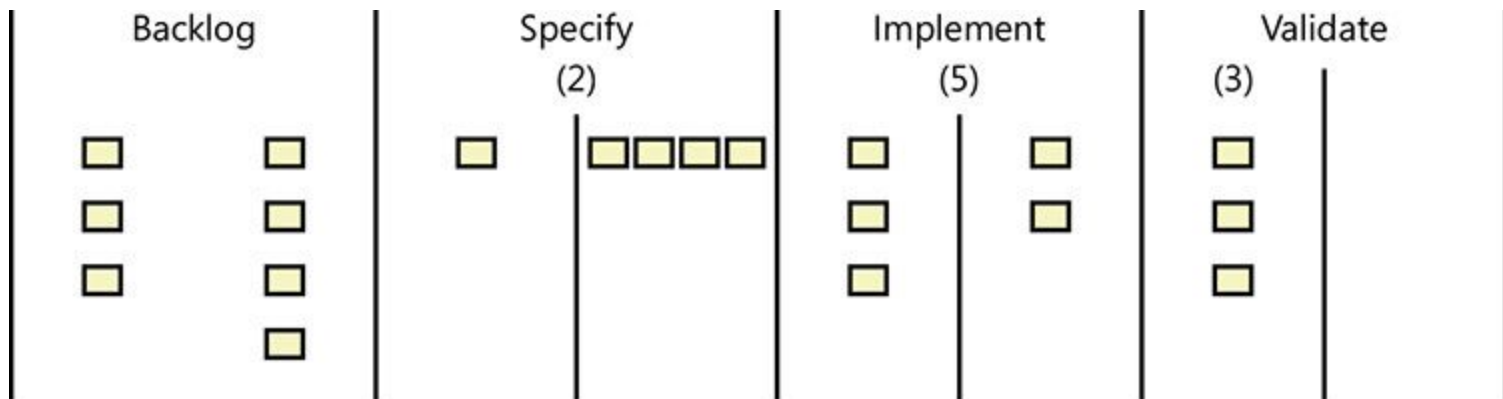
Kanban Board

- Valoriza-se um quadro real, com "post-it" cards, em vez de ferramenta online
- Motivo: permitir que todos possam "visualizar" o trabalho do time



WIP Limits

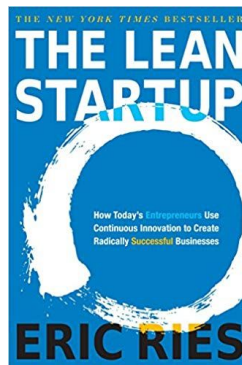
- Números abaixo de cada fase são chamados de WIP (working in progress) limits, isto é, máximo de itens em cada fase (em andamento e concluídos)
- "Specify" é um caso especial: no exemplo abaixo, existem dois itens (logo, respeitando o WIP); porém, um deles foi já quebrado em quatro



Lean Startup

Lean Startup

- Método "ágil" para gerenciamento de startups, com ciclos curtos e feedback
- Proposto por Eric Ries, em um livro com o mesmo nome (bastante famoso)
- Boa parte das startups são empresas de software; logo, muitas dos conceitos de "lean startup" valem para gerenciamento de desenvolvimento de software.



Lean Startup: Principais Práticas

- **MVP (Minimum Viable Product):** produto que permite coletar o máximo de feedback dos usuários, com o menor esforço; não espere pelo produto ideal
- **Feedback loop:** [build → measure → learn]*
- **Continuous deployment:** todo software produzido deve entrar rapidamente em produção (para receber feedback)
- **Testes A/B:** diferentes features são oferecidas para grupos de clientes, para testar sua aceitação; feature de maior sucesso é mantida no produto
- **Pivot:** correção de rumo; para testar uma outra hipótese, isto é, um outro tipo de produto e/ou cliente. Exemplo: Twitter nasceu como um sistema para compartilhamento de podcasts (nessa época, chamava-se Odeo).

Leituras e Videos Complementares

- The Scrum Guide, Ken Schwaber and Jeff Sutherland
 - Documento oficial que descreve Scrum de forma resumida (16 páginas)
 - <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- Video: Spotify Engineering Culture
 - <https://youtu.be/R2o-Xm3UVjs>
- Palestras e livros do Jeff Sutherland
 - Talks at Google: <https://youtu.be/2L1oBLTICx4>
 - TED Talk: <https://youtu.be/s4thQcgLCqk>
 - Scrum: The Art of Doing Twice the Work in Half the Time (livro)

Leituras e Videos Complementares

- Video: "Agile Project Management with Kanban", Eric Brechner
 - <https://youtu.be/CD0y-aU1sXo>
- Livro: Sprint How to Solve Big Problems and Test New Ideas in Just Five Days, Jake Knapp