

# Conceitos Fundamentais da Engenharia de Software

Princípios essenciais para o desenvolvimento de sistemas  
eficazes

1

1

## Introdução à Engenharia de Software

José Gonçalves

2

2

## Objetivos da Aula

### Entendimento dos Conceitos Fundamentais

A aula visa garantir uma compreensão abrangente dos conceitos essenciais da engenharia de software.

### Fases do Ciclo de Vida do Software

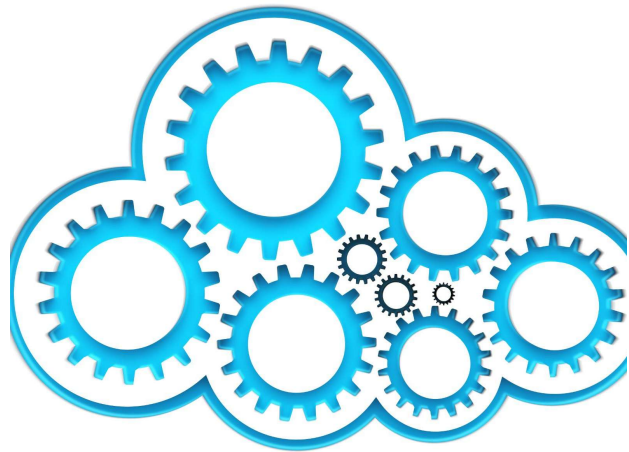
Serão abordadas as fases do desenvolvimento: requisitos, design, implementação, testes e manutenção.

### Abordagem Progressiva e Participativa

A aula é organizada em blocos temáticos para facilitar a aprendizagem e incentivar a participação ativa.

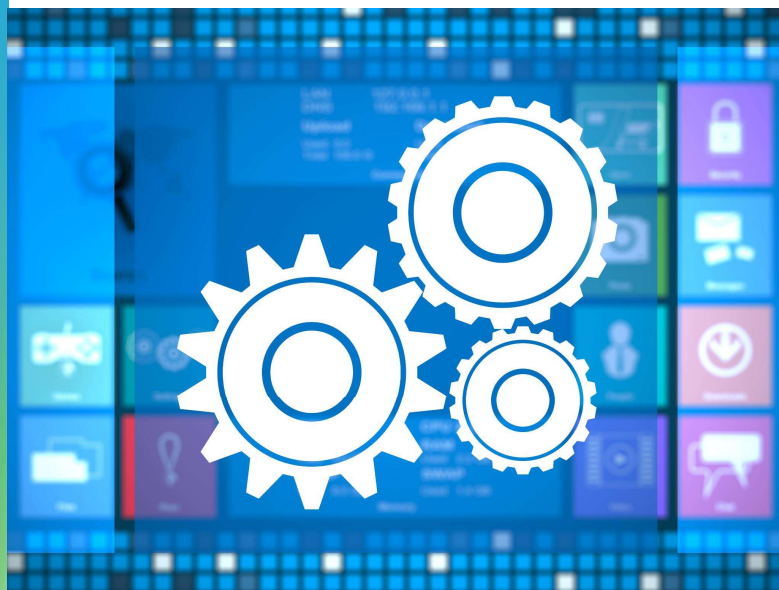
### Importância da Engenharia de Software

Destaca a relevância da engenharia de software para qualidade, eficiência e sustentabilidade dos sistemas.



3

3



## Definição de Engenharia de Software

### Definição e Âmbito

Engenharia de software aplica métodos, técnicas e ferramentas para criar software de qualidade ao longo do seu ciclo de vida.

### Planeamento e Gestão

Inclui planeamento, gestão de projetos e garantia de qualidade para assegurar software funcional e fiável.

### Benefícios e Resultados

Boas práticas reduzem custos, minimizam riscos e aumentam a satisfação dos clientes com software eficiente.

4

4

# Processo de Desenvolvimento de Software

5

5

## Visão Geral do Processo

### Fases do Desenvolvimento

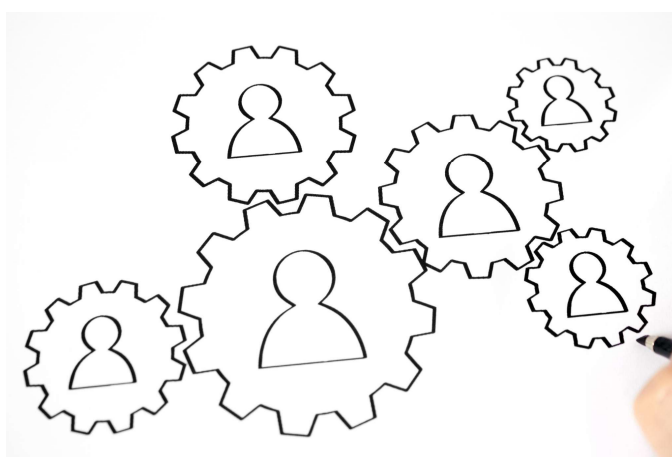
O desenvolvimento de software envolve recolha de requisitos, design, codificação, testes e manutenção para garantir qualidade.

### Modelos de Processo

Modelos como cascata, iterativo e ágil orientam as atividades com base no projeto e necessidades da equipa.

### Gestão Eficaz de Projetos

Entender o processo permite gerir projetos garantindo entregas pontuais e dentro do orçamento.



6

6

## Levantamento de Requisitos

### Importância da Fase

O levantamento de requisitos é fundamental para definir as necessidades dos clientes e garantir o sucesso do projeto.

### Tipos de Requisitos

Requisitos funcionais descrevem ações do sistema, enquanto não funcionais tratam de desempenho, segurança e usabilidade.

### Técnicas de Coleta

Entrevistas, questionários, observação e análise documental são usadas para captar informações relevantes.

### Benefícios da Documentação

Documentar requisitos previne mal-entendidos e serve de base para design, testes e validação do software.



7

7

## Design de Software

### Definição de Design de Software

O design de software transforma requisitos em arquitetura e componentes do sistema para facilitar o desenvolvimento.

### Princípios Fundamentais

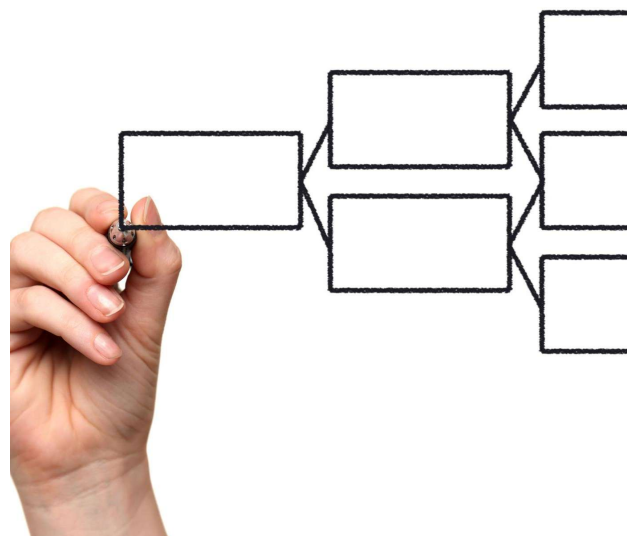
Modularidade, coesão e baixo acoplamento são essenciais para um design de software robusto e flexível.

### Diagramas de Design

Diagramas de classes, casos de uso e sequência ajudam na representação visual do design de software.

### Benefícios do Bom Design

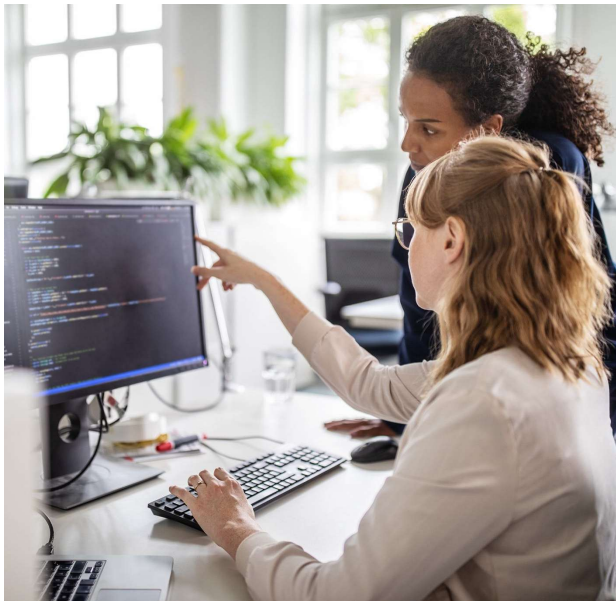
Um bom design facilita implementação, testes, manutenção e promove reutilização e escalabilidade do código.



8

8

## Implementação



### Codificação e Boas Práticas

Programadores escrevem código claro, consistente e bem documentado conforme o design definido.

### Controle de Versões

Sistemas como Git gerem alterações e facilitam a colaboração entre a equipa.

### Testes Unitários

Testes verificam o funcionamento correto de cada componente durante a implementação.

### Qualidade e Manutenção

Padrões de codificação e revisões regulares garantem fiabilidade e evolução do software.

9

9

## Testes de Software

### Objetivo dos Testes

Testes verificam e validam que o software funciona conforme requisitos especificados.

### Tipos de Testes

Incluem testes unitários, de integração, sistema e aceitação para garantir qualidade em diferentes níveis.

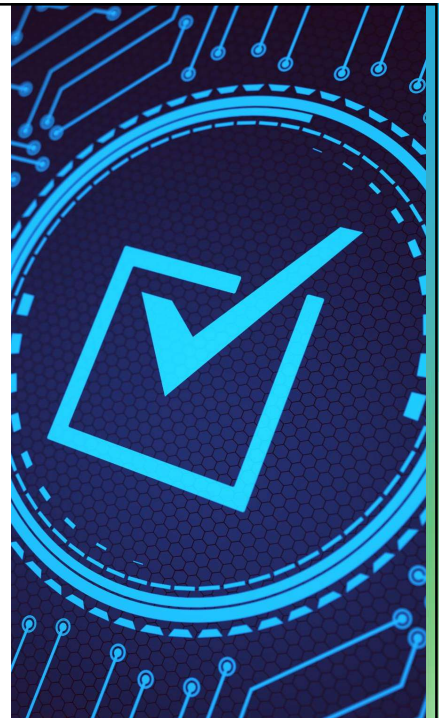
### Importância dos Testes

Testes sistemáticos identificam e corrigem erros, aumentando a confiança na qualidade do produto.

### Testes Automatizados

Automação de testes melhora eficiência e facilita manutenção contínua do software.

10



10

## Manutenção de Software

### Fases da Manutenção

A manutenção inclui correções de erros, melhorias, e adaptações a novas necessidades ou ambientes.

### Tipos de Manutenção

Manutenção pode ser corretiva, evolutiva, adaptativa ou preventiva conforme a intervenção necessária.

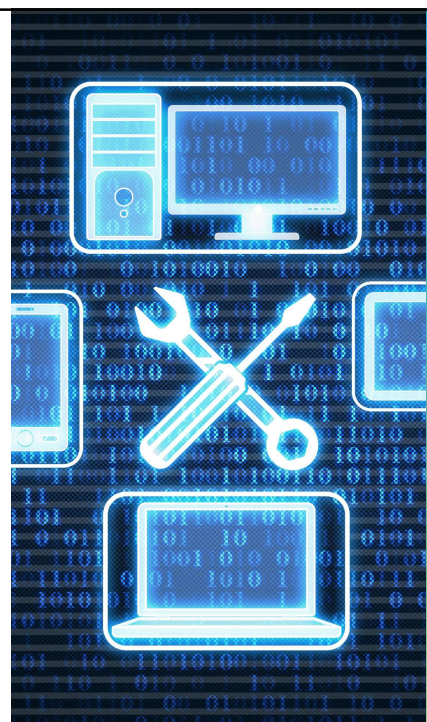
### Documentação e Gestão de Mudanças

Documentação clara e gestão de mudanças são essenciais para garantir segurança e estabilidade no software.

### Importância da Manutenção

Manutenção contínua é vital e pode representar grande parte do custo total do ciclo de vida do software.

11



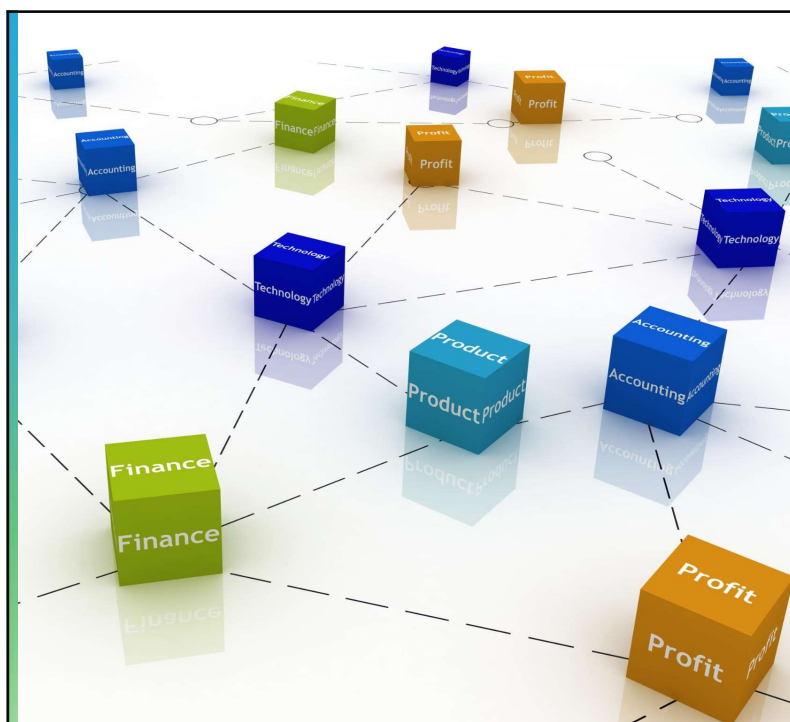
11

## Modelos de Processo de Software

12

12





## Modelo em Cascata

### Processo Sequencial Estruturado

Cada fase do modelo em cascata ocorre de forma sequencial e estruturada, respeitando uma ordem fixa.

### Adequado para Requisitos Estáveis

Ideal para projetos com requisitos bem definidos e estáveis, permitindo planejamento detalhado.

### Limitações em Ambientes Dinâmicos

Pouca flexibilidade dificulta a adaptação a mudanças frequentes ou projetos ágeis.

13

13



## Modelos Iterativos, Incrementais e Ágeis

### Desenvolvimento Iterativo e Incremental

Desenvolver software em ciclos sucessivos permite adicionar funcionalidades graduais e adaptar-se a mudanças rápidas.

### Modelos Ágeis e Colaboração

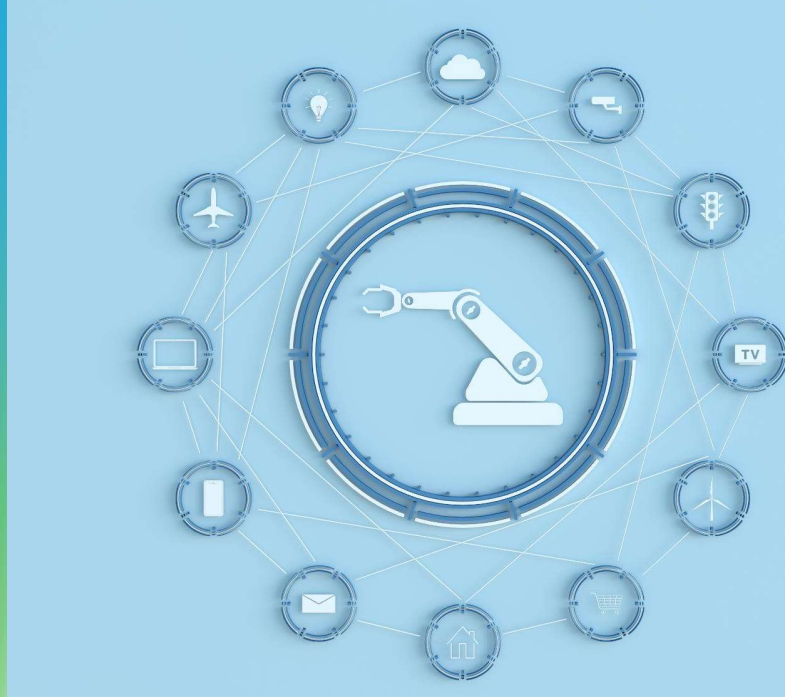
Modelos ágeis promovem colaboração entre equipes e entrega contínua de valor através de sprints curtos e revisões regulares.

### Benefícios da Agilidade

Agilidade reduz riscos, melhora comunicação e aumenta satisfação dos utilizadores, especialmente em ambientes incertos.

14

14



## Resumo dos Conceitos Abordados

### Ciclo de Vida do Software

As fases principais incluem levantamento de requisitos, design, implementação, testes e manutenção, essenciais para o desenvolvimento.

### Modelos de Processo

Foram apresentados modelos como cascata, iterativos, incrementais e ágeis, guiando a gestão do projeto de software.

### Importância da Compreensão

Compreender conceitos ajuda a planejar, executar e gerir projetos, aplicando boas práticas profissionais.

15