Scrum

Introdução ao framework

Ruhan de Oliveira Baiense

arcoinformatica.com.br

Introdução aos Métodos Ágeis

- Agile é uma metodologia.
- Métodos ágeis focam em pouca documentação, e mais produtividade.
- Em 2001, um grupo de programadores lançou o Manifesto Ágil, uma metodologia que tem como objetivo satisfazer os clientes ou envolvidos em um projeto entregando com rapidez e, com maior frequência, versões do projeto, conforme as necessidades.

Manifesto Ágil

- Indivíduos e interações mais que processos e ferramentas
- Software em funcionamento mais que documentação abrangente
- Colaboração com o cliente mais que negociação de contratos
- Responder a mudanças mais que seguir um plano

> *...mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.*

12 Princípios ageis

1. Satisfazer o cliente.
2. Dar boas-vindas às mudanças.
3. Entregar com frequência resultados.
4. Trabalhar como um time.
5. Motivar as pessoas.
6. Comunicação face a face.
7. Medir o software em funcionamento.
8. Manter um ritmo sustentável.

9. A qualidade deve ser algo bem visto, bem preparado e importante também.
10. Manter as coisas sempre simples.
11. Os designs têm que ser evolutivos.
12. Refletir regularmente sobre o processo, sobre o que se está fazendo e analisar de que maneira nós podemos melhorar continuamente.

Genrênciamento de Projeto

História do Scrum

- O termo "Scrum" foi cunhado por **Takeuchi** e **Nonaka** em alusão à formação do Rugby, em que todo o time está junto buscando a posse da bola;
- Para Scrum, o conceito de time é extremamente importante;
- A metáfora que utiliza o Rugby ainda se relaciona ao conceito de sprint, que é quando o time rouba a bola e o jogador deve correr rapidamente com ela, e esse processo se repete até a conclusao do jogo.

Exemplo de um Scrum no Rugby

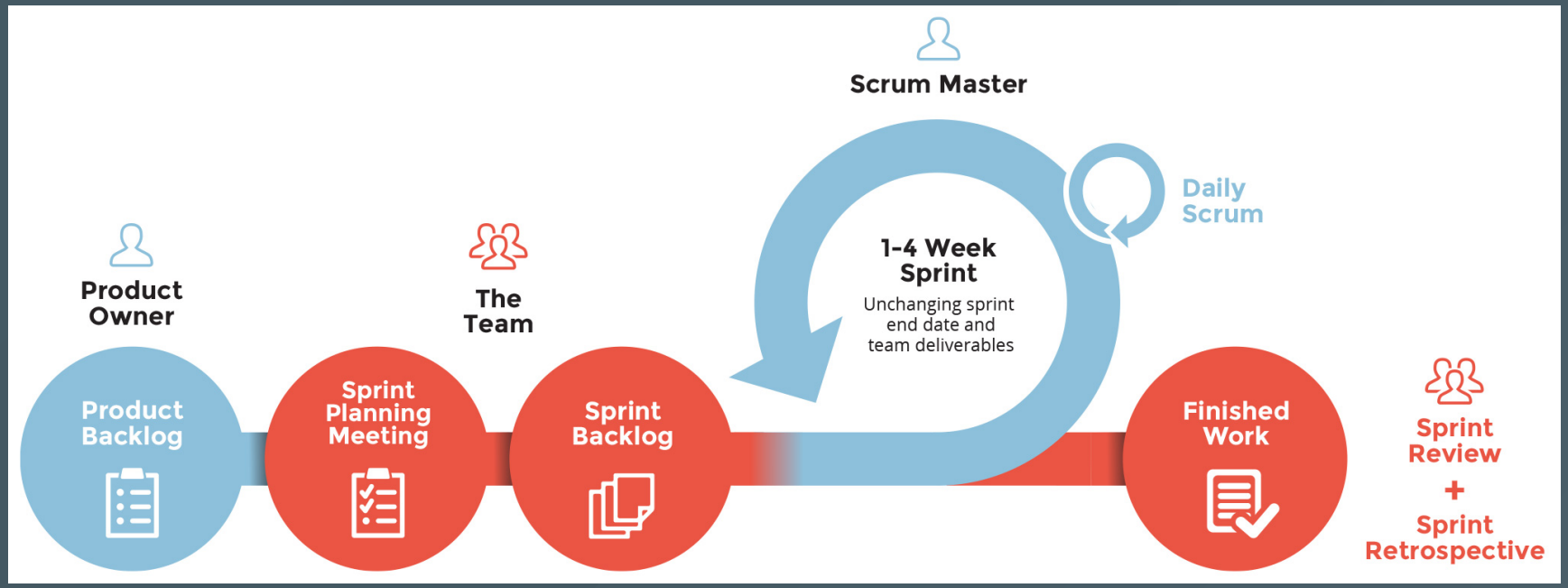
O que é Scrum?

- O Scrum é um framework ágil de gerenciamento de projetos.
- Sua principal característica é trabalhar com time-boxes: caixas de tempo com capacidade definida, rígida e não negociável.
- Uma vez definidas as durações das time-boxes, elas não mudam durante o Sprint atual.

Sprints

- A **Sprint** é o tempo que temos para agregar valor e marcar pontos no projeto.
- Para delimitar o tamanho da Sprint vários fatores devem ser levados em consideração. A duração da Sprint atual não pode ser alterada. No entanto, ao final de um ciclo, ela pode ser revista.
- O Sprint é formado por varias etapas: Planning Meeting, Desenvolvimento, Daily Scrum, Review Meeting e Retrospective.

Sprint Life Cycle



Planning Meeting

- É uma reunião de planejamento que reúne a equipe inteira.
- Entramos nela com uma lista de todos os afazeres (Product Backlog) e saímos com outra, de tarefas específicas a serem concluídas (Sprint Backlog).
- Duraçã de 5% do tamanho do Sprint, pra 1 semana de Sprint, teremos `2 horas` de planejamento.
- O **P.O.** deve ter passado um tempo considerável pegando o topo do Backlog, os itens mais importantes, e refinando-os, se necessário em contato com os membros do time.

Como realizar a Planning Meeting

- O P.O. chega com o Product Backlog, e apresenta os itens de maior prioridade para o cliente.
- Os desenvolvedores vão discutir sobre o item, estimar quanto de esforço deve ser empregado nesse tipo de tarefa.
- Na próxima parte da reunião, é feita a negociação daquilo que realmente cabe na Sprint.
- Objetivo é não sobrecarregar o time.
- Definir uma Meta para o Sprint.

Daily Scrum

- Reunião diária e rápida de no máximo `15 minutos` no próprio ambiente de trabalho, sempre no mesmo horário.
- Três perguntas principais devem ser respondidas: O que fez? O que fará? Quais problemas enfrentou?
- Toda a equipe deve participar.

Review Meeting

- É o momento no qual o cliente e o time de desenvolvimento se reúnem para mostrar os incrementos feitos na Sprint.
- É uma time box que não deve ultrapassar `2.5%` tempo total da Sprint. Para um Sprint de 1 semana, o Review Meeting terá 1 hora de duração.
- Aprensetar os itens prontos para o cliente testar.
- Coletar o feedback e se necessário inserir no Product Backlog.
- Validar se a meta do Sprint foi atendida ou não.

Definição de pronto

- A definição de pronto tem que estar clara e bem definida.
- Sequência de passos a serem seguidos, para que possa considerar um item (task) pronto.
- Exemplo: Desenvolvido -> Testado -> Aprovado pelo cliente.
- Não deve ser um critério muito grande.

Exemplo de quadro de acompanhamento

Retrospective

- Esta reunião fornece a possibilidade de melhoria contínua em que pode-se "lavar roupa suja" para nos reinventarmos para uma próxima Sprint.
- A cada 1 semana de Sprint, recomenda-se 2 horas de retrospectiva.
- É um momento de melhoria contínua, não um momento de apontar culpados.
- Esta reunião gera ações que devem ser tomadas para solucinar os problemas.
- Deixar o resultado desta reunião visivel para o time, e valida-lo na proxima retrospectiva.

> Não importa o que descobriremos nessa reunião, consideraremos que as pessoas agiram dessa forma devido aos conhecimentos que possuíam na época, tempo e recursos disponíveis. Considerando esses aspectos, as pessoas zeram seu melhor, e agora devemos seguir adiante.
>
> -- *Prime Directive*

Como executar uma Retrospetive

- Levantar os pontos positivos e negativos do Sprint.
- Agrupar esses pontos por assunto para facilitar a discução.
- Identificar com o time quais ações podem ser tomadas para solucinar o problema levantado.
- Sair da reunião de restropectiva com pelo menos 1 ação de melhoria definida.
- A mensagem importante é que nós precisamos pensar sobre o nosso trabalho.
- Manter uma lista das ações a serem tomadas.

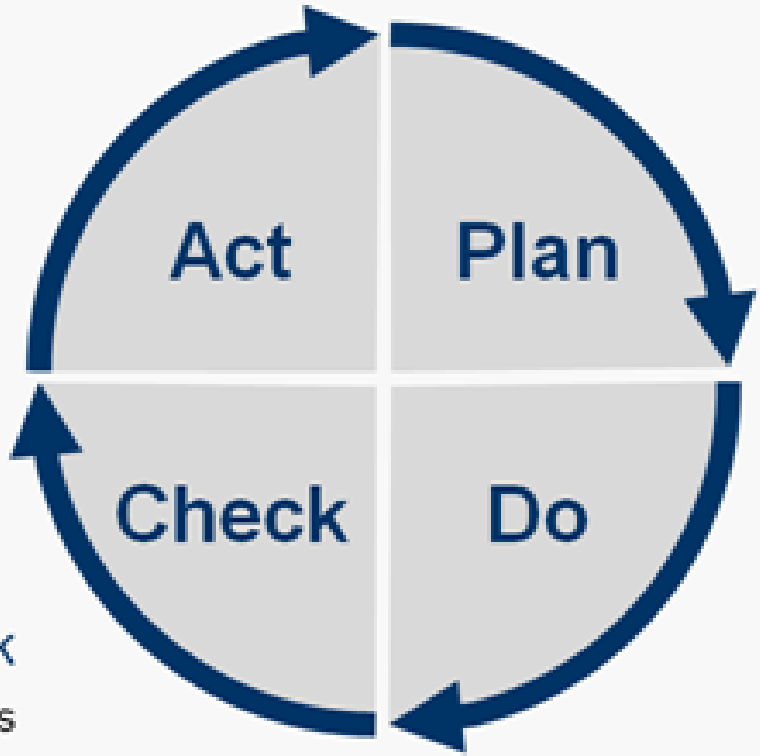
Ciclo de Deming (PDCA)

- Ferramenta utilizada para controle e melhoria continua de processos;
- Primeiramente, é feito um planejamento, que depois é executado, verificado e validado. Após estes passos, o time vai refletir sobre o que aconteceu e irá agir com base no que foi aprendido.

Deming Cycle

ACT

Take action to standardize or improve the process



PLAN

Plan ahead for change

Analyze and predict the results

CHECK

Study the results

DO

Execute the plan, taking small steps in controlled circumstances

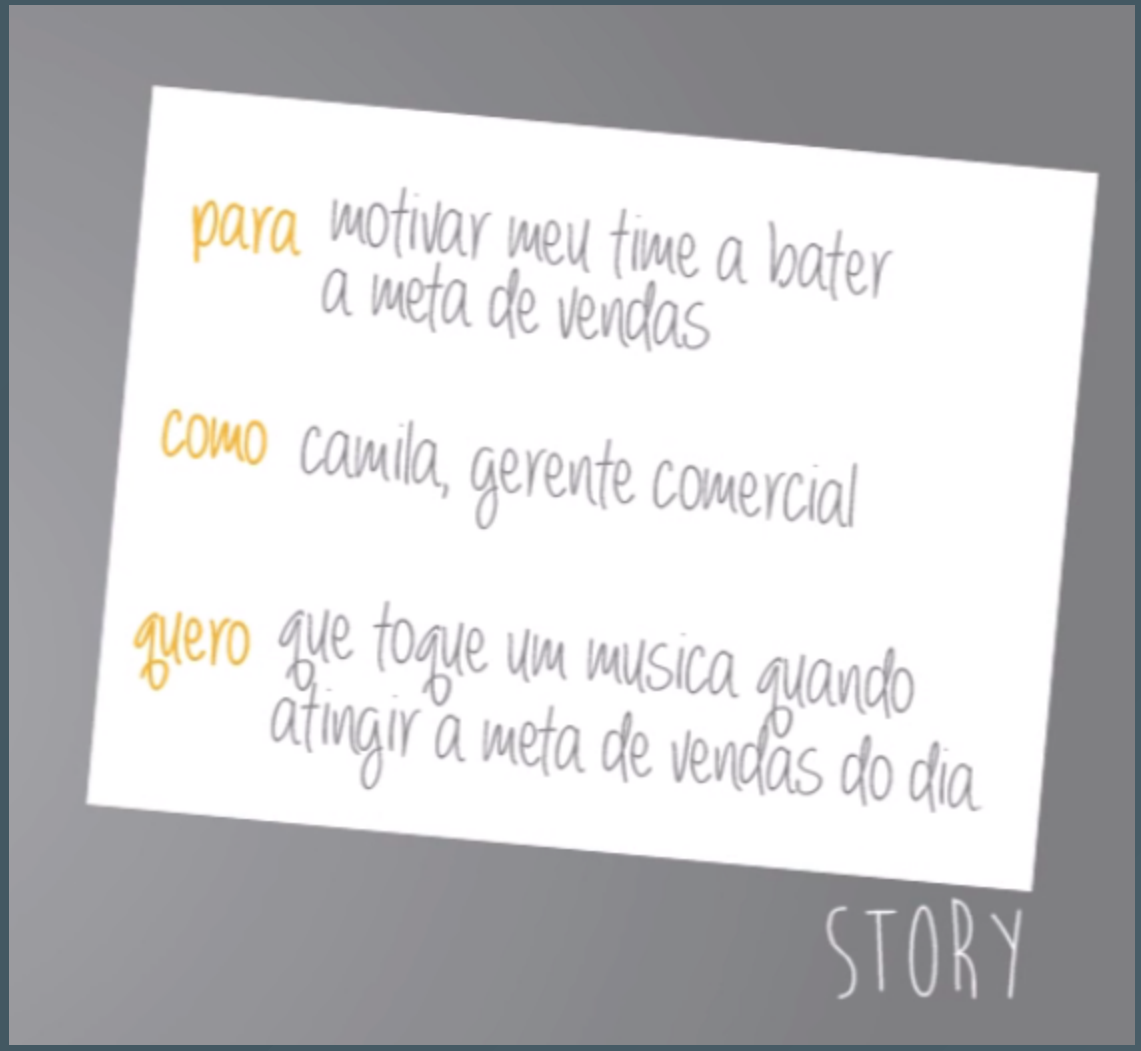
Histórias

- A história é um item que agrega valor ao usuário, escrita de forma bem diferente ao caso de uso, deve ter um título, um porquê , para quem esse item é importante e obetivo da história.
- Uma história faz parte do Product Backlog, mas no momento de desenvolvê-la, vamos paralelizar os sub-itens da história, chamados de tarefa, `task`, para terminá-las o mais rápido possível.
- Quem possui maiores informações sobre a história é o P.O.

Características de boas histórias

- Indepêndencia: máximo de indepêndencia possível entre as histórias.
- Negociáveis: O que não for negociável deve estar explícito.
- Valiosas: específicar bem o valor de cada hisória.
- Estimáveis: detalhadas o suficiente para serem estimaveis (custo benefício).
- Pequenas: que caibam em 1 Sprint, histórias grandes podem ser divididas em menores.
- Testaveis: critérios de aceitação explícitos.

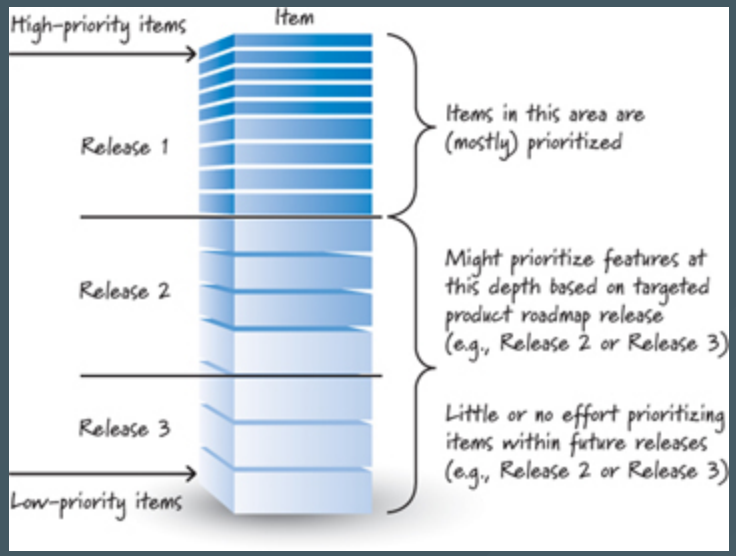
Exemplo de História



Product Backlog e Sprint Backlog

- O ***Product Backlog*** é a lista priorizada das histórias que agregam valor para o cliente. São histórias que envolvem o projeto inteiro. Somente o P.O. mexe nele, mas todo o time pode palpitar. Conforme as histórias vão subindo na prioriedade, elas se tornam mais detalhadas.
- O ***Sprint Backlog*** engloba histórias e tarefas que estão no topo das prioridades e que serão feitas nesta Sprint. No Sprint Backlog o time altera essas tarefas sem que o cliente palpite sobre elas.

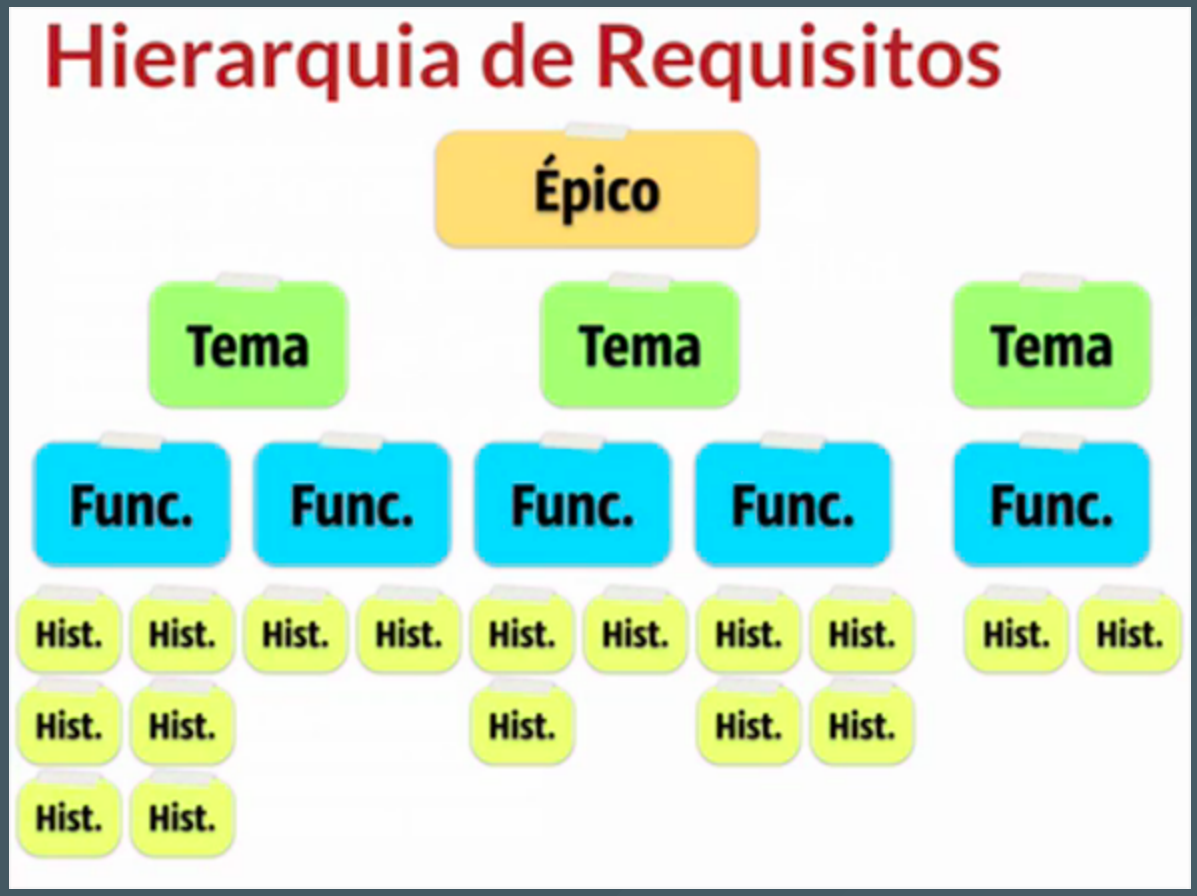
DEEP: Detailed appropriately, estimated, emergent, and prioritized



Backlog Grooming

- Processo feito por meio de uma reunião para definir prioridades do Product Backlog;
- Utilizar hierarquia de requisitos para facilitar a priorização dos itens;
- Não detalhar itens que estão distantes de serem implementados;
- Utilizar técnicas para ajudar a estimar o tempo de cada história, como por exemplo Planning Poker;

Hierarquia de Requisitos



Dívida Técnica

- Problemas técnicos que depois, em um segundo momento, serão corrigidos;
- Quanto mais tempo você demora para resolver esse problema, mais juros você vai acumulando, mais difícil vai ficando pagar essa dívida;
- Reservar um perído de tempo para pagar as dívidas técnicas;
- Manter um backlog das dívidas;

Papéis no Scrum

- **Scrum Master**
- **Product Owner**
- **Desenvolvedores**

Scrum Master

- O papel do Scrum Master é facilitar o ensino sobre o que é o Scrum e as responsabilidades de cada uma das partes envolvidas.
- Não se trata de ser um chefe, mas sim um líder servidor, aquele que está disponível para ajudar e facilitar a comunicação.
- Sua terceira função é resolver impedimentos.

Impedimentos e problemas

- Todo impedimento nasce como um problema, e a resolução disso é função do time.
- Passa-se da categoria problema para impedimento **quando o time tenta resolver uma situação mas não é capaz**.

```
Um bom Scrum Master trabalha para que os impedimentos
de hoje não se tornem problemas amanhã.
```

Product Owner

- Responder dúvidas dos desenvolvedores sobre histórias ou indicar quem poderia respondê-las melhor;
- Deixar claro para o time qual o valor de negócios de cada Sprint e de cada história;
- Manter o Product Backlog atualizado;
- **Uma pessoa, não um comitê.**
- Ele é influenciado tanto por clientes quanto pelo time de desenvolvimento.

Desenvolvedores

- Desenvolvedor é aquele que ajuda a executar o projeto e o faz andar para a frente.
- O time de desenvolvimento que estima trabalho e tempo necessários.
- Os desenvolvedores também decidem o quanto de trabalho pode ser feito em um time-box. Não é uma pessoa externa que define tempo, são os próprios envolvidos no trabalho.

Melhoria contínua

- Melhroia contínua é responsabilidade do time.
- Sempre aplicar as ações retiradas da retrospectivas e resolver os problemas à medida em que eles aparecem;
- A melhoria contínua é essencial;
- Todos devem pensar em melhorar constantemente, em conjunto;