PROJETO DE SOFTWARE

AULA 5

Metodologias de Desenvolvimento

**Metodologias**

São processos, práticas, regras que orientam o desenvolvimento do software.

Envolve **organização de tarefas; papéis e responsabilidades** e o **fluxo do trabalho**.

**Por que usar?**

Maior controle sobre o projeto; Redução de riscos; Melhor comunicação e Padronização de processos

**Metodologias Clássicas**

São metodologias mais antigas, surgiram entre a década de 70-90. São baseadas em um planejamento detalhado e execução sequencial.

Elas tinham uma **Documentação muito extensa** e **Pouca flexibilidade**

* **Cascata (Waterfall)**
  + Dividida em fases sequenciais. Cada fase dependia da conclusão da anterior.
  + Simples de entender
  + Não há retorno para etapas passadas sem refazer TUDO
  + Requisitos 🡪 Design 🡪 Implementação 🡪 Testes 🡪 Manutenção
* **Modelo Incremental**
  + Funciona por módulos (ex: Módulo de Login, Módulo de Compra e Módulo de Pagamento)
  + Cada desenvolvimento de módulo envolve planejamento, execução e fim
  + A entrega trabalha com funcionalidades, que já podem ser usadas
  + Entregas parciais 🡪 Facilita testes
* **Modelo Espiral**
  + Processo de looping, para resolver o problema do **Cascata**(Retrabalho).
  + Mais flexibilidade
  + Cada volta no “espiral” representa uma **iteração** no projeto
  + Etapas: **planejamento, análise de riscos, desenvolvimento e teste, avaliação com o cliente**
* **Modelo V**
  + Uma evolução do modelo **Cascata** para detectar problemas mais cedo
  + Mantém a estrutura do **Cascata**, porém, introduz uma correspondência direta entre cada etapa de desenvolvimento e uma etapa de teste
  + Verificação e validação acontecem em paralelo (Verifica cada etapa do outro lado) 🡪 Exemplo: Do lado esquerdo, há “Design do Sistema” e do lado direito, que verifica, há “Teste do Sistema”, o lado esquerdo sempre é testado
  + O “formato” desse modelo é um V

**Não existe uma metodologia melhor do que outra, tudo depende do contexto em que ela é inserida.**

**Metodologias Ágeis**

Surgiram como uma resposta às limitações das metodologias clássicas. Foram criadas a partir do **Manifesto Ágil** (2001).

São características: Curto ciclo de desenvolvimento (iterações); Feedback contínuo; Adaptabilidade e Entrega incremental de valor.

**Manifesto Ágil**

Criado em 2001 por 17 desenvolvedores. É um conjunto de valores e princípios com o objetivo de tornar o desenvolvimento mais flexível.

Os principais valores envolvem: **Indivíduos e Interações; Software funcionando; Colaboração com o cliente; Resposta à mudanças** (plano rígido)

**Metodologias Ágeis**

* **Scrum**
  + É um framework (estrutura) que organiza o trabalho em ciclos chamados de **Sprints** (duram geralmente de 1 a 4 semanas)
  + Pode ser dividido em **papéis principais**, **artefatos** e **eventos**
  + Papéis principais: Define os papéis do time
    - **Product Owner**: Define e prioriza os requisitos;
    - **Scrum Master**: Facilita o processo (observa os sprints), remove impedimentos;
    - **Development**: É o time que desenvolve o produto.
  + Eventos: Momentos regulares de planejamento, inspeção e adaptação
    - **Sprint Planning**: Planejamento da Sprint;
    - **Sprint**: Ciclo de trabalho;
    - **Daily Scrum**: Reunião diária de sincronização;
    - **Sprint Review**: Demonstração do trabalho realizado;
    - **Sprint Retrospective**: Reflexão de melhorias no processo.
  + Artefatos: Itens que registram o que foi feito e organiza o que será feito
    - **Product Backlog**: Lista de tudo que será feito;
    - **Sprint Backlog**: Lista de tarefas para uma sprint;
    - **Incremento**: Resultado final da sprint.
* **Kanban**
  + Método **visual** que gerencia o fluxo de trabalho, é baseado em um quadro com colunas que representam diferentes etapas do processo
  + Os principais fundamentos são:
    - **Visualizar o trabalho**: Usa um quadro para mostrar as tarefas
    - **Limitar o trabalho em progresso**: Define limites para cada coluna
    - **Gerenciar o fluxo**
  + A estrutura das colunas é muito flexível, mas possui uma padrão que é: **A fazer 🡪 Em Andamento 🡪 Concluído**
  + Normalmente utilizado em conjunto com o Scrum
* XP (Extreme Programming)
  + Ficada em qualidade de código e resposta rápida a mudanças
  + Principais fundamentos:
    - Comunicação constante entre equipes e clientes;
    - Feedback rápido sobre o software desenvolvido.
  + Práticas da metodologia:
    - **Programação em Par (Pair Programming)**
    - **Testes automatizados (TDD)**: Testes são escritos **antes** do código
    - **Integração contínua**: Teste do código várias vezes ao dia
    - **Refatoração**: Melhora continuamente o código sem alterar a funcionalidade
    - **Propriedade coletiva**: Qualquer dev pode alterar qualquer parte (pode ser um problema)