**Processo de desenvolvimento tradicional**

* Concepção
* Análise & design
* Desenvolvimento
* Testes
* Implantação

O Princípio de Pareto também se aplica ao desenvolvimento de software, onde 20% das funcionalidades costumam gerar 80% ou mais de benefícios esperados

**Gestão de Projetos Tradicional x Ágil**

* **Tradicional (Waterfall - Cascata):** só permite que o projeto avance quando uma fase está inteiramente completa.
* **Ágil:** Software construído por partes (incremental) e cada parte executa-se em um ciclo (interativo)

|  |  |
| --- | --- |
| **Tradicional** | **Ágil** |
| Escopo definido na fase Inicial do Projeto (**Preditivo**) | Escopo definido ao longo do Projeto (**Adaptativo**) |
| Projeto é controlado por fases e marcos. | Projeto é controlado por funcionalidades entregues |
| Cliente só vê o software funcionando na fase final do Projeto | Cliente pode ver parte do software funcionando na parte inicial do Projeto |
| Resistência a Mudanças | Mudanças constantes de acordo com feedbacks contínuos |

Em projetos tradicionais (cascata), você corre o risco de descobrir que estava errado depois de meses. Com o SCRUM, você descobre que estava errado em no máximo 30 dias.

Ágil é diferente de Rápido.

O que é ser Ágil?

* Rapidez na mudança e no desembaraço;
* Fazer coisas complexas de forma simples;
* Equipe comprometida com os objetivos;
* Maior valor para o cliente;
* Ter capacidade de responder rapidamente a mudanças.
* SCRUM é um dos **frameworks de Gerenciamento de projetos ágeis;**
  + Outros frameworks
    - lim
    - xp
    - cambam
    - trello
* Projetos usando equipes pequenas e multidisciplinares produzem os melhores resultados

Os três pilares do SCRUM:

* **Transparência**
  + Conversar mais e escrever menos
  + Demonstrar o software constantemente aos usuários e obter feedbacks constantes;
* **Adaptação**
  + Requisitos mudam ao longo do tempo
* **Inspeção**
  + Aprender progressivamente com o uso do software

**Razões para adotar o SCRUM:**

* Desenvolvimento e entregue em partes menores (2 a 4 semanas), com constante feedback dos usuários;
* Melhor gerenciamento de riscos; (Redução de incertezas)
* Comprometimento, motivação e transparência da equipe. (Daily Meeting)
* **Maior valor para o negócio;(Priorização do Backlog)**
* Usuários envolvidos durante todo o ciclo;
* Aplicação das Lições Aprendidas. (melhoria contínua)

**Características do time SCRUM:**

* Equipes capazes de se auto-organizarem;
* As tarefas são do time e todos são responsáveis;
* Forte comprometimento com os resultados.

**Product Owner (PO)**

* Representante da área de Negócios
* PO não é um comitê
* Define as funcionalidades do software (Product Backlog)
* Prioriza as funcionalidades de acordo com o valor do negócio
* Garante que o time de desenvolvimento entenda os itens do Backlog no nível necessário

**Scrum Master (SM)**

* Garantir o uso correto do SCRUM
* Scrum Master não é Gerente de Projetos
* Age como facilitador
* Auxilia o PO no planejamento e estimativas do backlog
* Auxilia a equipe a remover impedimentos
* Treina o time em autogerenciamento e interdisciplinaridade

**Time de Desenvolvimento (DEV)**

* Possui habilidades suficientes para desenvolver, testar, criar e desenhar, ou seja, tudo que for necessário para entregar o software funcionando.

**Time Box**

Tempo máximo para fazer uma cerimônia ou todo o desenvolvimento em uma Sprint

Sprint = corrida, arrancada (é o principal evento do SCRUM).

Duração de 1 Sprint: (30 dias corridos ou menos)

**Composição de uma Sprint:**

* Planejamento da Sprint
  + Quem participa: PO, SM, DEV
  + Duração: 08 horas
    - 04 primeiras horas para definir o que fazer,
      * PO explica para o time o que ele deseja naquele Sprint
        + Funcionalidades
        + o porquê
        + o que o time tem que entregar naquele Sprint
        + equipe de DEV tira todas as dúvidas sobre a Sprint
    - 04 horas restante para definir como fazer
      * a equipe de DEV quebra as atividades
      * entende tecnicamente o que tem que fazer
      * o próprio time estima as atividades como fazer
        + como fazer para criar aquilo que o PO deseja
  + Ferramentas de planejamento:
    - Planning poker
      * Quem faz o planejamento são os desenvolvedores
      * Evita o efeito de ancoragem (quando uma pessoa mais experiente dá sua opinião e todos ficam “ancorados” naquela idéia)
      * Selecionam as estórias a serem discutidas na SPRINT
      * Começa-se a fazer a estimativa através de pontuação
      * Verifica-se o resultado e começa-se uma discursão sobre a escolha, chegando ao ponto comum, é feita uma nova votação até que todos cheguem ao consenso e finaliza-se a tarefa, partindo assim para a próxima.
      * Em cima da SPRINT, é retornado para o PO a estimativa de tempo para analisarem e verificarem se é possível fazer no tempo estimado, caso não seja, é revisto as tarefas a serem feitas nesta SPRINT.
* Reuniões diárias (daily meeting)
  + Tabela: O que foi feito no dia anterior /O que está fazendo/Qual foi o impedimento
  + Standup Meeting
    - Reuniões feitas em pé todos os dias
    - No mesmo horário e local definido pelo time – para ser rápido.
    - Tempo máximo de 15 minutos
    - Respondem a 03 perguntas:
      * To do
      * Doing
      * Done

**Quadro Kanban**

|  |  |  |
| --- | --- | --- |
| **To do (Tarefas para fazer)** | **Doing (Tarefas em andamento)** | **Done (Tarefas finalizadas)** |
|  |  |  |

* Revisão da Sprint (Review)
  + Time DEV apresenta para o PO o trabalho feito
  + Time box de 04 horas
  + Apresenta tudo que desenvolveu, passo-a-passo, para colocar em produção para que não seja uma surpresa para o PO.
  + Como é apresentado no último dia, o PO aceita ou não
    - Caso aceite será colocado para produção.
* Retrospectiva da Sprint
  + Reunião da equipe para Lições Aprendidas – ***Transparência***
  + No máximo 03 horas, podendo ser menor de acordo com o tempo.
  + Para a resolução de problemas dentro da equipe, todos devem colocar:
    - Todos os problemas
    - Dificuldades
    - Problemas de definições



**Diferença Prática entre:**

* **Gestão de Projetos Tradicional**
  + Planeja todo o projeto para depois executar
  + Tempo muito grande, podendo tornar o produto inviável
  + Só é entregue o produto assim que finalizar todas as etapas anteriores de todo o projeto
* **Gestão de Projetos Ágil**
  + Planeja partes do projeto
  + Tempo curto com rápido aceite do cliente
  + Partes do produto são entregues assim que estiver ok

**Fundamentos do Projeto Ágil**

**Parte 01: Papéis e Responsabilidades do Product Owner (PO)**

* **Product Owner (PO):** 
  + Profissional que tem a visão
    - Do que será desenvolvido,
    - As necessidades a serem atendidas,
    - O público que vai utilizar os serviços e
    - Os objetivos a serem alcançados

**Refining e Planning**

**Refining:** PO apresenta as estórias a serem trabalhadas

**Planning:** É discutido o que será feito e nenhum dos componentes do grupo pode sair com dúvidas.

Nesta fase o PO apresenta para o time quais são as principais estórias do Product Backlog

Todas as dúvidas que passarem o DEV tem que tirar com o PO.

Na segunda parte, somente a equipe de DEV participa, sem o PO.

É composto por **Épicos e Estórias**

* **Épicos:** Incremento sem muito detalhamento, ajuda a te direcionar dos caminhos que deve seguir
* **Estória:** Detalhamento dos **Épicos**, um **Épico** normalmente se divide em várias **estórias**, onde ficam descritos o que deve acontecer em suas regras de negócio.
  + **Escrevendo uma Estória:**
    - **Nome da Estória**
    - **Descrição da Estória(**Eu, como, quero, quando**)**
    - **Regras de Negócio(**Separar Regras de Front-End de Regras de Back-end**)**
    - **Tela(**Link ou imagem das telas a serem desenvolvidas**)**
    - **KPI(**Quais os objetivos/valor a estória precisa atingir**)**
    - **Tagueamento(**Como a estória será “Tagueada” para poder mensurar os **KPI)**
    - **Critérios de Aceite(**Qual o passo-a-passo de todos os Caminhos Felizes possíveis para a estória deve cumprir para que ela seja considerada aceita)

**Riscos Positivos:**

* **Positivos:** muito ignorado nos projetos, porém um dos fatores de maiores ganhos no desenvolvimento de sistemas.
* **Negativos:** itens que podem afetar o prazo, custo ou escopo de um projeto de maneira que pode acabar inviabilizando-o

**Release Planning**

Liberação ou lançamento de **software** (em inglês: **release**) é o lançamento de nova versão oficial de produto de **software.** Cada vez que um produto de **software** é criado ou modificado, o fabricante e seus desenvolvedores decidem sobre como distribuir ou o novo produto ou a modificação às pessoas que o utilizam.

**Tipos de Release Planning:**

* De Múltiplas Squads
* De Projeto

**Definindo Escopo e Prioridade**

Comece definindo o objetivo ou valor que deseja atingir.

**Transformação Digital**

A **Transformação digital** é um processo no qual as empresas fazem uso da tecnologia para melhorar o desempenho, aumentar o alcance e garantir resultados melhores

É uma mudança estrutural nas organizações dando um papel essencial para a tecnologia.

**Critério de Aceite**  é uma lista de critérios que precisam ser alcançados para que o User Story atenda os requisitos do usuário e seja aceita pelo Product Owner.

Os critérios de aceitação têm o objetivo de:

* Definir limites para as users Stories.
* Ajudar o PO a detalhar em alto nível o que é necessário para entregar valor ao cliente.

**Cliente / Stakeholders**

O **stakeholder** é uma pessoa ou um grupo que legitima as ações de uma organização e que tem um papel direto ou indireto na gestão e resultados dessa mesma organização. Desta forma, um **stakeholder** pode ser afetado positivamente ou negativamente, dependendo das suas políticas e forma de atuação.

Alguns exemplos de **stakeholders** de uma empresa podem ser os seus funcionários, gestores, gerentes, proprietários, fornecedores, concorrentes, ONGs, clientes, o Estado, credores, sindicatos e diversas outras pessoas ou empresas que estejam relacionadas com uma determinada ação ou projeto.

**Daily e Retrospectiva**