

Modelagem de Software

Prof.: Sônia A Santana



Modelagem de Software



Carga Horária

160 horas

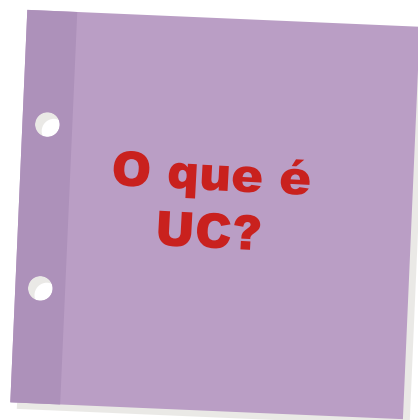
Área

Tecnologia

Cursos

Análise de Sistemas,
Sistemas de Informação,
Ciência da Computação

Introdução



“**Unidade Curricular** transforma seu currículo integrado em uma solução transformadora que eleva o processo do aprendizado a outro patamar: a abordagem em comunidades de aprendizagem multiprofissionais.”

Modelagem de Software

Agenda

01

Plano de Ensino

02

Avaliações e Notas

03

Frequência

04

Plataforma Ulife

05

Bibliotecas

06

Oportunidades Una

07

Contatos importantes

Modelagem de Software: Ementa

- Análise de Problema
- Análise de Requisitos
- Modelagem Orientada a Objetos
- Modelagem de Banco de Dados

Modelagem de Software

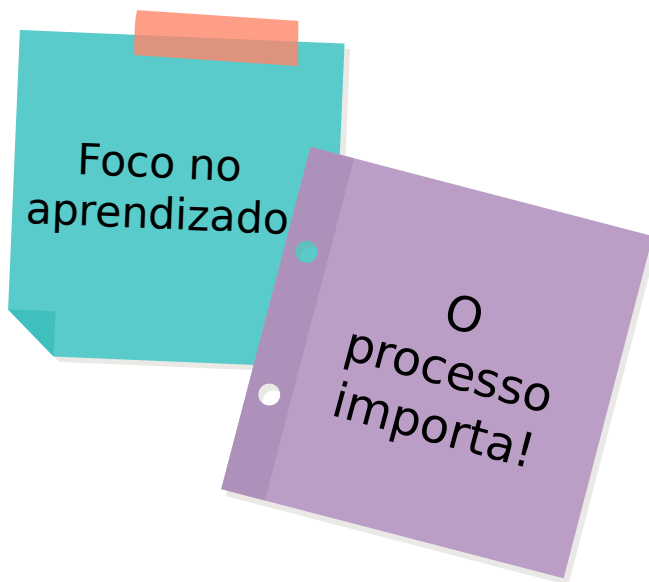
Metas de Compreensão

O que
vamos
aprender?

- Analisar problemas avaliando as necessidades dos clientes.
- Criar a especificação de software, elicitando os requisitos funcionais e não funcionais do software em conformidade com os requisitos do usuário.
- Utilizar ferramentas de prototipagem de software e aplicar os tipos de prototipagem conforme o projeto.
- Criar modelos de sistemas de software utilizando o paradigma orientado a objetos a partir dos principais diagramas da UML.
- Especificar modelos conceituais de banco de dados, analisando aspectos do mundo real a serem tratados pelos sistemas de informação e representando-os corretamente de acordo com o metamodelo selecionado e integrando-os com as diretrizes de administração de dados da organização.
- Criar modelos lógicos e físicos de banco de dados de acordo com os propósitos das necessidades do sistema de informação, especificando estruturas e mecanismos de armazenamento, busca e recuperação de dados.

Avaliações

Como seremos avaliados?



Avaliação A1 – 30 pontos

Dissertativa – avalia a expressão da linguagem da área, códigos e signos



Avaliação A2 – 30 pontos

Múltipla escolha – avalia leitura, interpretação, análise e estabelecimento de relações



Produto A3 – 40 pontos

Avaliação dos desempenhos como resultado do processo

Busca ativa

Critérios de Aprovação

01

Nota final acima de 70 pontos

Somando as notas A1 + A2 + A3 o aluno deve atingir um mínimo de 70 pontos!

02

Frequência de mínimo 75%

O aluno precisa ter um mínimo de 75% de presença!

Segunda Chance

Avaliação Integrada (AI)



Substitutiva

Substitui a menor nota em A1 **OU** A2

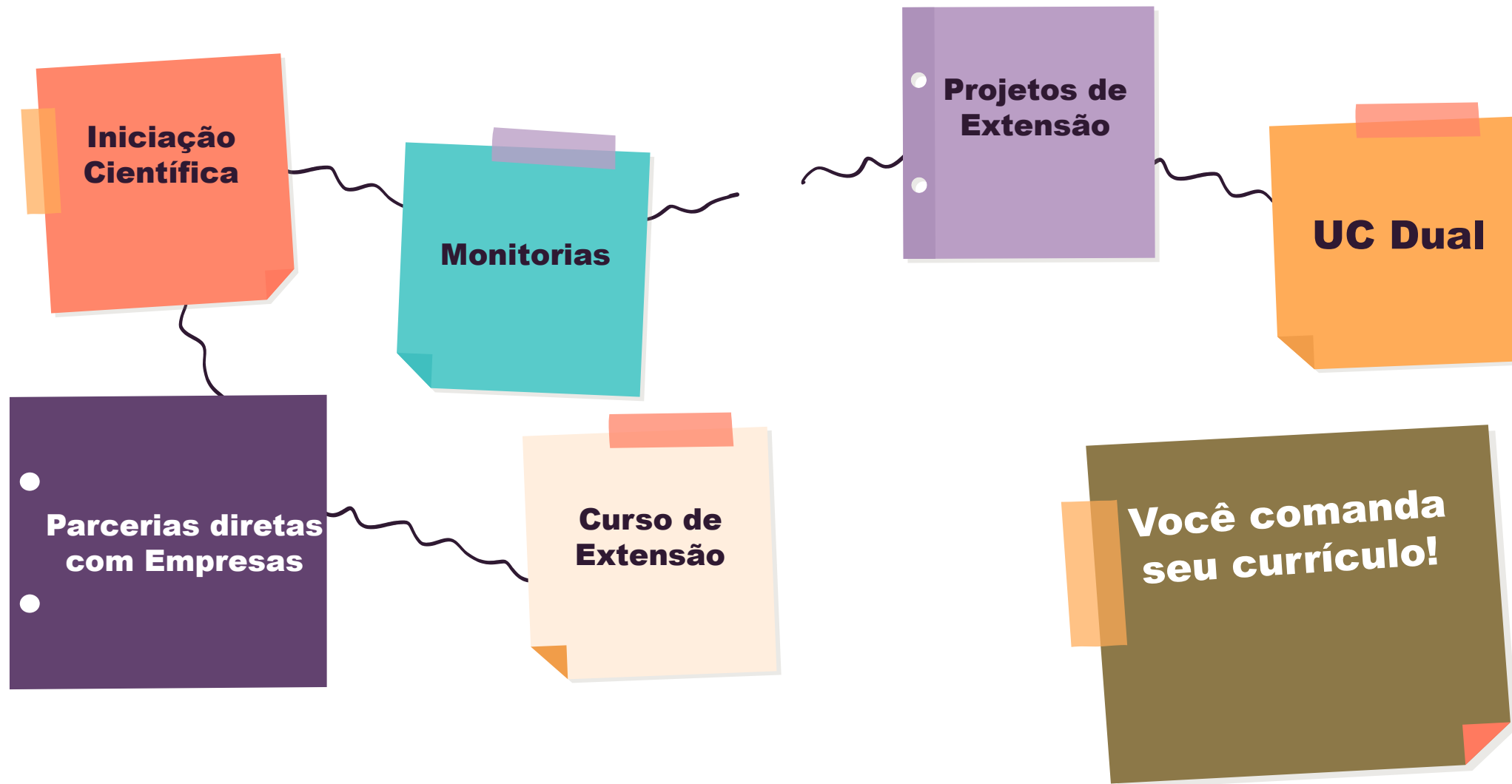
Elegíveis

Alunos com **mínimo de 40 pontos em A3 + A1 ou A2**

Valor

Valor de 30 pontos

Vamos falar de Oportunidades?



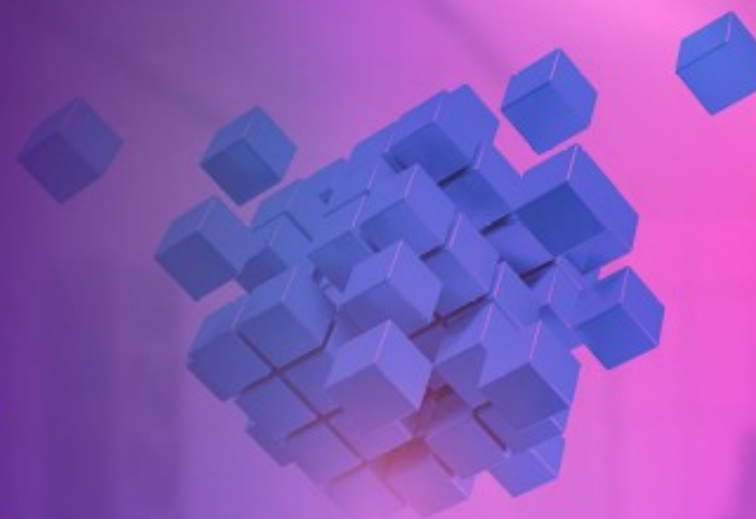
Maiores Detalhes

**Isso e muito mais
você encontra no
Manual do Aluno**



TRANSFORMANDO-SE
PARA 
TRANSFORMAR

**Vamos Juntos fazer
um excelente
semestre!**



ănima
anima

Introdução

A maior empresa de táxi, não tem carros

Uber

A maior empresa de hospedagem, não tem hotéis

 **airbnb**

Os maiores varejistas, não têm lojas

amazon

O maior cinema, não tem salas

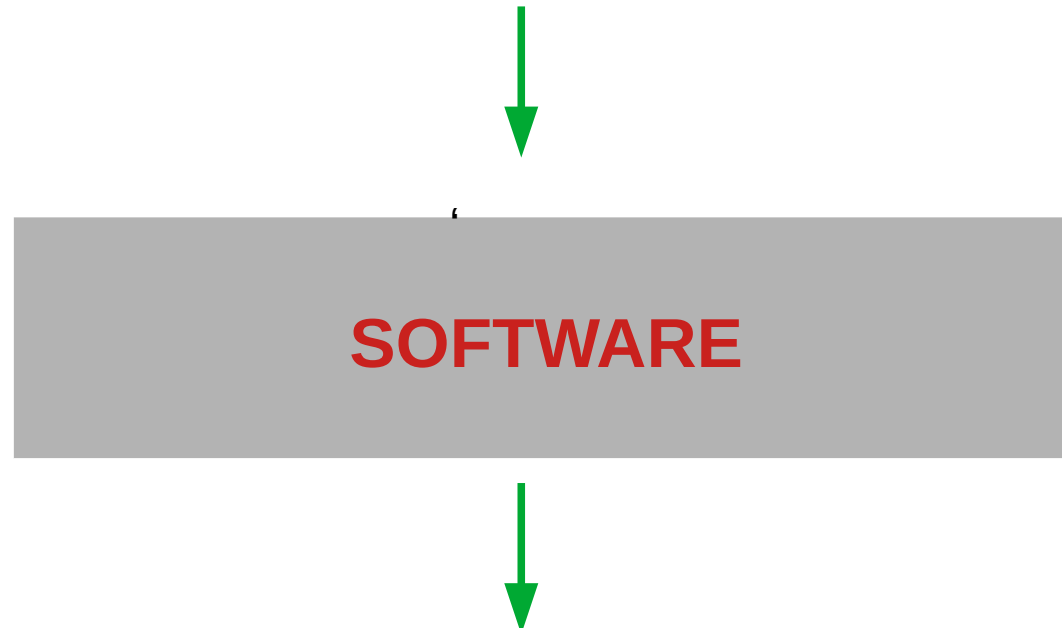
NETFLIX

Introdução

- Vocês já pararam para pensar que, no mundo atual, a maior empresa de taxi não tem carros?
- A maior empresa de hospedagem, não possui hotéis?
- Os maiores varejistas como a Amazon, por exemplo, não tem nenhuma loja física;
- O maior cinema, a Netflix, não tem nenhuma sala de cinema física.
- O que então eles têm em comum?

Introdução

- O que elas possuem em comum?
- Resposta: O principal recurso que as move como empresas, como negócio:



Ou seja: os aplicativos que essas empresas desenvolveram para atender a uma certa necessidade de negócio.

O que é um Software

Software



Conjunto de instruções que devem ser seguidas e executadas por um computador ou um aparato eletromecânico.

Outra Definição

- *"Software são programas de computadores com um **documentação associada** e que os produtos de software podem ser desenvolvidos para um determinado cliente ou para um mercado mais generalizado."*

Sommerville (2011)

Atributos de um bom Software

Manutenabilidade	Deve ser escrito de forma que possa evoluir para atender às necessidades dos clientes. É um atributo crítico, pois a mudança em um software é inevitável em um ambiente de negócio em mudança.
Confiança e Proteção	Um software não deve causar prejuízos físicos ou econômicos no caso de falha; usuários maliciosos ou não autorizados não devem ser capazes de acessar e/ou prejudicar o sistema.
Eficiência	Não deve desperdiçar os recursos do sistema (memória, ciclo de processamento). Inclui capacidade de resposta, tempo de processamento, uso de memória/disco, etc.
Aceitabilidade	Deve ser aceitável para o tipo de usuário para o qual foi projetado. Deve ser compreensível, usável e compatível aos demais sistemas usados por ele.

Ciclo de Vida de Software

- O ciclo de vida de um software é uma estrutura que indica processos e **atividades envolvidas no desenvolvimento**, operação e manutenção de um software, abrangendo de fato toda a vida do sistema.
- Neste ciclo, **existem modelos** que definem como o software será desenvolvido, lançado, aprimorado e finalizado.

Modelagem de Software

- Por que modelar software?
 - Ajuda a ter uma visão geral do sistema;
 - Permite especificar a estrutura e o comportamento do sistema;
 - Proporciona um guia para a construção do software
 - Documenta as decisões tomadas

Processo de Desenvolvimento de Software

- Podemos entender o processo de software como um conglomerado de atividades, políticas, normas, padrões, processos, tecnologias, procedimentos e conhecimentos utilizados em conjunto para desenvolver um sistema computacional

SOMMERVILLE, 2011.

Processo de Desenvolvimento de Software

- Existem vários processos de desenvolvimento de software diferentes mas todos envolvem:
 - Especificação – definição do quê o sistema deve fazer;
 - Projeto e implementação – definição da organização do sistema e implementação do sistema;
 - Validação – checagem de que o sistema faz o que o cliente deseja;
 - Evolução – evolução em resposta a mudanças nas necessidades do cliente.

Processo de Desenvolvimento de Software

- Descrições de processos também podem incluir:
 - **Produtos**, que são os resultados de uma atividade do processo;
 - **Papéis**, que refletem as responsabilidades das pessoas envolvidas no processo;
 - **Pré e pós-condições**, que são declarações que são verdadeiras antes e depois de uma atividade do processo ser executada, ou um produto produzido.

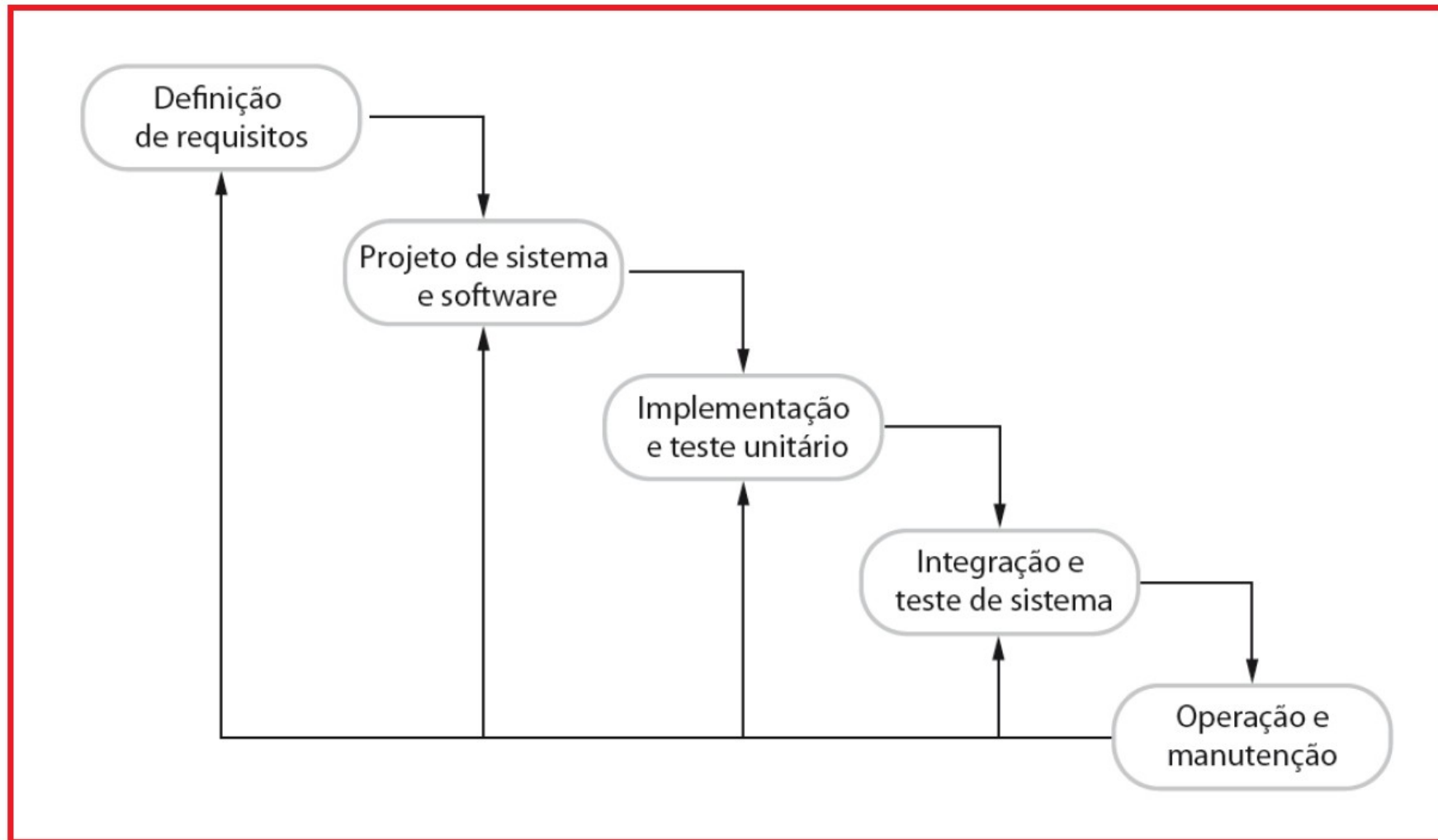
Não existe processo de software certo ou errado

Modelos de Processos de Software

- Representação simplificada de um processo de software;
- Modelos mais comuns:
 - **Cascata** (sequencial, *water fall*, preditivo): modelo dirigido a planos. Fases de especificação e desenvolvimento separadas e distintas.
 - **Incremental** (iterativo e incremental, evolucionário:): Especificação, desenvolvimento e validação são intercaladas.
 - **Orientado a reuso**: o sistema é montado a partir de componentes já existentes.

DETALHANDO....

Modelo Cascata



Modelo Cascata

- **Análise e definição de requisitos:** Os serviços, restrições e metas do sistema são estabelecidos por meio de consulta aos usuários.
- **Projeto de sistema e software:** Aloca os requisitos tanto para sistemas de hardware como para sistemas de software, por meio da definição de uma arquitetura geral do sistema.
- Identificação e descrição das abstrações fundamentais do sistema de software e seus relacionamentos.

Modelo Cascata

Implementação e teste unitário: Desenvolvido como um conjunto de programas ou unidades de programa.

O teste unitário envolve a verificação de que cada unidade atenda a sua especificação.

Modelo Cascata

Integração e teste de sistema: As unidades individuais do programa são integradas e

Realiza-se testes como um sistema completo para assegurar que os requisitos do software tenham sido atendidos.

Entrega para o cliente.

Modelo Cascata

Operação e manutenção: O sistema é instalado e colocado em uso.

A manutenção envolve a correção de erros que não foram descobertos em estágios iniciais e possivelmente serão demonstrados novas necessidades.

Vantagens do Modelo Cascata

- Documentação rígida (idealmente completa) em cada atividade
- Reflete abordagens adotadas em outras engenharias
- Aderência a outros modelos de processo
 - Pode ser combinado a outros modelos

Desvantagens do Modelo Cascata

- Projetos reais raramente seguem um fluxo sequencial
- Em geral, é difícil para o cliente estabelecer todos os requisitos à priori
- Difícil se adequar a mudanças inevitáveis de requisitos
- Uma versão executável somente ficará pronta na fase final do projeto

Quando Aplicar o Modelo Cascata

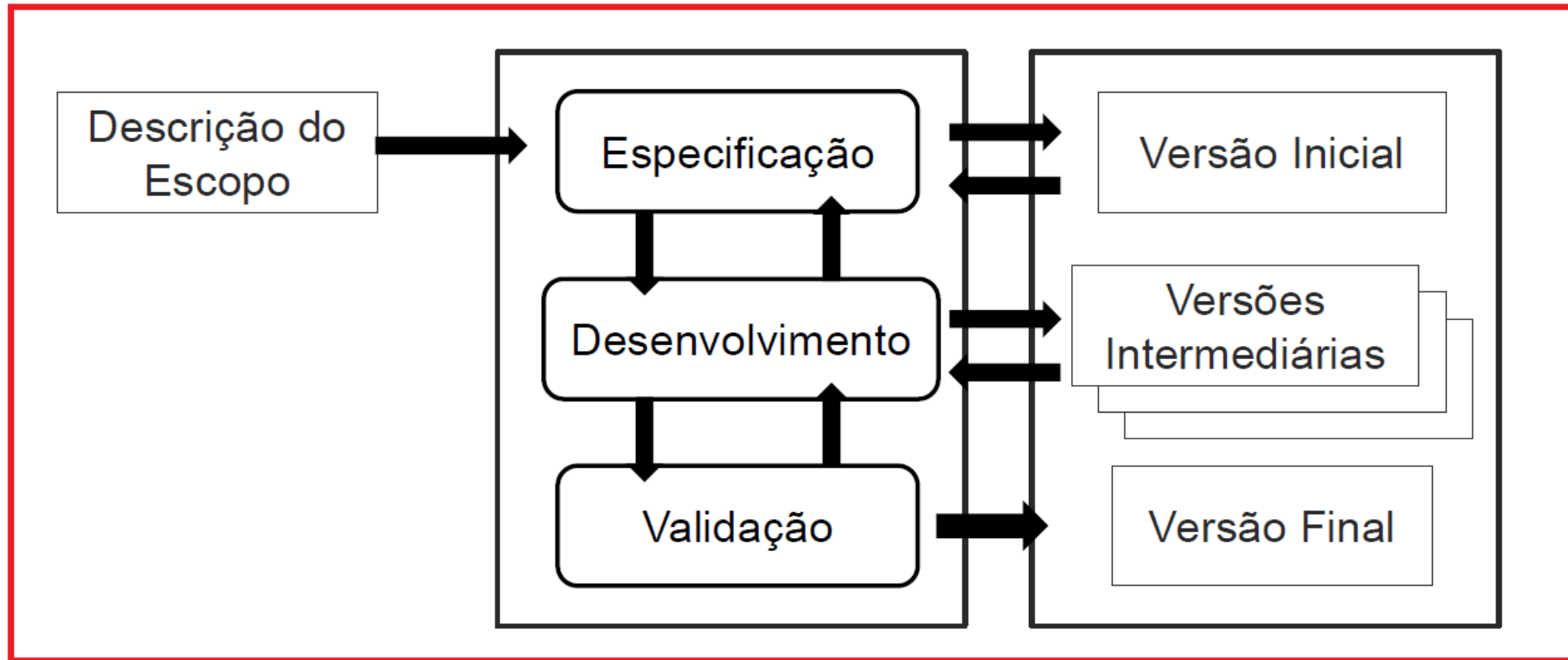
- Sistemas críticos
- Quando os requisitos são bem compreendidos
- Quando há pouca probabilidade dos requisitos mudarem
- Projetos de engenharia de grandes sistemas onde o sistema é desenvolvido em vários locais.
 - Nessas circunstâncias, a natureza do modelo cascata dirigida a planos ajuda a coordenar o trabalho

Exercício de Fixação

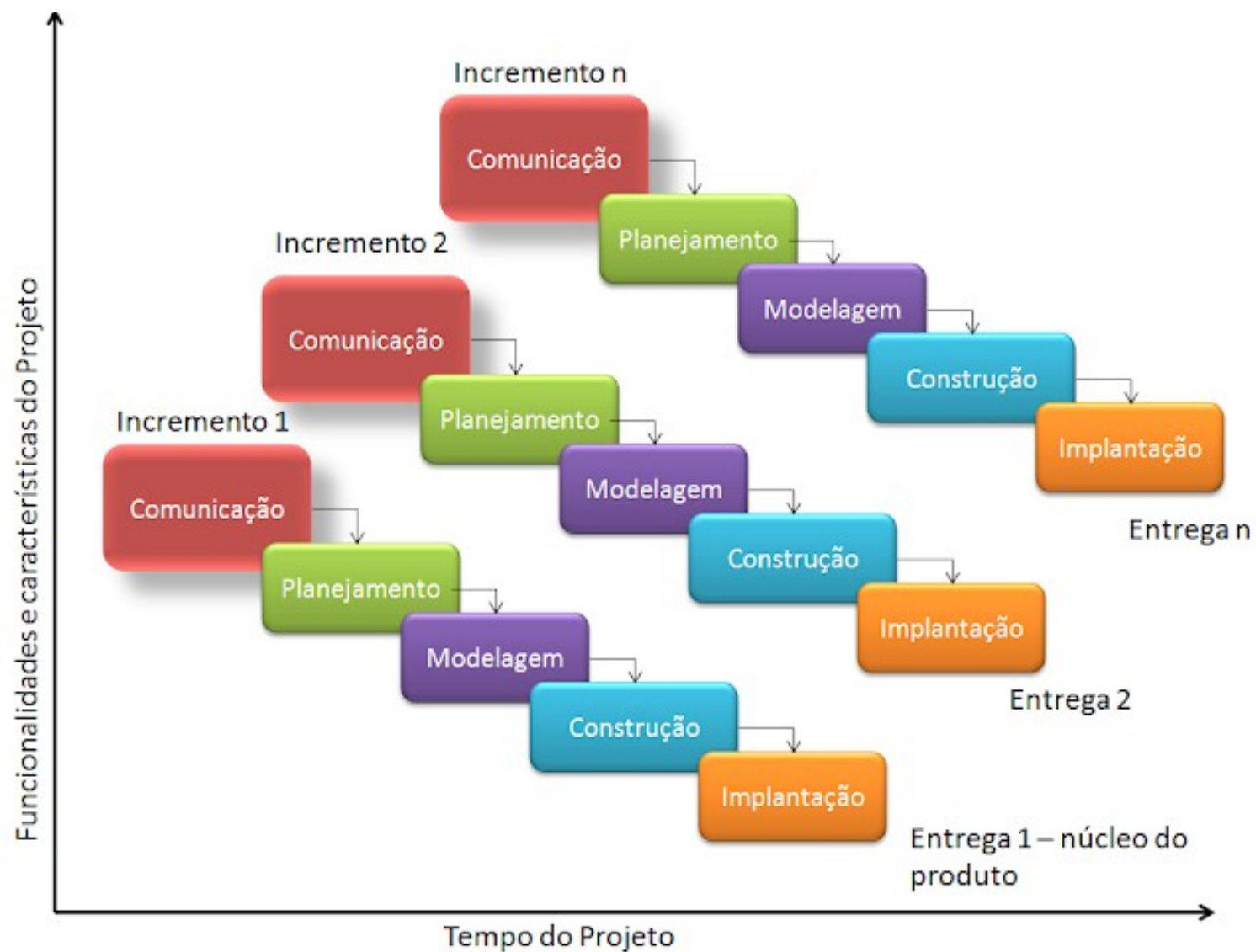
- Julgue como Certo ou Errado:
- a. () O modelo de ciclo de vida em cascata tem como características o estabelecimento, no início do projeto, de requisitos de maneira completa, correta e clara, e a **possibilidade de disponibilização de várias versões operacionais do software antes da conclusão do projeto.**
- b. () O modelo de desenvolvimento em cascata é utilizado em caso de divergência nos requisitos de um software, para permitir a evolução gradual do entendimento dos requisitos durante a implementação do software.
- c. () **Durante a fase de levantamento de requisitos para a construção de um software, compete aos desenvolvedores organizar as necessidades em ordem de prioridade.**

Modelo Incremental

- Atividades são intercaladas
- Objetivo: dar feedback rápido ao cliente



Modelo Incremental



Iterativo x Incremental

Incremental



Iterativo



Iterativo x Incremental

- Uma diferença importante é se você especifica os requisitos de forma antecipada (incremental) ou os descobre ao construir seu produto (iterativo).
- Um processo de desenvolvimento de software é incremental quando a cada rodada é desenvolvido um pedaço inteiro do software. Já no iterativo, a cada iteração, se avança no conhecimento do projeto, novos requisitos são elicitados e a arquitetura do software é revisada.

Exercícios de Fixação

- Assinale a alternativa que traz a informação correta sobre o modelo iterativo e o modelo incremental.
 - a) O modelo iterativo sempre disponibiliza uma versão que pode ser utilizada pelo usuário final.
 - b) No modelo incremental, iterações são realizadas e, ao fim de cada uma delas, diversas partes do sistema podem ter sido desenvolvidas e não concluídas.
 - c) O modelo cascata e o modelo incremental são muito semelhantes, uma vez que, em ambos ciclos de desenvolvimento, um módulo só pode ser entregue após a conclusão dos demais que já estavam em andamento.
 - d) O modelo iterativo deixa livre que diversos módulos sejam iniciados na mesma iteração sem a devida conclusão ao fim do ciclo.

Vantagens do Modelo Incremental

- Custo de acomodar mudanças nos requisitos é reduzido
- Mais fácil obter feedback do cliente
- Permite trabalhar com o cliente o entendimento dos requisitos
- Pode-se começar o sistema pelas partes melhor entendidas

Desvantagens do Modelo Incremental

- O processo pode não ser muito claro
- A gerência do software é complicada
 - O sistema não é completamente especificado à priori
- A estrutura do produto tende a se corromper com a adição de incrementos
 - O produto final é pode se tornar mal estruturado

Reflexão

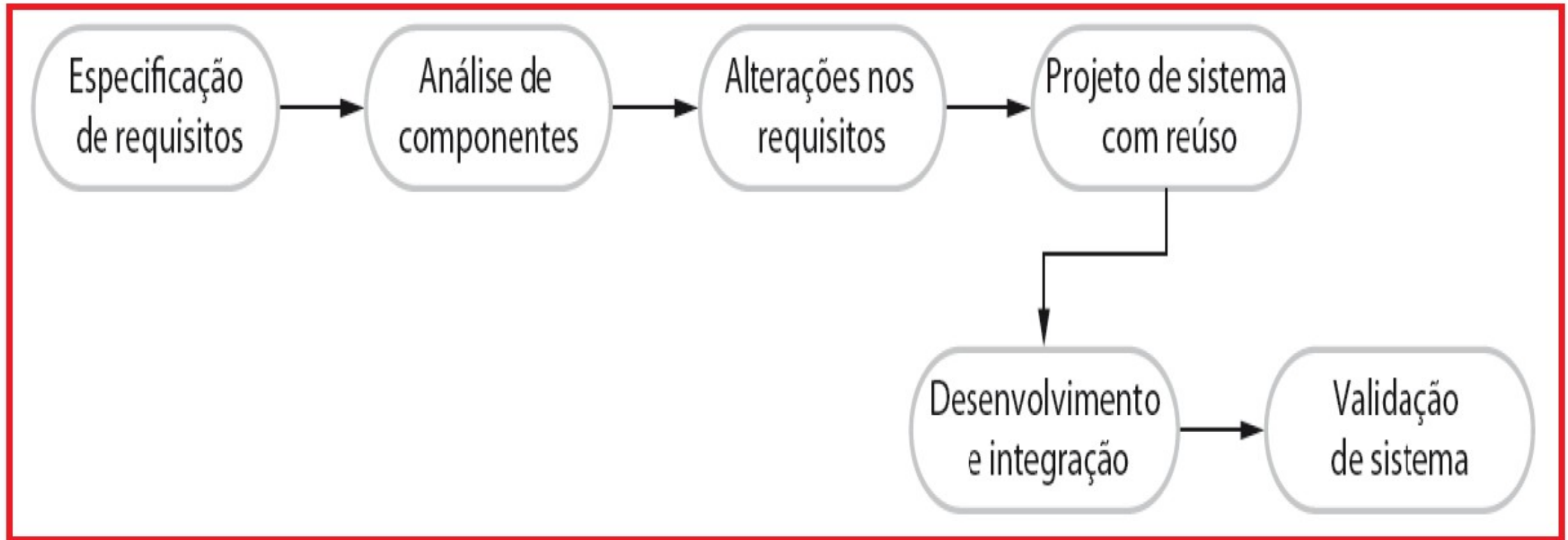
- Por que os problemas de desenvolvimento incremental se tornam mais graves quando se trata de desenvolvimento de sistemas de tempo real?



Engenharia de software orientada a reúso

- Baseada no reuso sistemático em que os **sistemas** são **integrados** com **componentes existentes** ou sistemas COTS (Commercial-off-the-shelf).
- Estágios do processo:
 - ✓ Análise de componentes;
 - ✓ Modificação de requisitos;
 - ✓ Projeto de sistema com reuso;
 - ✓ Desenvolvimento e integração.
- Atualmente, o reuso é a abordagem padrão para a construção de vários tipos de sistemas de negócio.

Engenharia de Software orientada a Reuso



Engenharia de software orientada a reúso

Análise de componentes: Dada a especificação de requisitos, é feita uma busca por componentes para implementar essa especificação.

Em geral, não há correspondência exata, e os componentes que podem ser usados apenas fornecem alguma funcionalidade necessária.

Engenharia de software orientada a reúso

Modificação de requisitos: Durante esse estágio, os requisitos são analisados usando-se informações sobre os componentes que foram descobertos.

Em seguida, estes serão modificados para refletir os componentes disponíveis.

No caso de modificações impossíveis, a atividade de análise dos componentes pode ser reinserida na busca por soluções alternativas.

Engenharia de software orientada a reúso

Projeto do sistema com reúso: Durante esse estágio, o framework do sistema é projetado ou algo existente é reusado.

Os projetistas têm em mente os componentes que serão reusados e organizam o framework para reúso.

Alguns softwares novos podem ser necessários, se componentes reusáveis não estiverem disponíveis.

Engenharia de software orientada a reúso

Desenvolvimento e integração: Softwares que não podem ser adquiridos externamente são desenvolvidos, e os componentes e sistemas são integrados para criar o novo sistema.

A integração de sistemas, nesse modelo, pode ser parte do processo de desenvolvimento, em vez de uma atividade separada.

Tipos de Componentes

- Web services que são desenvolvidos de acordo com padrões de serviço e ficam disponíveis para chamada remota.
- Coleções de objetos que são desenvolvidas como um pacote para ser integrado com um framework como .NET ou J2EE.
- Sistemas de software stand-alone (COTS) que são configurados para uso em ambientes específicos.

Exercício de Fixação

- “Não existe um modelo de processo de desenvolvimento de software que seja adequado a todo e qualquer projeto. Ao contrário, processos devem ser definidos caso a caso”. Você concorda com essa afirmação? Justifique sua resposta e caso tenha concordado cite quais fatores influenciam a definição de um modelo de processo de software que será adotado no desenvolvimento de um produto de software

Exercício de Fixação

- Justificando sua resposta com base no tipo de sistema a ser desenvolvido, sugira o modelo de processo de software mais apropriado que poderia ser usado como base para gerenciar o desenvolvimento dos seguintes sistemas:
- a. um sistema para controlar um antibloqueador de freios de um automóvel;
- b. um sistema de realidade virtual para apoiar a manutenção de carros;
- c. Um sistema de contabilidade de uma universidade, que substituirá um já existente;
- d. Um sistema interativo que permite aos passageiros encontrar os horários dos trens por meio de terminais instalados nas estações;

Referência Bibliográfica

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.