



Engenharia de Software I

Prof. Agnaldo Cieslak, MsC



Atividade em Sala

- Os grupos do projeto se reúnem para discutir, avaliar e elaborar a solução da questão abaixo:

Justificando sua resposta com base no tipo de sistema a ser desenvolvido, sugira o modelo genérico de processo de software mais adequado para ser usado como base para a gerência do desenvolvimento dos sistemas a seguir:

Um sistema para controlar o antibloqueio de frenagem de um carro.

Um sistema de realidade virtual para dar apoio à manutenção de software.

Um sistema de contabilidade para uma universidade, que substitua um sistema já existente.

Um sistema interativo de planejamento de viagens que ajude os usuários a planejar viagens com menor impacto ambiental.

Aula 2

~~Metodologias Ágeis~~ Processos de Software

Processos de software

O que é ?

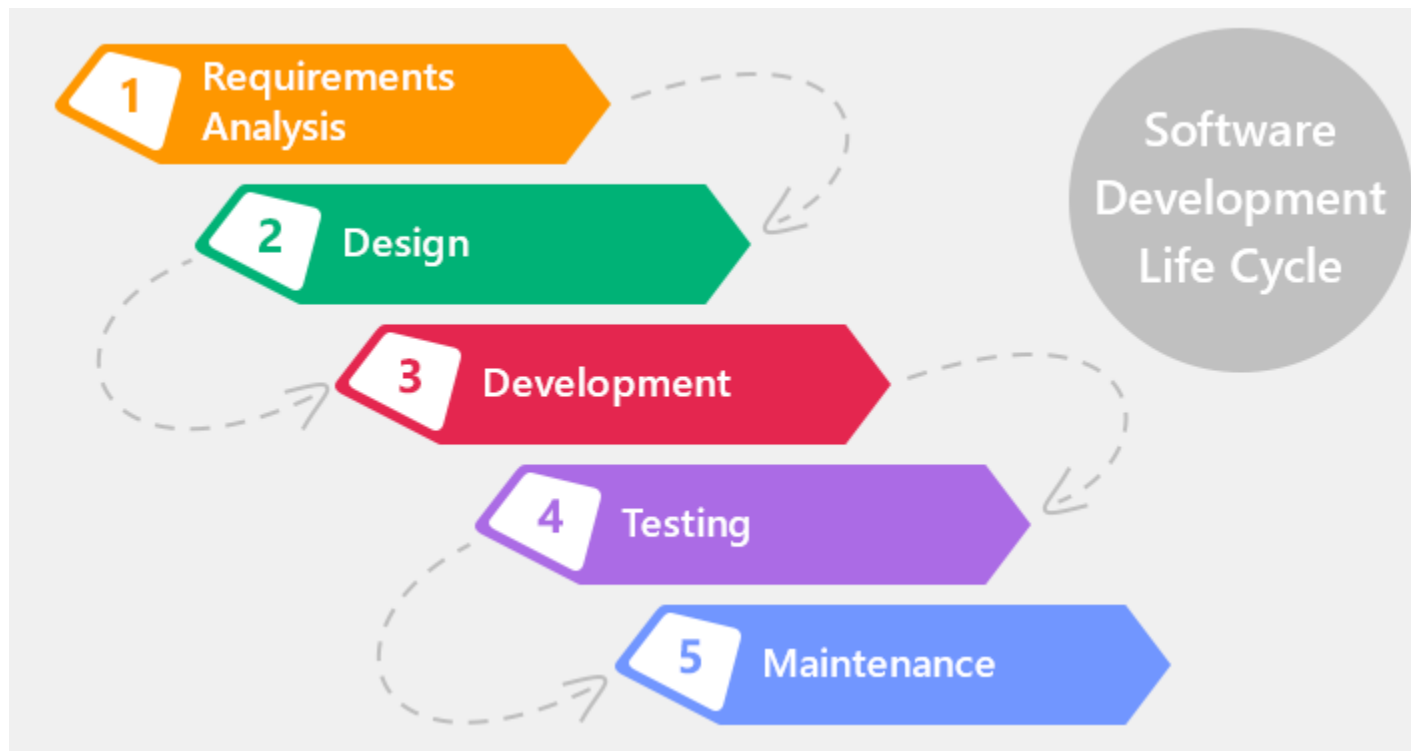
- Um processo de software é um **conjunto de atividades** que leva à **produção** de um **produto de software**.
- Essa atividade podem envolver o **desenvolvimento** de software propriamente dito, usando a **linguagem de programação** como Java ou C.
- Contudo, **novo software é desenvolvido** com a **ampliação** e a **modificação** de **sistemas existentes** e através da **configuração** e **integração** de software comercial ou **componentes** de sistemas.

Os **processos de software** são importantes, pois **estabelecem** para os membros da equipe de projeto **uma diretriz** de **o que** e **como** deve ser **feito** para atender aos **objetivos do software**.



Processos de software

Visão geral das etapas de desenvolvimento de um software



Pensando num Sistema

1. Quais seriam as **atividades** presentes em um Processo de Desenvolvimento ?
2. Esboce a **sequencia/ordem de execução** dessas atividades de desenvolvimento .
3. Existem **atividades acessórias** para a produção de software ?
4. Que **ferramentas** poderiam auxiliar a executar as atividades idealizadas ?

Converse com seu amigo do lado e responda as perguntas acima.



Pensando num Sistema

As atividades sugeridas por você anteriormente teriam diferenças nos seguintes cenários:

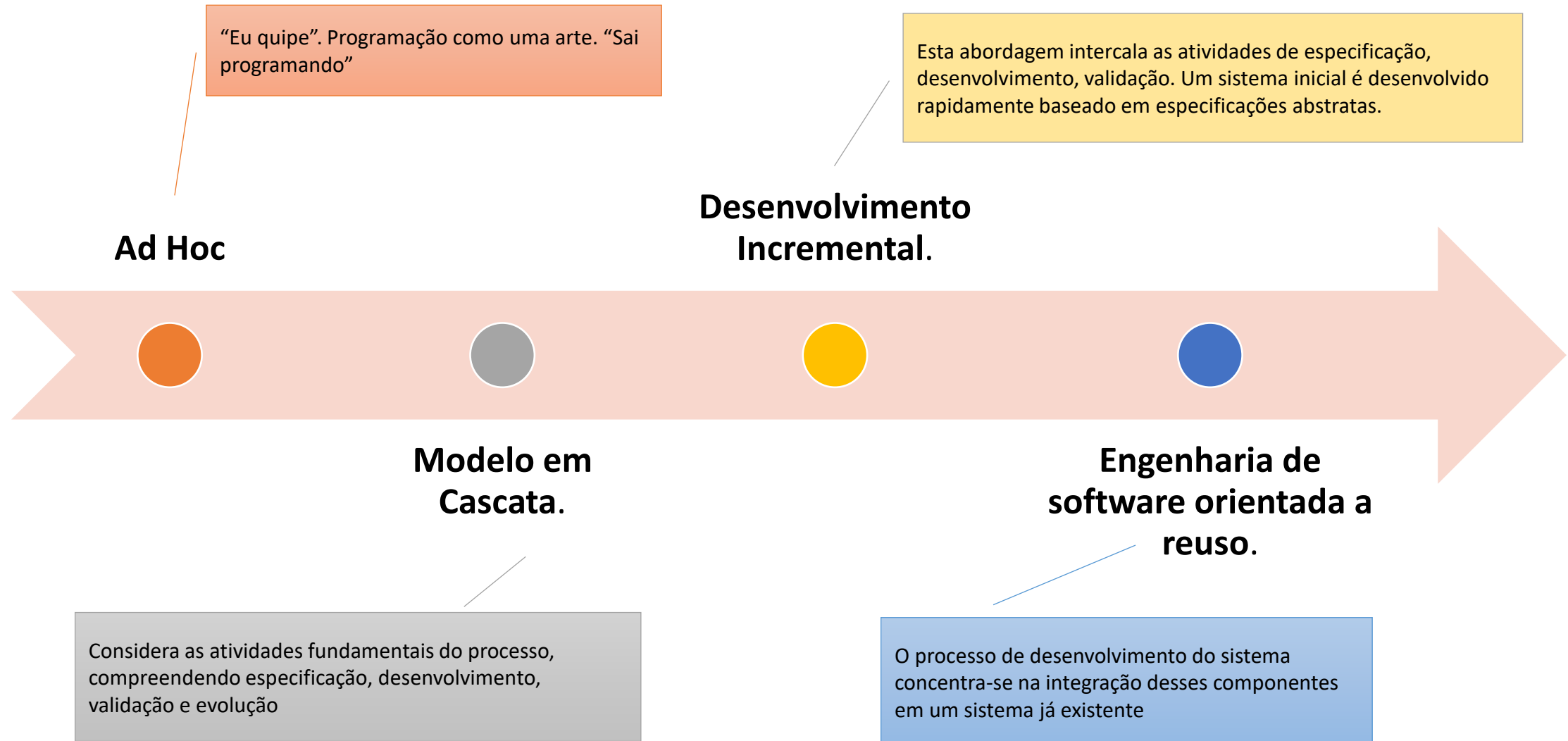
- a) Você Desenvolvendo sozinho
- b) Manutenção de um sistema, em uma TI de uma grande empresa.
- c) Em uma Fábrica de Software
- d) Desenvolvimento de Sistemas de Missão Crítica (Ex. Controle de Tráfego Aéreo)

Converse com seu amigo do lado e
responda as perguntas acima.



Modelos de Processo de Software

Evolução de Modelo de Processos de Software



Modelo de Processos de software

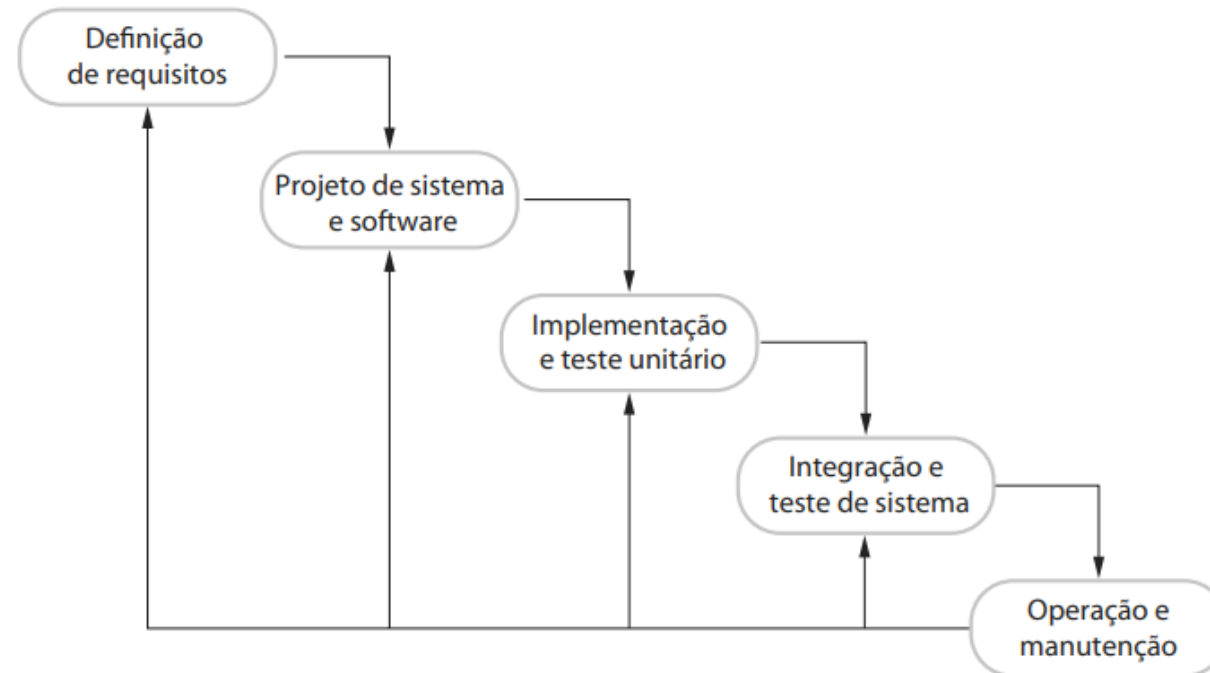
1. **“EuQuipe”** – não é um modelo!!! Programador adHOc
2. **Cascata.** Esse modelo considera as atividades fundamentais do processo de especificação, desenvolvimento, validação e evolução, e representa cada uma delas **como fases distintas**, como: especificação de requisitos, projeto de software, implementação, teste e assim por diante.
3. **Desenvolvimento incremental.** Essa abordagem intercala as atividades de especificação, desenvolvimento e validação. O sistema é desenvolvido como **uma série de versões** (incrementos), de maneira que **cada versão adiciona funcionalidade** à anterior
4. **Engenharia de software orientada a reuso.** Essa abordagem é baseada na existência de um número significativo de componentes reusáveis. O processo de **desenvolvimento** do sistema **concentra-se na integração desses componentes** em um sistema já existente em vez de desenvolver um sistema a partir do zero.

Vamos olhar cada um destes modelos...

O Modelo em Cascata

- **Primeiro modelo** do processo de desenvolvimento de software a ser publicado
- Derivado de processos mais gerais da engenharia de sistemas
- Executado em **fases encadeadas**

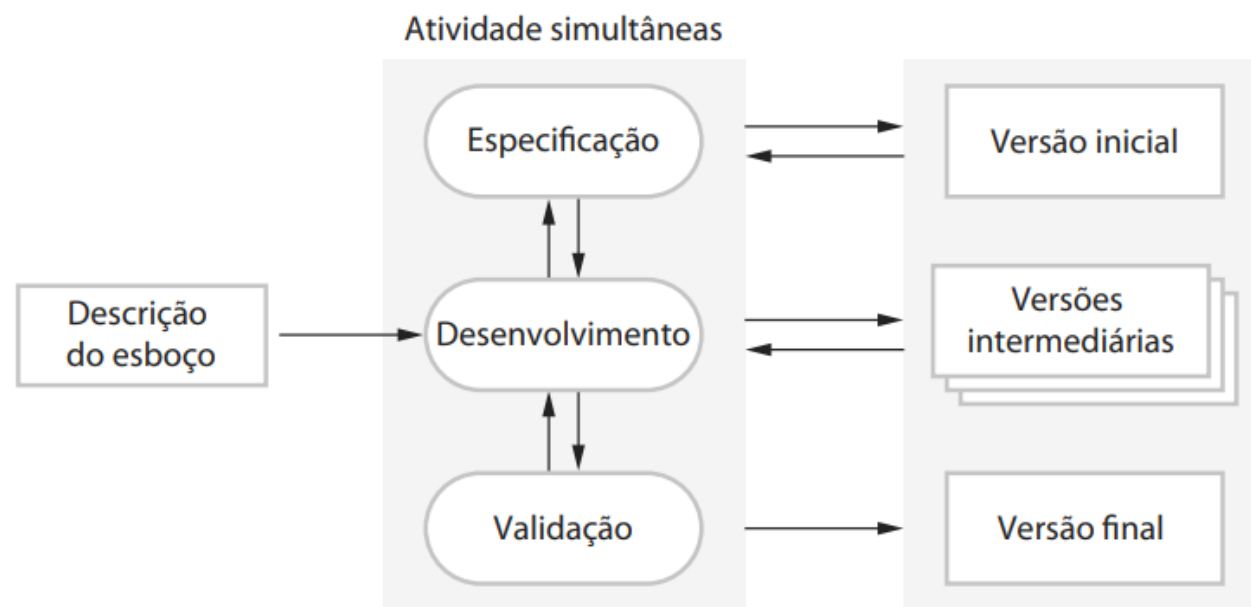
O modelo em cascata



Desenvolvimento Incremental

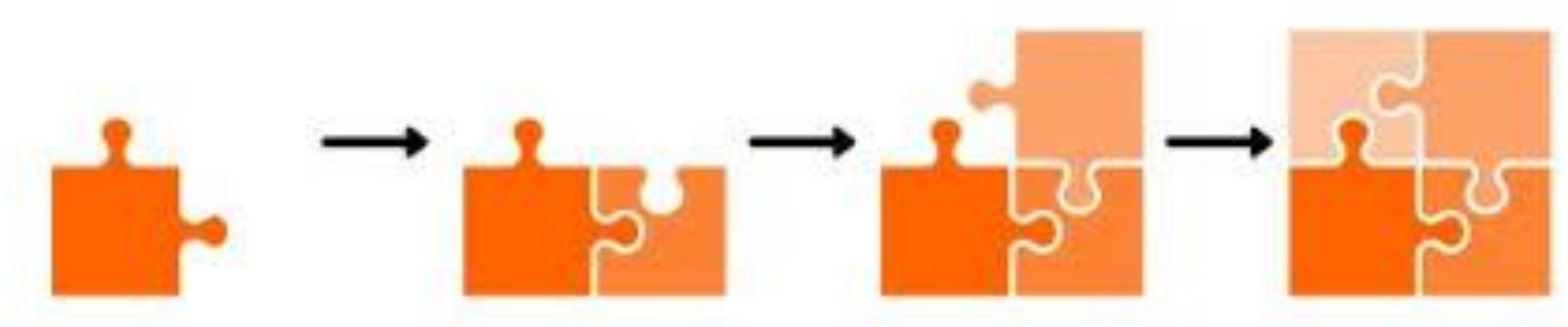
- Baseado na ideia de desenvolver uma **implementação inicial**, **expô-la** aos comentários dos **usuários** e continuar por meio da **criação de várias versões** até que um sistema adequado seja desenvolvido

Desenvolvimento incremental



- Custo** de acomodar as **mudanças** nos requisitos do cliente é **reduzido**
- Mais **fácil** obter **feedback** dos clientes
- É possível obter **entrega** e implementação **rápida** de um software útil ao **cliente**

Ciclo de vida incremental



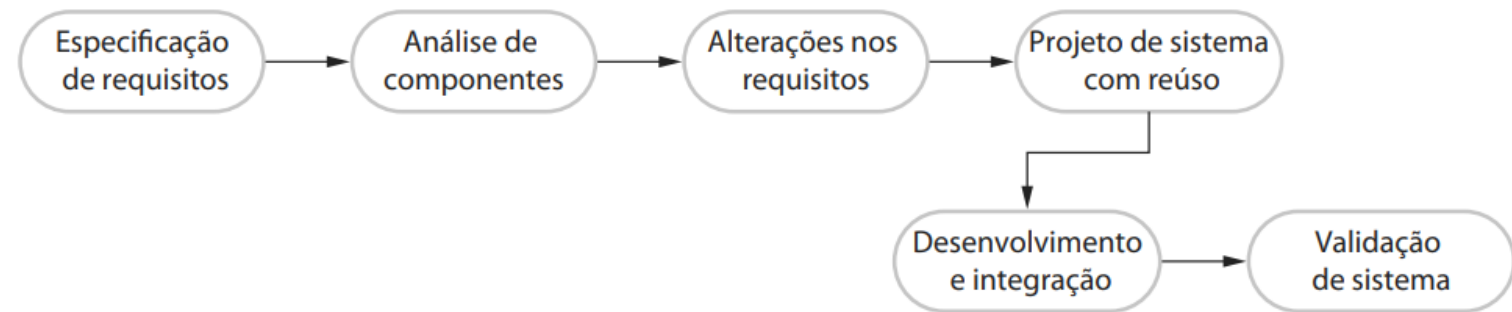
Desenvolvimento Incremental

- Várias versões do produto (Software)

Engenharia de Software Orientada a Reuso

- Na **maioria dos projetos** de software, **há algum reuso** de software.
- Isso acontece muitas vezes informalmente, quando as pessoas envolvidas no projeto sabem de projetos ou códigos semelhantes ao que é exigido.
- Elas os **buscam**, fazem as **modificações** necessárias e **incorporam-nos** a seus sistemas

3 Engenharia de software orientada a reuso



Engenharia de Software Orientada a Reuso

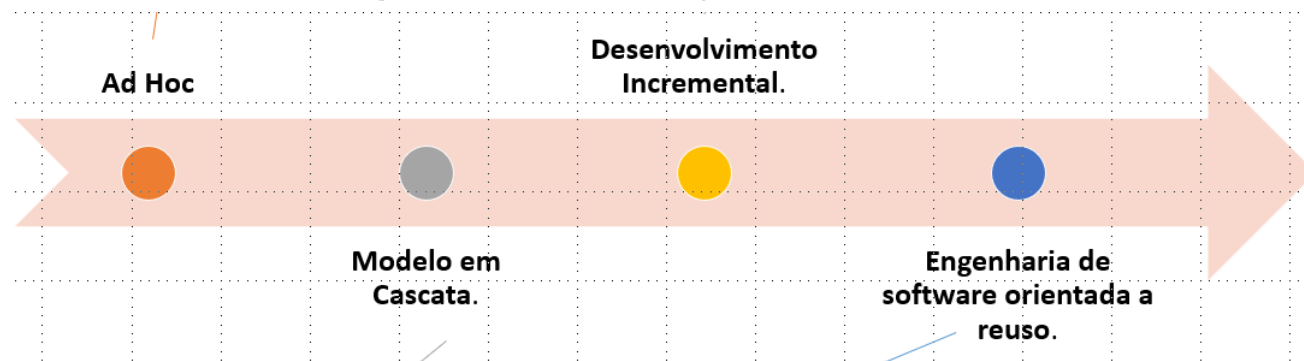
- Vantagens da Engenharia de Software Orientada a Reuso:
- Produtividade
- Eficiência
- Valor agregado
- + colaboradores



Pensando num Modelo...

Para cada “bullet” da linha do slide de modelos de Software descreva os seguintes pontos:

- a) Qual a característica?
- b) Qual a vantagem em relação ao anterior?



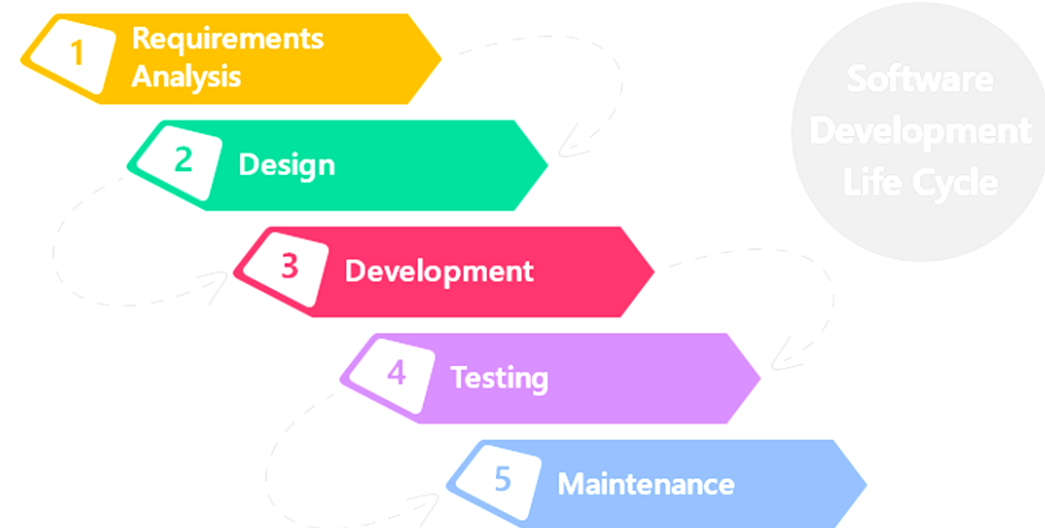
Converse com seu amigo do lado e
responda as perguntas acima.



Atividades do Processo De Software

Processos de software

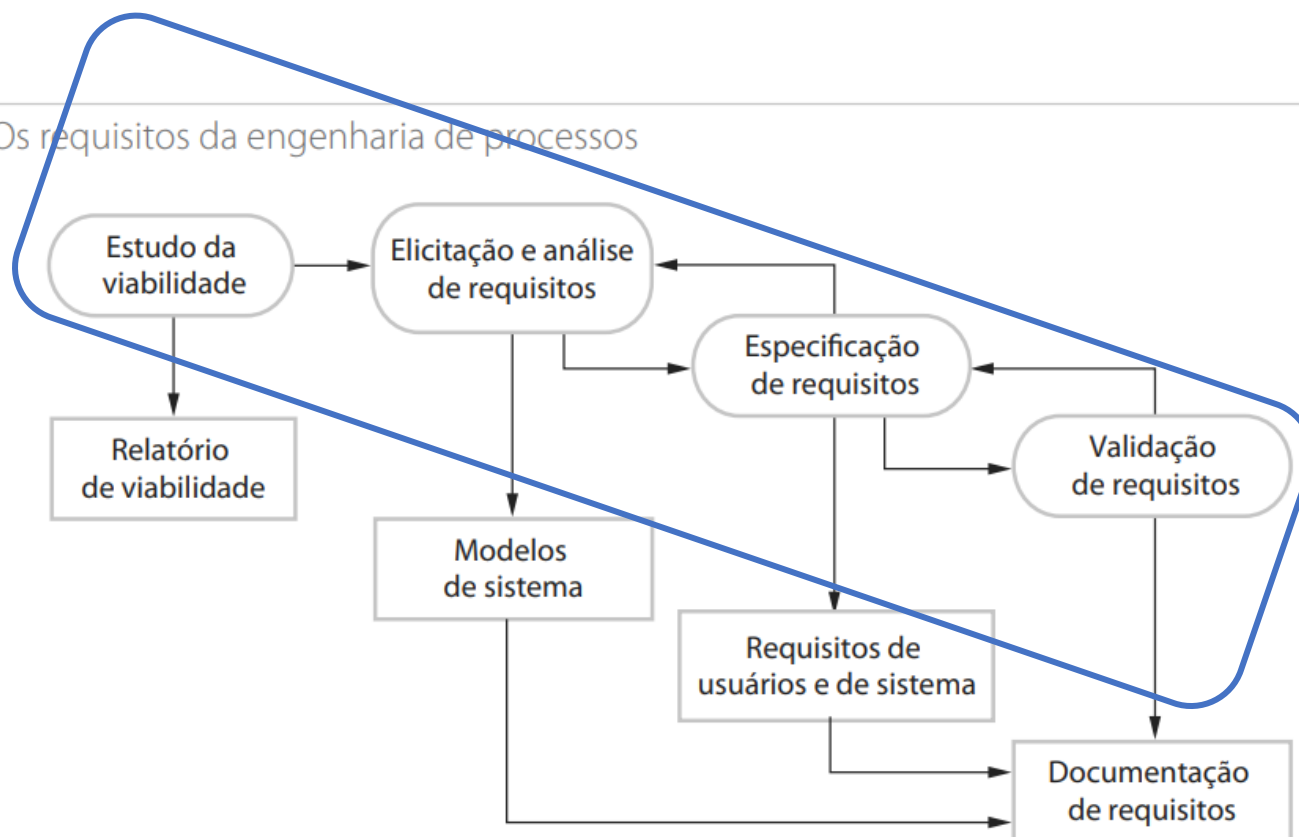
- As quatro atividades básicas do processo — **especificação, desenvolvimento, validação e evolução** — são organizadas de forma diferente conforme o processo de desenvolvimento.



Especificação de software

- Especificação de software ou engenharia de requisitos é o processo de **compreensão** e **definição** dos **serviços requisitados** do sistema e **identificação** de **restrições** relativas à operação e ao desenvolvimento do sistema.

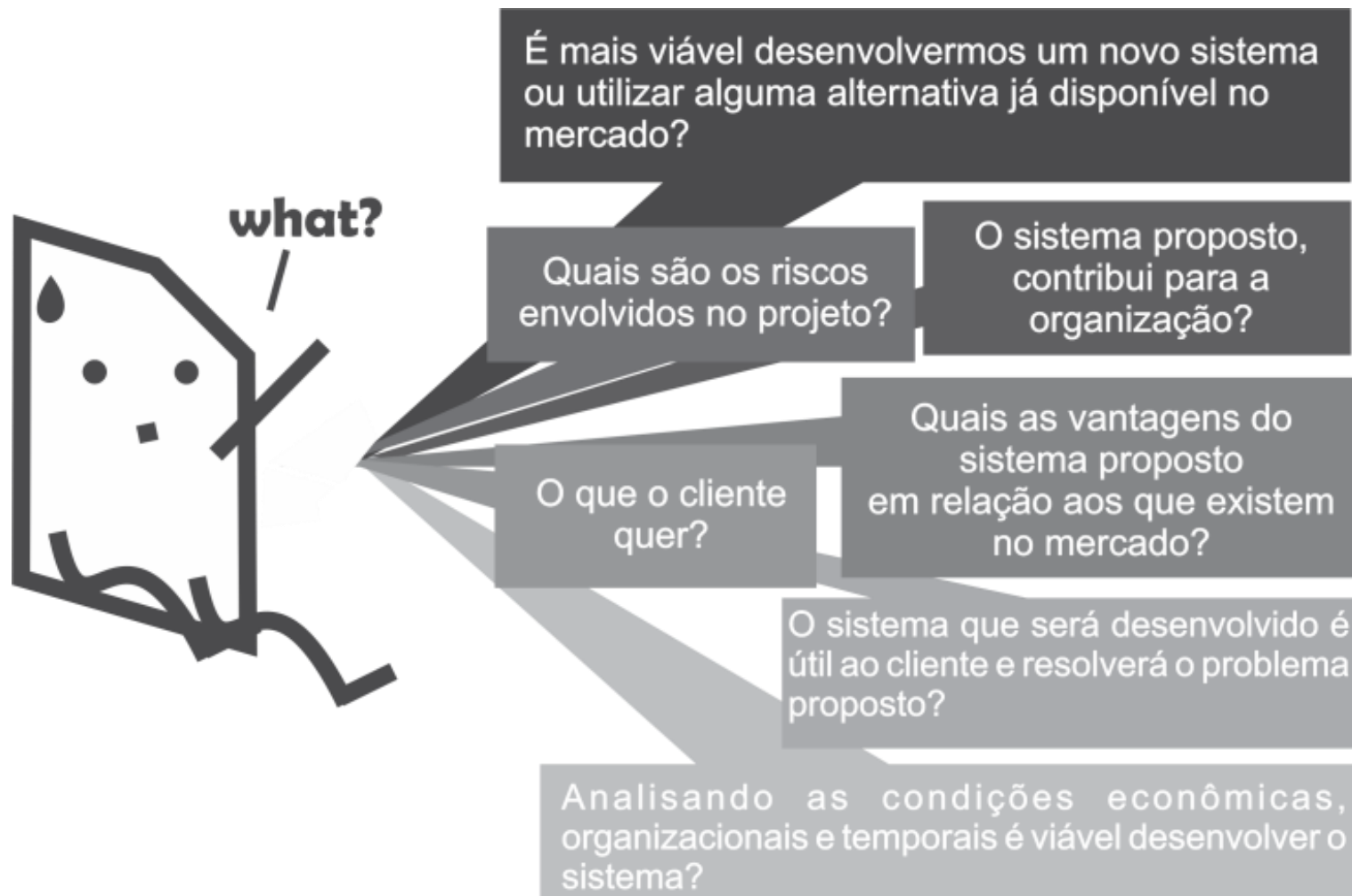
Os requisitos da engenharia de processos



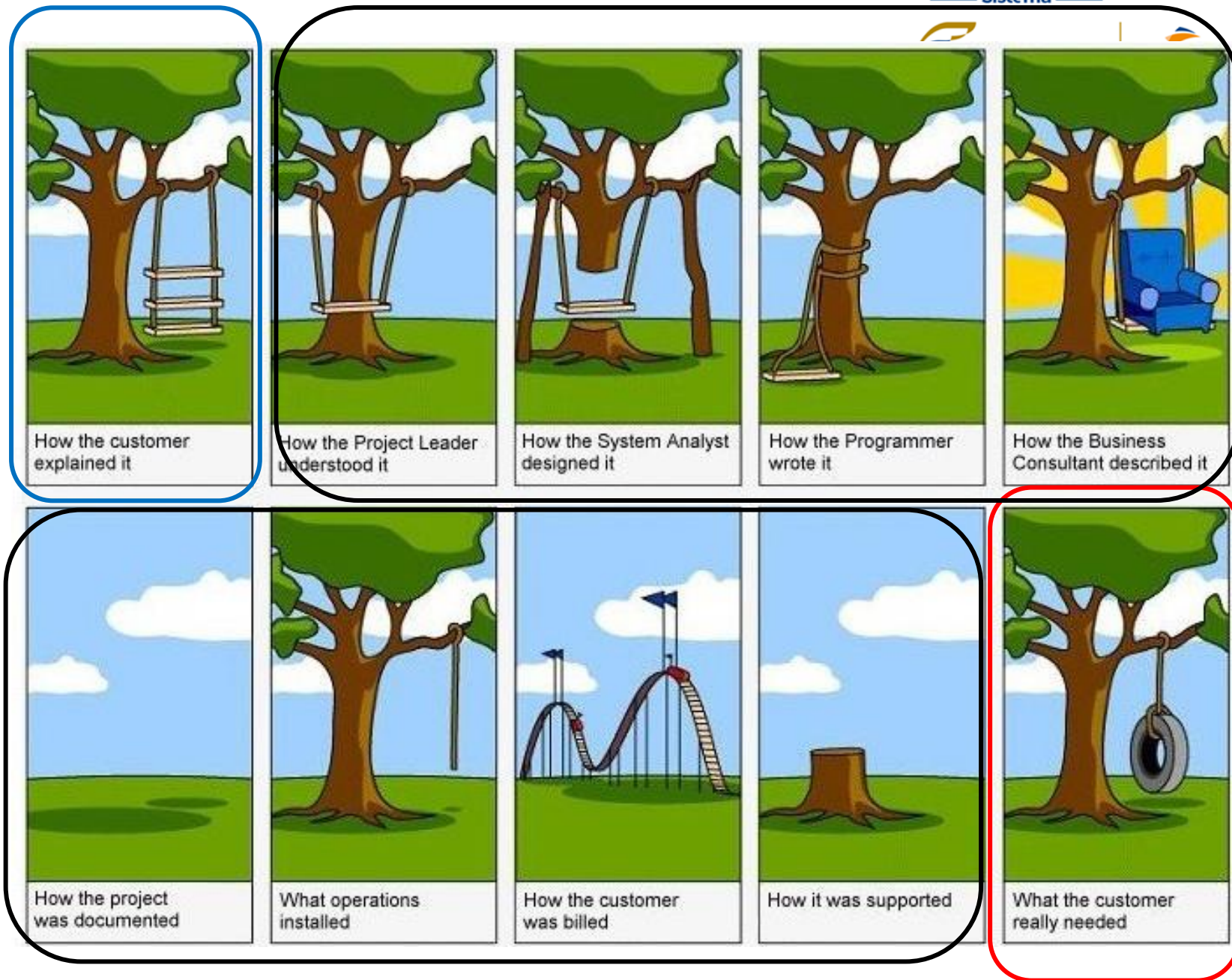
Etapas de Especificação

1. **Estudo de viabilidade.** É feita uma **estimativa** acerca da possibilidade de se **satisfazerem as necessidades** do usuário identificado usando-se tecnologias atuais de software e hardware
2. **Elicitação e análise de requisitos.** Esse é o processo de **derivação dos requisitos** do sistema por meio da observação dos sistemas existentes, além de discussões com os potenciais usuários e compradores, análise de tarefas, entre outras etapas.
3. **Especificação de requisitos.** É a atividade de **traduzir as informações** obtidas durante a atividade de análise em um documento que defina um conjunto de requisitos.
4. **Validação de requisitos.** Essa atividade **verifica** os requisitos quanto a **realismo, consistência e completude**.

Exemplo Estudo de Viabilidade



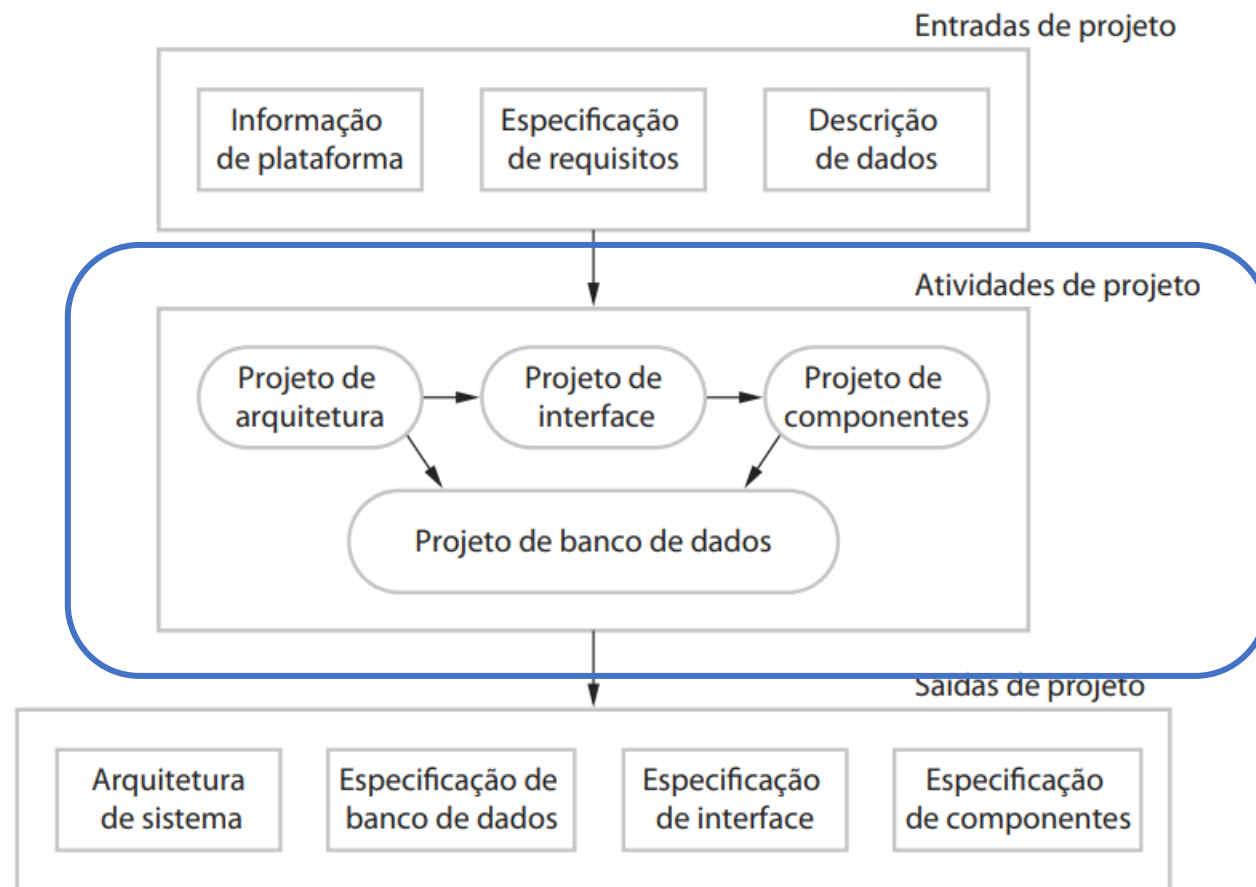
Análise, Especificação e Validação de Requisitos



Projeto e implementação de software

- O estágio de **implementação** do desenvolvimento de software é o **processo de conversão** de uma **especificação** do sistema em um **sistema executável**.

Um modelo geral do processo de projeto



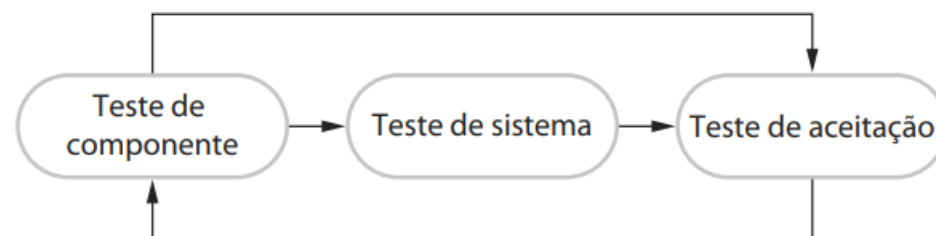
Etapas de Projeto

1. **Projeto de arquitetura**, no qual você pode **identificar** a **estrutura geral** do **sistema**, os **componentes principais** (algumas vezes, chamados subsistemas ou módulos), seus **relacionamentos** e como eles são **distribuídos**.
2. **Projeto de interface**, no qual você **define as interfaces** entre os **componentes** do sistema. Essa especificação de interface deve ser inequívoca.
3. **Projeto de componente**, no qual você toma cada **componente** do **sistema** e **projeta seu funcionamento**.
4. **Projeto de banco de dados**, no qual você **projeta as estruturas de dados** do sistema e como eles devem ser **representados** em um **banco de dados**.

Validação de software

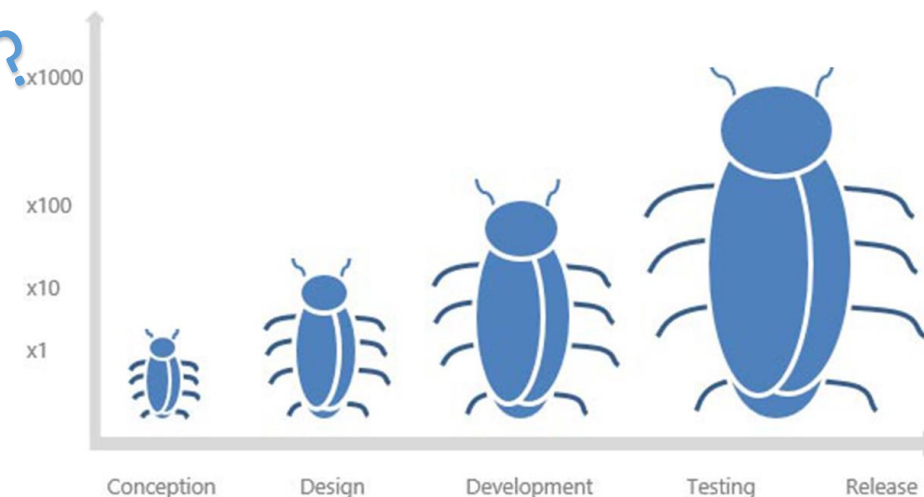


Estágios de testes



- Validação de software ou, mais genericamente, verificação e validação (V&V), tem a intenção de **mostrar que um software se adequa** a suas **especificações** ao mesmo tempo que satisfaz as especificações do cliente do sistema. .

Quanta custa um erro?



Etapas

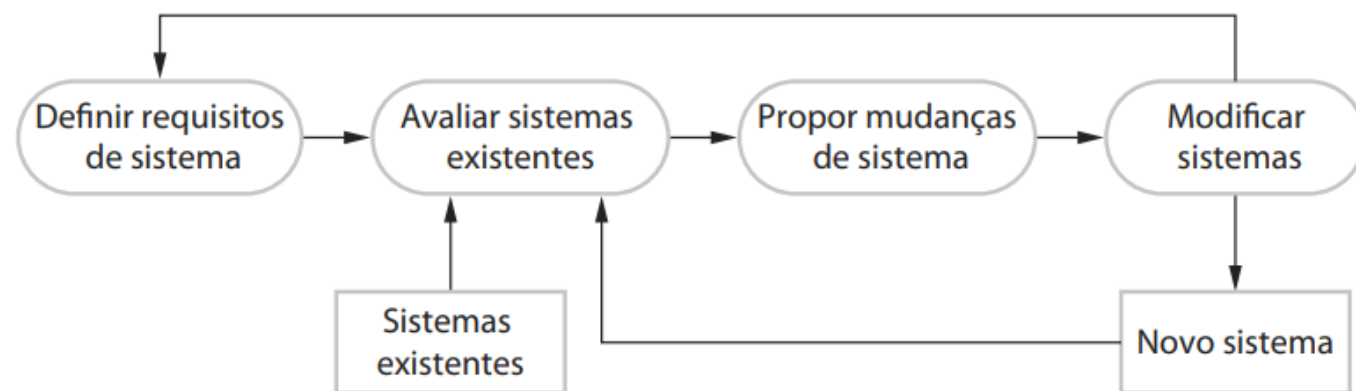


1. **Testes de desenvolvimento.** Os componentes do sistema são testados pelas pessoas que o desenvolveram. Cada **componente é testado** de forma **independente**, separado dos outros.
2. **Testes de sistema.** **Componentes** do **sistema são integrados** para criar um **sistema completo**. Esse processo se preocupa em **encontrar os erros resultantes das interações** inesperadas **entre componentes** e **problemas de interface** do componente.
3. **Testes de aceitação.** Esse é o **estágio final** do processo de testes, antes que o sistema seja aceito para uso operacional. O sistema é **testado com dados fornecidos pelo cliente**, e não com dados advindos de testes simulados.

Evolução do software

- A flexibilidade dos sistemas de software é uma das principais razões pelas quais os softwares vêm sendo, cada vez mais, incorporados em sistemas grandes e complexos.
- Uma vez que a decisão pela fabricação do hardware foi tomada, **é muito caro fazer alterações em seu projeto.**
- Entretanto, as **mudanças no software podem ser feitas a qualquer momento** durante ou após o desenvolvimento do sistema

Evolução do sistema



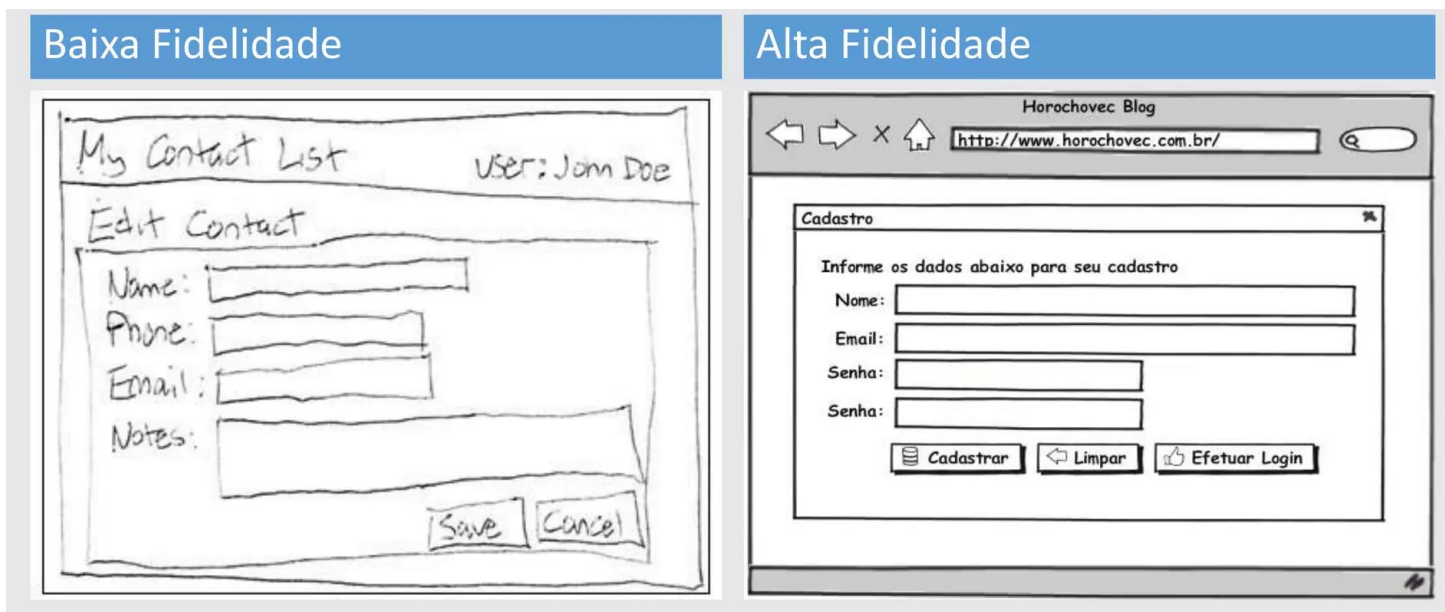
Essa **distinção** entre o **desenvolvimento** e a **manutenção** é cada vez mais **irrelevante**. Poucos sistemas de software são completamente novos, e faz muito mais sentido ver o **desenvolvimento e a manutenção** como **processos contínuos**.

Lidando com Mudanças

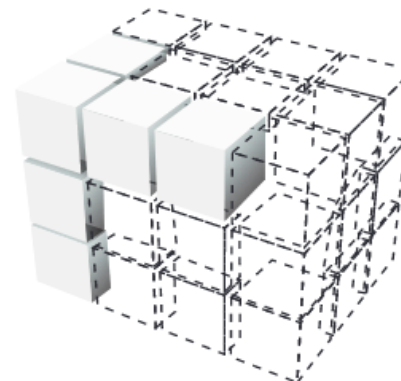
- **Mudança** é **inevitável** em todos os grandes projetos de software.
- Os **requisitos** do sistema **mudam**, ao mesmo tempo que o **negócio** que adquiriu o sistema responde a mudanças de direcionamento estratégico
- Pode, então, ser **necessário re-projetar** o sistema de acordo com os **novos requisitos**

Existem duas abordagens para o retrabalho:

1. **Prevenção de mudanças**, que inclui atividades capazes de **antecipar as mudanças** possíveis antes de qualquer retrabalho necessário (ex: prototipação)



2. **Tolerância a mudanças**, em que o processo foi projetado para que as **mudanças** possam ser **acomodadas** a um **custo relativamente baixo**.



Prototipação

- Um protótipo é **uma versão inicial de um sistema** de software, usado para **demonstrar conceitos**, **experimentar** opções de projeto e **descobrir** mais sobre o problema e suas possíveis soluções.
- O **desenvolvimento rápido** e **iterativo** do **protótipo** é **essencial** para que os custos sejam controlados e os stakeholders do sistema possam experimentá-lo no início do processo de software.



1. No processo de **engenharia de requisitos**, um protótipo pode **ajudar** na **elicitação** e **validação** de **requisitos** de sistema.
2. No processo de **projeto de sistema**, um protótipo pode ser usado para **estudar soluções** específicas do software e para **apoiar** o projeto de interface de usuário

Entrega Incremental

- Abordagem para desenvolvimento de software na qual alguns dos **incrementos desenvolvidos** são **entregues** ao **cliente** e **implantados** para uso em um ambiente operacional.
- Em um processo de entrega incremental os **clientes identificam**, em linhas gerais, os **serviços** a serem **fornecidos** pelo **sistema**, estabelecendo quais dos **serviços** são mais e menos **importantes** para eles.
- Uma série de incrementos de entrega são, então, definidos, com **cada incremento proporcionando** um **subconjunto** da **funcionalidade** do sistema.

