**Origem do Scrum**

\* 1986 takeuchi e nonaka

- equipes pequenas e multidisciplinares

- the new product development game

\* 1993 jeff sutherland

- Documentação e implementação na Easel

\* 1995 ken sutherland

- formalização e implementação no desenvolvimento SW

\* 2001 Manifesto Ágil

**Manifesto para desenvolvimento ágil de software**

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

\* os indivíduos e suas interações acima de procedimentos e ferramentas;

Dar prioridade nas conversas com reuniões pessoais para tomada de decisões para correção de algum problema específico.

\* O funcionamento do software acima de documentação abrangente

Deve existir documentação, porém a funcionalidade do software e correção do bug deve ser prioridade em relação a documentação.

A documentação deve ser suficiente para o cliente e desenvolver entender, não podendo ser exagerada.

\* A colaboração dos clientes acima da negociação de contratos.

O cliente deve estar do seu lado pois nos projetos tanto o cliente ou o desenvolvedor pode precisar um do outro.

Ex.: 4 entregas, entrega 1 correta e no prazo.

Porém na entrega 2 existe problema com time de desenvolvimento e tenta negociar com o cliente a redução de itens na entrega.

A colaboração entre as partes e o jogo de cintura ajuda em eventuais necessidades de ambas as partes.

\* A capacidade de resposta a mudanças acima de um plano preestabelecido.

contornar as situações e dificuldades do dia a dia para entrega e desenvolvimentos para tomar decisões com base no atual momento e dificuldades.

\* Ou seja, mesmo havendo valor nos itens a direita, valorizamos mais os itens a esquerda.

- Os indivíduos e suas interações acima de procedimentos e ferramentas;

- O funcionamento do software acima de documentação abrangente;

- A colaboração dos clientes acima da negociação de contratos;

- A capacidade de respostas a mudanças acima de um plano preestabelecido.

Entre 17 integrantes que escreveram o manifesto estão ken schwaber e jeff sutherland (2001).

**O que é ser ágil**

Agilidade e velocidade com controle.

Nos projetos a entrega deve ser ágil, porém o desenvolvendo ágil deve prever os testes das documentações e sem bugs.

**Quando utilizar o Scrum**

Para que um projeto seja simples a equipe deve saber sobre os requerimentos e ter conhecimento da tecnologia que será aplicada no projeto.

**\* Projeto Simples**

Quanto mais existir conhecimento destes dois pontos mas o projeto será "simples" (sem dúvidas nas regras e na construção).

**\* Projeto Complicado**

Quase todas as regras e suas construções são conhecidas.

**\* Projeto Complexo**

Conhecimento médio das regras de negócios e de sua construção.

**\* Projeto Caos**

Conhecimento baixo das regras de negócios e de sua construção.

Scrum é um framework para desenvolvimento ágil de projetos, uma metodologia ágil de gestão de projetos que se baseia em ciclos curtos de trabalho. Chamados de sprints, com duração de duas a quatro semanas.

Durante cada sprint, a equipe se concentra em entregar um conjunto de funcionalidades ou melhorias ao produto.

O Scrum é uma metodologia eficaz para gerenciar projetos complexos e dinâmicos, pois permite que as equipes se adaptem às mudanças e entreguem valor aos clientes com rapidez.

O Scrum pode ser usado em qualquer setor, mas é mais comumente usado no desenvolvimento de software.

O desenvolvimento cascata é uma metodologia linear, o que significa que as fases são executadas em sequência. Uma vez que uma fase é concluída, não é possível voltar atrás e fazer alterações.

O desenvolvimento cascata é uma metodologia eficaz para projetos com requisitos estáveis e que não estão sujeitos a mudanças ao longo do tempo.

No entanto, o desenvolvimento cascata também tem algumas desvantagens, como:

**- Dificuldade de adaptação:** como as fases são lineares, é difícil adaptar o projeto às mudanças que ocorrem ao longo do tempo.

**- Alto risco:** como as fases são lineares, um erro em uma fase pode afetar todas as fases subsequentes.

O Scrum é uma metodologia que pode ser utilizada em qualquer projeto, mas é mais adequado para projetos que apresentam as seguintes características:

**- Complexidade:** o Scrum é uma metodologia eficaz para gerenciar projetos complexos, pois permite que as equipes se adaptem às mudanças e entregam valor aos clientes com rapidez.

**- Dinâmica:** o Scrum é uma metodologia flexível, que pode ser adaptada às mudanças que ocorrem durante o projeto.

**- Requisitos mutáveis:** o Scrum é uma metodologia que permite que os requisitos sejam alterados ao longo do projeto, sem comprometer o cronograma ou o orçamento.

**- Trabalhar em equipe:** o Scrum é uma metodologia que incentiva a colaboração entre os membros da equipe.

**- Necessidade de entregas frequentes:** o Scrum permite que as equipes entreguem valor aos clientes de forma rápida e frequente.

**- Necessidade de feedback constante:** o Scrum incentiva o feedback constante do cliente, o que ajuda a garantir a qualidade do produto.

A colaboração do time e a vontade de entregar as sprints é muito importante.

O scrum é:

- Leve

- Simples de entender

- Extremamente difícil de dominar, sempre será utilizado por outras equipes que observam o ambiente em que se aplica o scrum.

O Scrum Guide possuí:

- Papéis (3)

- Eventos (5)

- Artefatos (N)

- Regras

O scrum é baseado em processo incremental e iterativo. Sendo um processo empírico de gerenciamento e controle.

O processo empírico se baseia no que deu certo e na correção de rotas com base no conhecimento adquirido nas entregas anteriores.

O scrum é muito mais atitude do que processos em si, sendo escalável em projetos grandes e largos.

**Pilares do Scrum**

Os pilares do Scrum são três princípios fundamentais que sustentam a metodologia. Esses pilares são essenciais para o sucesso do Scrum. Eles ajudam a garantir que as equipes Scrum sejam eficazes, produtivas e inovadoras.

**Transparência**

Garantir que todos da equipe estejam falando a mesma língua (cliente x PO) (PO x desenvolvedores). E a definição de **Done (pronto)**.

As equipes Scrum são transparentes sobre seu trabalho e compartilham informações de forma aberta e honesta. Isso ajuda a garantir que todos estejam na mesma página e que as decisões sejam tomadas com base em informações precisas.

**Inspeção**

A cada sprint inspecionar o que está acontecendo para verificar se deve ter ajuste da rota.

As equipes Scrum inspecionam seu trabalho regularmente para identificar problemas e oportunidades de melhoria. Isso ajuda a garantir que o trabalho seja de alta qualidade e que as equipes estejam no caminho certo para alcançar seus objetivos.

**Adaptação**

Ao perceber o que está incorreto no projeto, realizar ajustes necessários o mais rápido possível.

As equipes Scrum se adaptam às mudanças à medida que elas acontecem. Isso ajuda a garantir que as equipes sejam capazes de responder às mudanças nas necessidades dos clientes ou no ambiente.

Aqui estão alguns exemplos de como os pilares do Scrum podem ser aplicados no mundo real:

* **Transparência:** uma equipe Scrum pode usar um quadro Kanban para visualizar seu trabalho. Isso ajuda a garantir que todos na equipe saibam o que está acontecendo e que as tarefas sejam priorizadas de forma eficaz.
* **Inspeção:** uma equipe Scrum pode realizar uma reunião diária para discutir o progresso do sprint. Isso ajuda a identificar problemas potenciais e a tomar medidas corretivas antes que eles se tornem grandes problemas.
* **Adaptação:** uma equipe Scrum pode usar uma revisão de sprint para receber feedback do cliente. Isso ajuda a garantir que o produto atenda às necessidades dos clientes e que possa ser adaptado às mudanças nas necessidades dos clientes.

**Definição de Done (Pronto)**

**Definition of Done (DOD)**

Todas as tarefas que precisam estar concluídas para que o PB itens seja considerado como **Pronto/Done**.

A definição de **Done** é um acordo entre o Product Owner, a equipe Scrum e quaisquer outras partes interessadas. Ela deve ser clara e concisa, de modo que todos estejam na mesma página sobre o que significa um item estar completo.

A definição de done é importante porque ajuda a garantir que os itens do Backlog do Produto sejam entregues com qualidade e que atendam às expectativas do cliente. Ela também ajuda a equipe Scrum a se organizar e a priorizar seu trabalho.

Exemplo para um item estar **Done**:

* Item deve estar documentado (PO) e aprovado pelo cliente;
* Nenhum bug impeditivo pode estar associado ao item;
* Teste deve ser realizado no ambiente de homologação;
* Código fonte deve estar no repositório e versionado;
* Manual do produto deve está atualizado com este item;
* Documento de gestão de mudança deve contemplar este item;

Mais itens possíveis para análise:

* Teste de performance
* Teste unitário
* Teste de regressão
* Teste de aceitação do usuário
* Revisão do código fonte

Em última análise, a definição de done é específica para cada projeto e organização. No entanto, as características principais mencionadas acima são geralmente aplicáveis em todos os contextos.