

# Conceitos Fundamentais de Engenharia de Software

Prof. Me. Lucas Bruzzone

Aula 01

# Objetivos da Disciplina

- Compreender os fundamentos da engenharia de software
- Conhecer os principais modelos de processo de desenvolvimento
- Aplicar metodologias ágeis no desenvolvimento de software
- Dominar técnicas de engenharia de requisitos
- Utilizar UML para modelagem de sistemas
- Desenvolver habilidades de planejamento e gerenciamento de projetos

## 1º Bimestre

- Conceitos fundamentais
- Modelos de processo
- Desenvolvimento ágil
- Engenharia de requisitos

## 2º Bimestre

- UML - Linguagem de modelagem
- Análise e projeto
- Padrões de projeto
- Projeto prático

## IEEE Definition

"A aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de software."

- **Sistemática:** Seguir métodos e processos estruturados
- **Disciplinada:** Aderência a padrões e boas práticas
- **Quantificável:** Uso de métricas e medições

## Programa

- Código executável
- Solução pontual
- Desenvolvimento individual
- Sem documentação formal
- Uso pessoal/limitado

## Software

- Sistema completo
- Documentação
- Testes
- Manutenção
- Uso em produção
- Equipe de desenvolvimento

# Características Únicas do Software

- **Intangível:** Não possui forma física
- **Não se deteriora:** Não sofre desgaste físico
- **Complexidade:** Crescimento exponencial da complexidade
- **Flexibilidade:** Pode ser modificado facilmente
- **Conformidade:** Deve se adaptar a outros sistemas
- **Invisibilidade:** Difícil de visualizar estrutura

- **Software de Sistema**
  - Sistemas operacionais, drivers, compiladores
- **Software de Aplicação**
  - Processadores de texto, jogos, sistemas empresariais
- **Software Científico/Engenharia**
  - Simulações, CAD, análise numérica
- **Software Embarcado**
  - Controle de dispositivos, sistemas automotivos
- **Software Web**
  - Aplicações web, e-commerce, redes sociais

# Por que Engenharia de Software?

- **Crescimento da dependência:** Software está em tudo
- **Complexidade crescente:** Sistemas cada vez maiores
- **Custos elevados:** Desenvolvimento e manutenção caros
- **Prazos críticos:** Time-to-market importante
- **Qualidade essencial:** Falhas podem ser catastróficas
- **Equipes grandes:** Necessidade de coordenação



## Técnicos

- Complexidade
- Requisitos mutáveis
- Integração de sistemas
- Performance e escalabilidade

## Gerenciais

- Estimativas precisas
- Gerenciamento de equipes
- Comunicação
- Controle de qualidade

- **Especificação:** O que o sistema deve fazer
- **Desenvolvimento:** Produção do software
- **Validação:** Verificar se atende às necessidades
- **Evolução:** Modificar para novos requisitos

**Essas atividades são presentes em todos os processos de software**

## Crise do Software e Características de Qualidade

Estudaremos os problemas históricos do desenvolvimento de software e como a qualidade se tornou um fator crítico