



# Processos de Software: Modelos e Atividades

Esta apresentação aborda os processos de software, explorando modelos, atividades e desafios. Discutiremos a importância de adaptar os processos às mudanças, prototipagem, entrega incremental e modelos como o Espiral e RUP. O objetivo é fornecer uma compreensão abrangente para estudantes de engenharia de software.

 por Waldinelly Costa

# Modelos de Processo de Desenvolvimento de Software

Os modelos de processo de software são representações abstratas das atividades envolvidas na produção de um sistema de software. Eles fornecem um guia para a organização e o gerenciamento do projeto, desde a concepção até a entrega. A escolha do modelo adequado é crucial para o sucesso do projeto.

## Modelo em Cascata

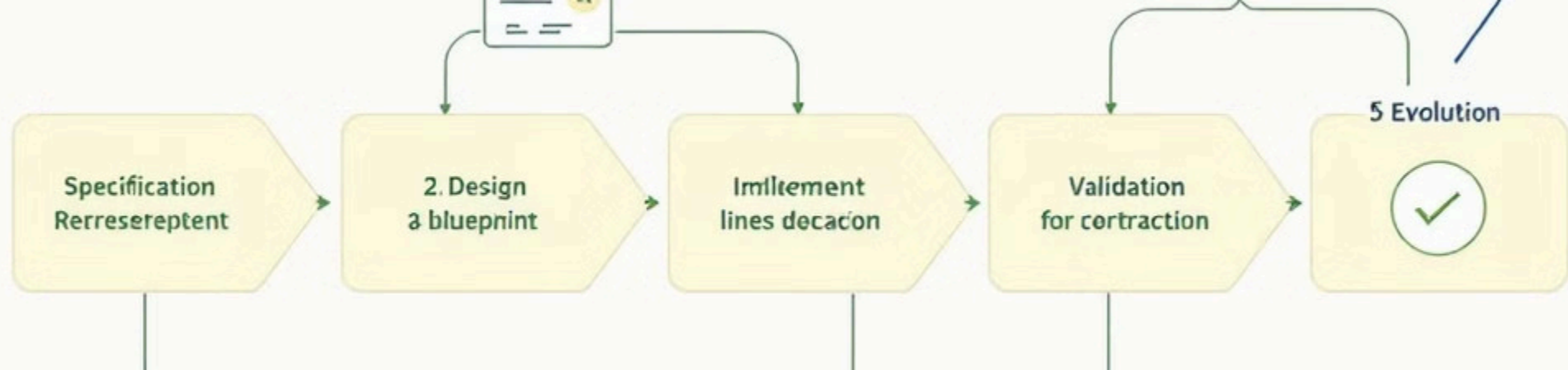
Abordagem sequencial, onde cada fase deve ser concluída antes de iniciar a próxima. Adequado para projetos com requisitos bem definidos.

## Desenvolvimento Incremental

O sistema é construído em incrementos, com cada incremento adicionando funcionalidades. Permite a entrega antecipada de versões funcionais.

## Engenharia Orientada a Reúso

Foca na integração de componentes de software existentes. Reduz o tempo e o custo de desenvolvimento.



# Atividades Fundamentais do Processo de Software

As atividades do processo de software são as tarefas essenciais que devem ser realizadas para transformar uma necessidade em um sistema funcional. Cada atividade tem um papel específico e contribui para o sucesso do projeto.

## 1 Especificação de Software

Definição detalhada dos requisitos do sistema, incluindo funcionalidades, desempenho e restrições.

## 2 Projeto e Implementação

Transformação da especificação em um sistema executável, através do projeto da arquitetura e codificação do software.

## 3 Validação de Software

Verificação de que o sistema atende aos requisitos especificados e às necessidades dos usuários.

## 4 Evolução do Software

Adaptação do sistema a novos requisitos e mudanças no ambiente, garantindo sua relevância contínua.

**Requirements  
Gathering**

**Design &  
Prototyping**



# Um Modelo Geral: Processo de Projeto de Software

O processo de projeto de software envolve a criação de uma representação do sistema que pode ser implementada. As principais atividades incluem projeto da arquitetura, projeto da interface, projeto de componentes e projeto de banco de dados.

1

**Análise de Requisitos**

Compreender as necessidades do cliente.

2

**Design da Arquitetura**

Definir a estrutura do sistema.

3

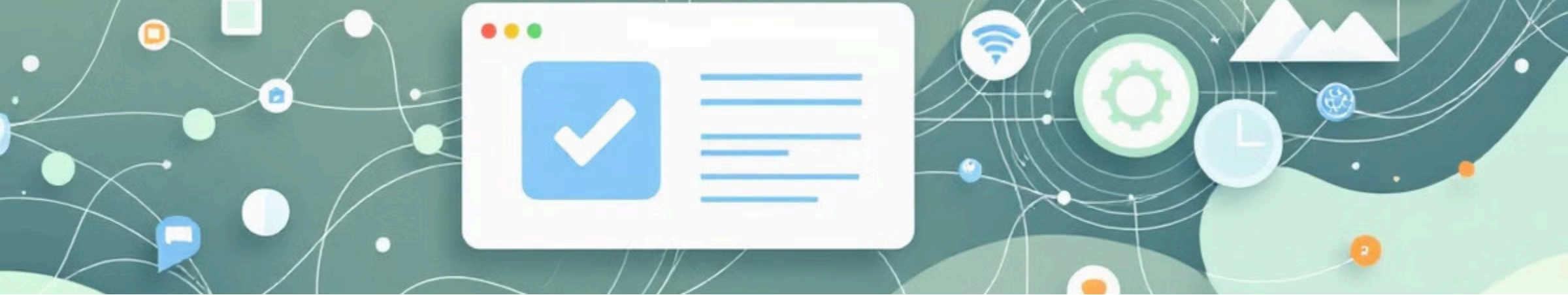
**Design Detalhado**

Especificar componentes e interfaces.

4

**Implementação**

Escrever o código do software.



# Lidando com Mudanças nos Processos de Software

A capacidade de lidar com mudanças é fundamental para o sucesso de um projeto de software. As mudanças podem surgir de novas necessidades do cliente, avanços tecnológicos ou descobertas durante o desenvolvimento. A prototipagem e a entrega incremental são estratégias eficazes para mitigar os riscos associados às mudanças.

## Identificação da Mudança

Reconhecer a necessidade de alteração.

## Implementação da Mudança

Modificar o sistema para acomodar a mudança.

1

2

3

## Análise do Impacto

Avaliar os efeitos da mudança no sistema.

# Modelos de Processo Adaptativos: Espiral e Prototipagem

O modelo espiral e a prototipagem são abordagens adaptativas que permitem lidar com a incerteza e as mudanças nos requisitos. Ambos os modelos envolvem iterações e feedback contínuo, garantindo que o sistema evolua de acordo com as necessidades do cliente.

## Modelo Espiral de Boehm

Abordagem iterativa que incorpora análise de riscos, prototipagem e planejamento em cada ciclo. Adequado para projetos complexos.

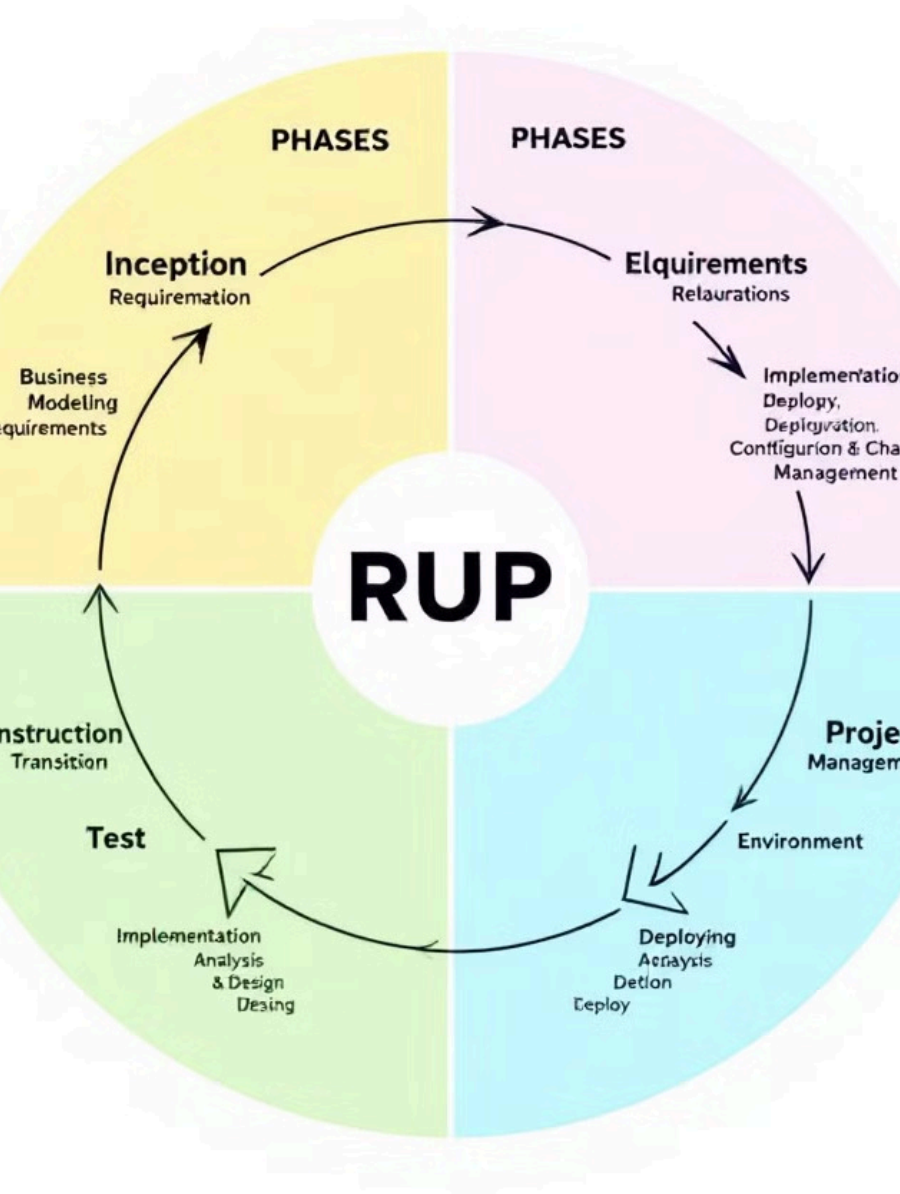
## Prototipagem

Criação de versões preliminares do sistema para obter feedback dos usuários e refinar os requisitos. Reduz riscos e melhora a usabilidade.



# Rational Unified Process (RUP)

O Rational Unified Process (RUP) é um modelo de processo iterativo e incremental que organiza o desenvolvimento de software em quatro fases: Concepção, Elaboração, Construção e Transição. Cada fase envolve diversas disciplinas, como requisitos, análise, projeto, implementação, teste e implantação.



## Concepção

Definição do escopo e objetivos do projeto.

## Elaboração

Planejamento e projeto da arquitetura.

## Construção

Implementação e teste do sistema.

## Transição

Implantação e treinamento dos usuários.



# Principais Conclusões e Próximos Passos

Os processos de software são essenciais para o desenvolvimento de sistemas de alta qualidade. A escolha do modelo adequado, o gerenciamento eficaz das atividades e a capacidade de lidar com as mudanças são fatores críticos para o sucesso. É importante conhecer os modelos, as atividades e os fatores críticos para escolher o melhor para cada projeto.



Entender os modelos



Gerenciar as atividades



Adaptar-se às mudanças



# EXERCÍCIOS

Descreva as principais atividades do processo de projeto de software e as saídas dessas atividades. Usando um diagrama, mostre as possíveis relações entre as saídas dessas atividades.

- Atividade em grupo;
- Enviar até 27/03/2025 - via Teams/arquivo .pdf.

# REFERÊNCIAS

Sommerville, Ian. **Engenharia de Software**. tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hiramã.  
— 9. ed. — São Paulo : Pearson Prentice Hall, 2011.

# Gabarito – Exercício

O processo de projeto de software envolve diversas atividades que têm como objetivo principal definir a estrutura e o comportamento de um sistema de software, atendendo aos requisitos funcionais e não funcionais especificados. Abaixo estão as principais atividades e suas respectivas saídas:

## 1. Análise de Requisitos

- **Atividade:** Identificar, analisar e documentar os requisitos do sistema.
- **Saídas:** Documento de requisitos, casos de uso, histórias de usuário.

## 2. Projeto de Arquitetura de Software

- **Atividade:** Definir a estrutura global do sistema, incluindo a divisão em módulos, componentes e subsistemas.
- **Saídas:** Diagrama de arquitetura, especificações de interfaces, visão geral dos componentes.

## 3. Projeto de Banco de Dados

- **Atividade:** Estruturar a modelagem dos dados que serão usados pelo sistema.
- **Saídas:** Diagrama entidade-relacionamento (DER), esquema do banco de dados, dicionário de dados.

## 4. Projeto de Interface de Usuário

- **Atividade:** Definir como será a interação do usuário com o sistema.
- **Saídas:** Protótipos de tela, wireframes, especificações de usabilidade.

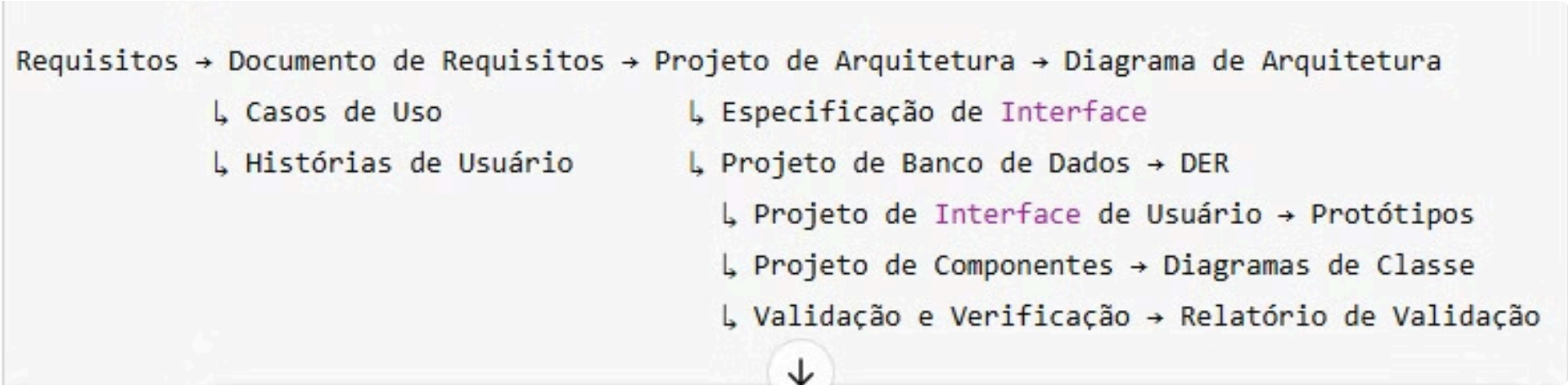
## 5. Projeto de Componentes e Módulos

- **Atividade:** Detalhar o comportamento interno de componentes específicos do sistema.
- **Saídas:** Diagramas de classes, diagramas de sequência, especificações de módulos.

## 6. Validação e Verificação

- **Atividade:** Avaliar se o projeto atende aos requisitos e se não há inconsistências.
- **Saídas:** Relatórios de validação, revisões de design, checklists de conformidade.

Imagem 01: Diagrama das Relações entre as Saídas



Fonte: Autor da atividade (2025)

**Atenção:** O diagrama mostra como as saídas de uma etapa alimentam as próximas, criando um fluxo coerente de desenvolvimento. **Agora você vai precisar elaborar um diagrama visual mais detalhado! Utilizar - Software de diagramas** (Exemplo: Visual Paradigm *Online*); Salvar e enviar via Teams.