

# PROCESSOS DE SOFTWARE

## Aula 02 – Agile e UP

Prof. Fabricio Freire



# METAS DE APRENDIZAGEM

- Compreender as diferenças entre projetos de software;
- Desenvolver conceitos básicos sobre PMI e PMBOK;
- Introduzir o Gerenciamento Ágil de Projetos;
- Conhecer modelos adotados no Gerenciamento de Software.

# O que vimos na aula passada

- Compreender as diferenças entre projetos de software.
- Desenvolver conceitos básicos sobre PMI e PMBOK.

# Ciclo de Vida do Projeto

**Conjunto de fases que compõem um projeto**

**Cada fase:**

- Possui atividades específicas;
- Gera produtos/entregas;
- Termina com revisão;
- Determina se prossegue para próxima fase.

# Características do Ciclo de Vida

- ✓ Define início e fim do projeto;
- ✓ Determina ações de transição entre fases;
- ✓ Define trabalho técnico de cada fase;
- ✓ Determina equipe envolvida;
- ✓ Estabelece mecanismos de controle e aprovação.

# Agile Project Management (APM)

## Gerenciamento Ágil de Projetos

Surgiu da necessidade de:

- **Agilidade** no desenvolvimento;
- **Flexibilidade** nas comunicações;
- **Rapidez** na entrega de software;
- Resposta às mudanças.

# Princípios da Agilidade

1. Prioridade: satisfazer o cliente;
2. Aceitar mudanças de requisitos;
3. Entregar software funcionando frequentemente;
4. Comercial e desenvolvedores trabalham juntos;
5. Equipe motivada e com ambiente adequado;
6. Conversa aberta como comunicação efetiva;

# Princípios da Agilidade

7. Software funcionando é medida de progresso;
8. Desenvolvimento sustentável;
9. Excelência técnica aumenta agilidade;
10. Simplicidade é essencial;
11. Equipes auto-organizadas;
12. Avaliação em intervalos regulares.

# Pessoas no Projeto

## Stakeholders identificáveis:

- **Gerentes** - Definem problemas do negócio;
- **Gerentes de Projeto** - Planejam e controlam;
- **Profissionais** - Habilidades técnicas;
- **Clientes** - Especificam requisitos;
- **Usuários** - Interagem com o sistema.

# Modelagem e Diagramas

## Por que modelar?

- Visualizar o sistema;
- Especificar estrutura/comportamento;
- Guiar a construção;
- Documentar decisões.

**Ferramenta principal:** UML (*Unified Modeling Language*)

# Cronogramas e Métricas

## **Cronograma:**

- Distribui esforço ao longo do projeto;
- Aloca tarefas específicas;
- Evolui e refina-se com o tempo.

## **Métricas:**

- Proporcionam base objetiva;
- Permitem avaliação quantitativa;
- Ajudam no planejamento e controle.

# Conclusão

**Gerenciamento de Software é essencial para:**

- Projetos complexos e de qualidade;
- Controle de prazos e custos;
- Compartilhamento de recursos;
- Software confiável e estável.

A chave é encontrar o equilíbrio entre gestão enxuta e gestão excessiva