

# Engenharia de Software e da Usabilidade – Aula 2

**Prof. José Maurício S. Pinheiro – 2020-1**

# Mitos do Desenvolvimento de Software

1

**A equipe de desenvolvimento dispõe de um manual com padrões e procedimentos;**

2

**A equipe possui ferramentas de desenvolvimento de software de última geração! Afinal lhes compramos os mais novos computadores;**

3

**Se os prazos atrasam, pode-se adicionar mais pessoas ao grupo e tirar o atraso;**

## Mitos do Desenvolvimento de Software

4

**Uma descrição geral dos objetivos é suficiente para se escrever programas e os detalhes são vistos posteriormente;**

5

**Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o software é flexível.**

6

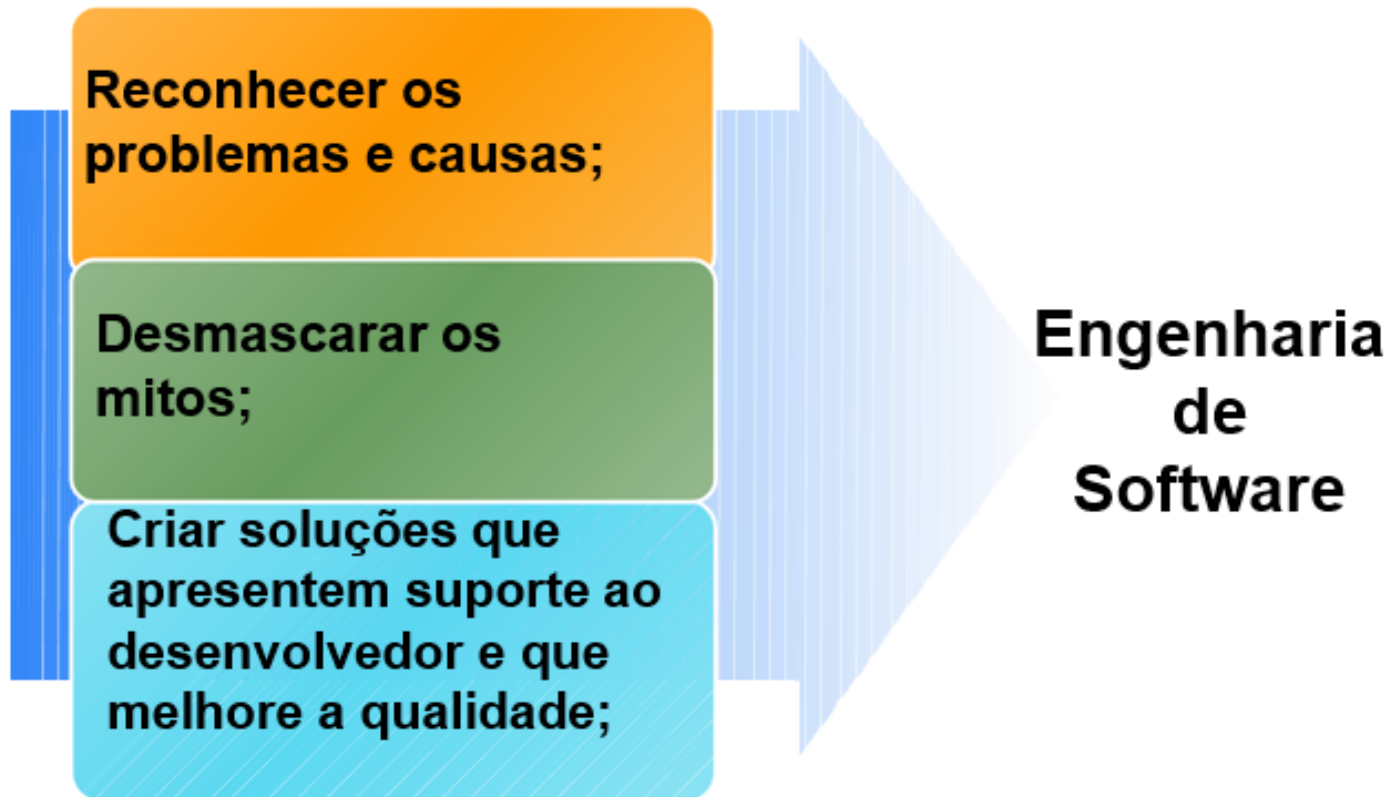
**Assim que escrevermos o programa e o colocarmos em funcionamento nosso trabalho estará completo.**

## Mitos do Desenvolvimento de Software

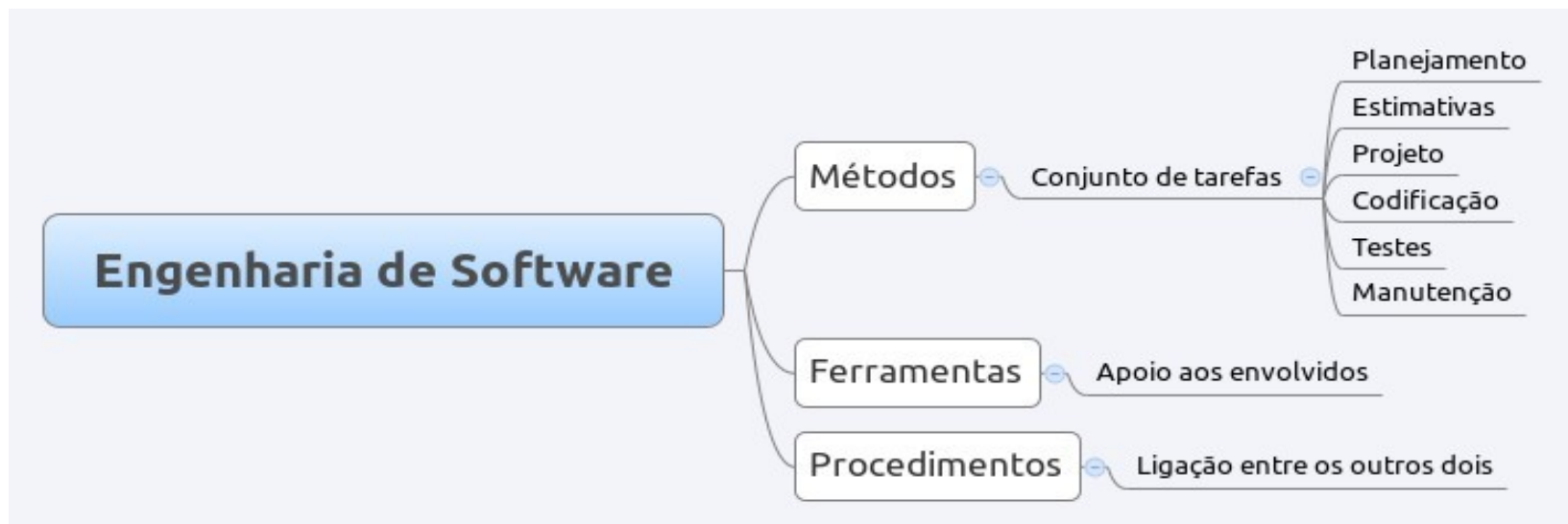
7

**Enquanto não tiver o programa "funcionando", eu não terei realmente nenhuma maneira de avaliar sua qualidade.**

# Engenharia de Software



# Engenharia de Software



# Modelo de Processo de Software

## Definições

*“Um modelo de processo de software é uma representação abstrata de um processo. Os modelos de processo podem ser desenvolvidos a partir de várias perspectivas e podem mostrar as atividades envolvidas em um processo, os artefatos, as restrições que se aplicam a ele e os papéis das pessoas que o aprovam.”*  
(SOMMERVILLE 2011)

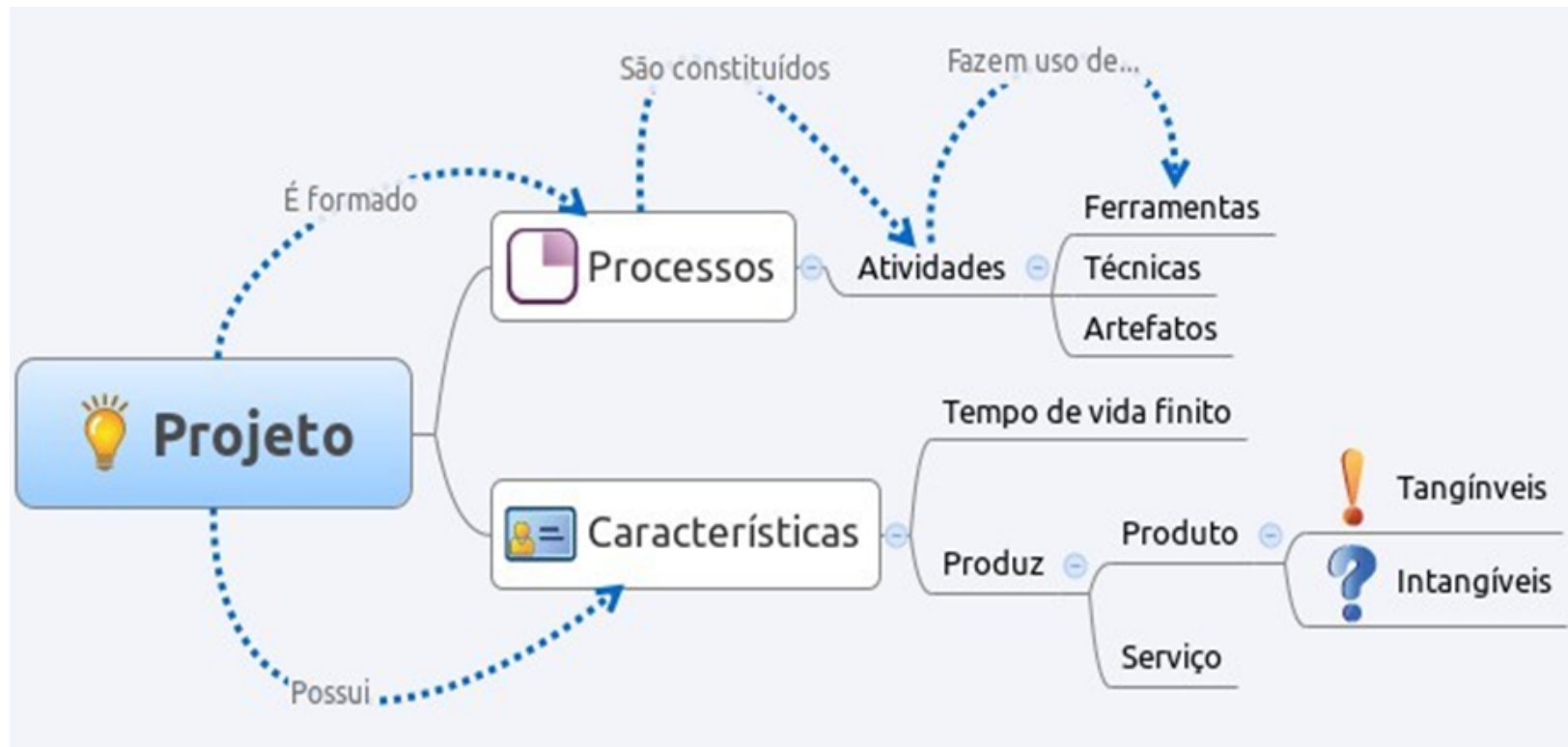
*“Um processo de desenvolvimento de software é um conjunto de atividades, parcialmente ordenadas, com a finalidade de obter um produto de software. É estudado dentro da área de Engenharia de Software, sendo considerado um dos principais mecanismos para se obter software de qualidade e cumprir corretamente os contratos de desenvolvimento.”*

## Modelo de Processo de Software

- ✓ Um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software. Essas atividades podem envolver o desenvolvimento de software a partir do zero em uma linguagem padrão de programação como Java ou C.
- ✓ Atualmente, novos softwares de negócios são desenvolvidos por meio da extensão e modificação de sistemas existentes ou por meio da configuração e integração de prateleira ou componentes do sistema.



# Modelo de Processo de Software



# Modelo de Processo de Software

- ✓ Existem processos de software diferentes, mas todos devem incluir quatro atividades fundamentais para a engenharia de software:

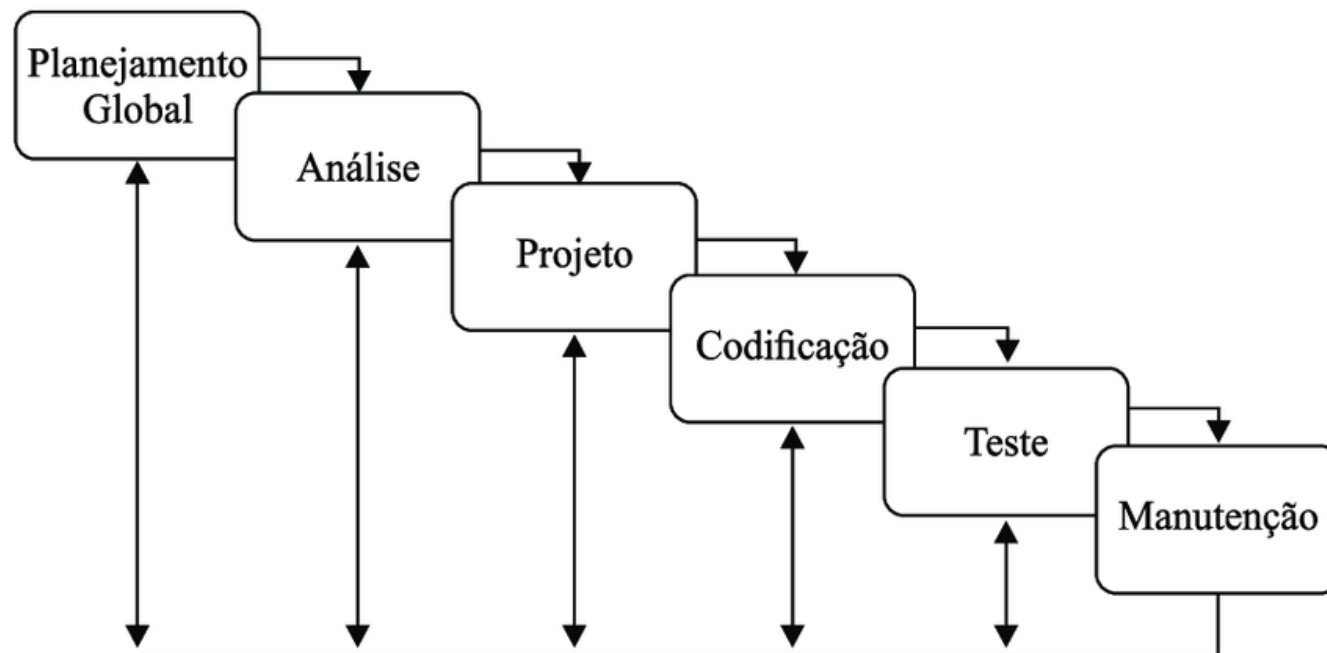


# Modelo de Processo de Software

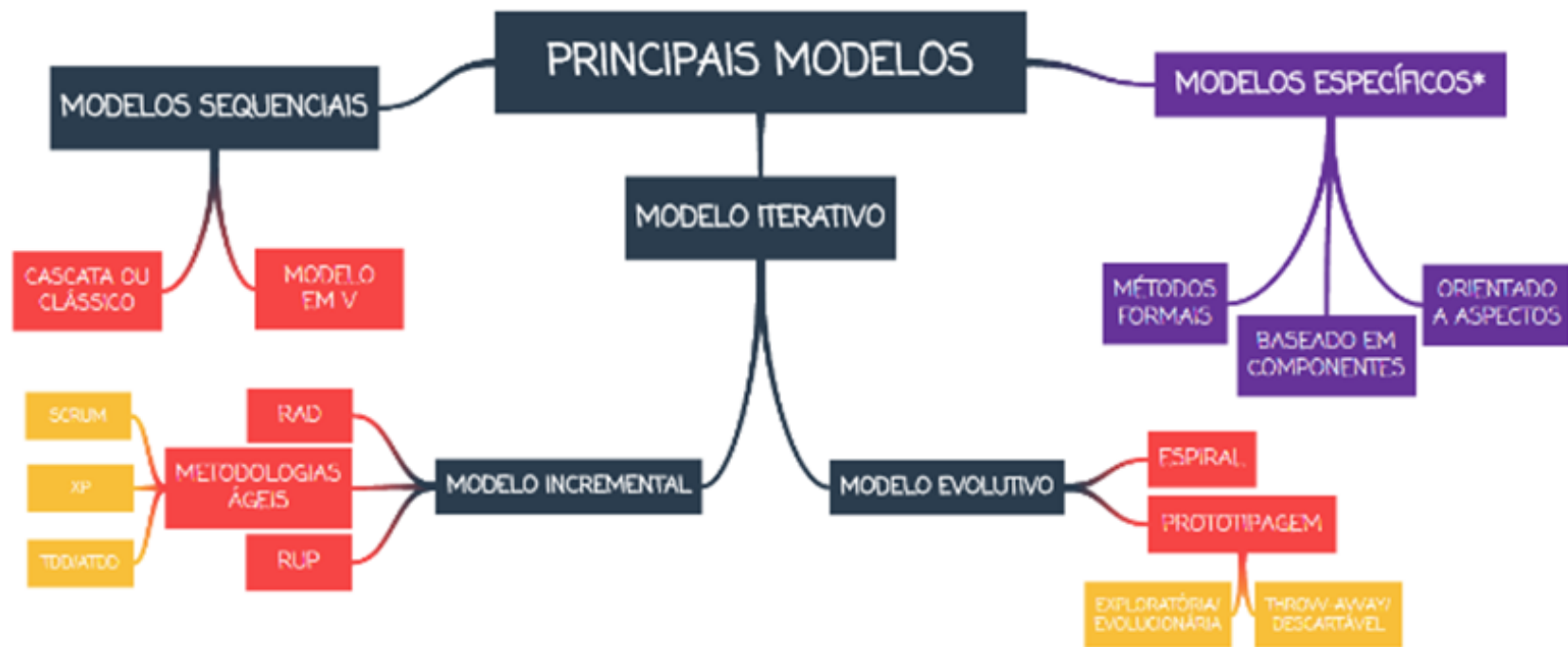
1. **Especificação:** A funcionalidade do software e as restrições a seu funcionamento devem ser definidas.
2. **Desenvolvimento:** O software deve ser produzido para atender às especificações.
3. **Validação:** O software deve ser validado para garantir que atenda às demandas do cliente.
4. **Evolução:** O software deve evoluir para atender às necessidades de mudanças dos clientes.

# Ciclo de Vida do Software

- ✓ O ciclo de vida é a estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema, desde a definição de seus requisitos até o término de seu uso.



# Modelos de Processo de Desenvolvimento de Software



# Modelos de Processo de Desenvolvimento de Software

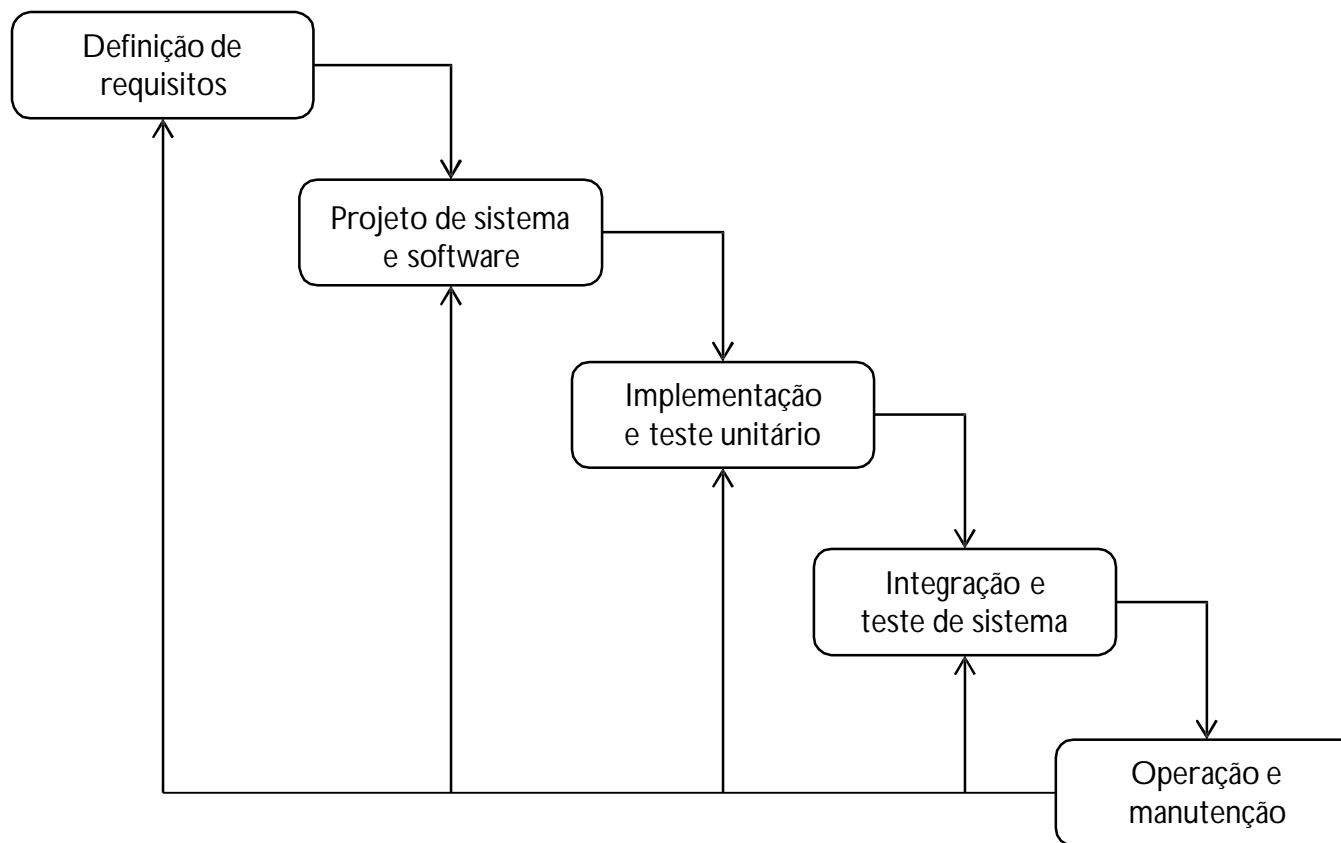
## **Modelos de processo de software**

✓ Os modelos de processo abordados:

- Cascata
- Modelo em V
- Incremental
- Evolutivo
- RAD
- Prototipagem
- Modelo de Ciclo de Vida Associado ao RUP
- Engenharia de Software Orientada a Reuso

# Modelos de Processo de Desenvolvimento de Software

## Modelo em cascata



# Modelos de Processo de Desenvolvimento de Software

## Modelo em Cascata

Os principais estágios do modelo em cascata refletem diretamente as atividades fundamentais do desenvolvimento:

- **Análise e definição de requisitos:** Os serviços, restrições e metas do sistema são estabelecidos por meio de consulta aos usuários. Em seguida, são definidos em detalhes e funcionam como uma especificação do sistema.
- **Projeto de sistema e software:** O processo de projeto de sistemas aloca os requisitos tanto para sistemas de hardware como para sistemas de software, por meio da definição de uma arquitetura geral do sistema. O projeto de software envolve identificação e descrição as abstrações fundamentais do sistema de software e seus relacionamentos.



# Modelos de Processo de Desenvolvimento de Software

## Modelo em Cascata (cont.)

- *Implementação e teste unitário.* Durante esse estágio, o projeto do software é desenvolvido como um conjunto de programas ou unidades de programa. O teste unitário envolve a verificação de que cada unidade atenda a sua especificação.
- *Integração e teste de sistema.* As unidades individuais do programa ou programas são integradas e testadas como um sistema completo para assegurar que os requisitos do software tenham sido atendidos. Após o teste, o sistema de software é entregue ao cliente.

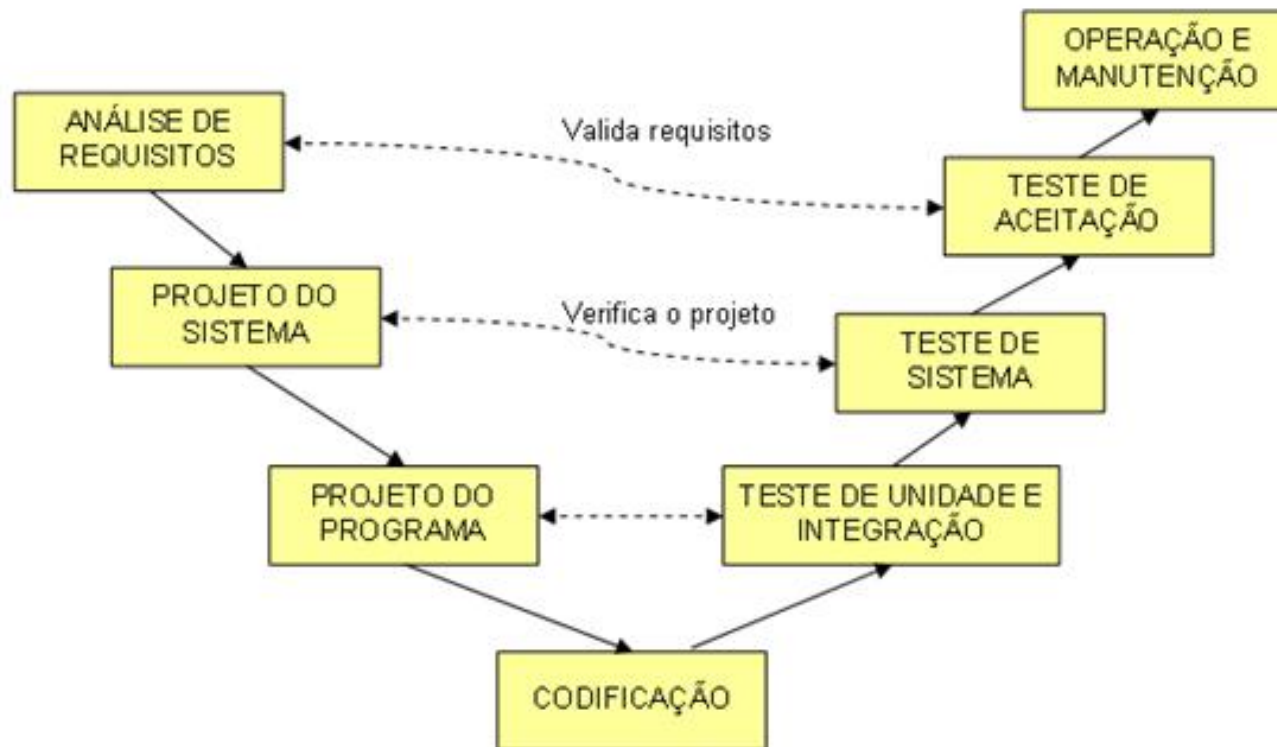
# Modelos de Processo de Desenvolvimento de Software

## Modelo em Cascata (cont.)

- **Operação e manutenção:** Normalmente (embora não necessariamente), essa é a fase mais longa do ciclo de vida. O sistema é instalado e colocado em uso. A manutenção envolve a correção de erros que não foram descobertos em estágios iniciais do ciclo de vida, com melhora da implementação das unidades do sistema e ampliação de seus serviços em respostas às descobertas de novos requisitos.

# Modelos de Processo de Desenvolvimento de Software

## Modelo em V



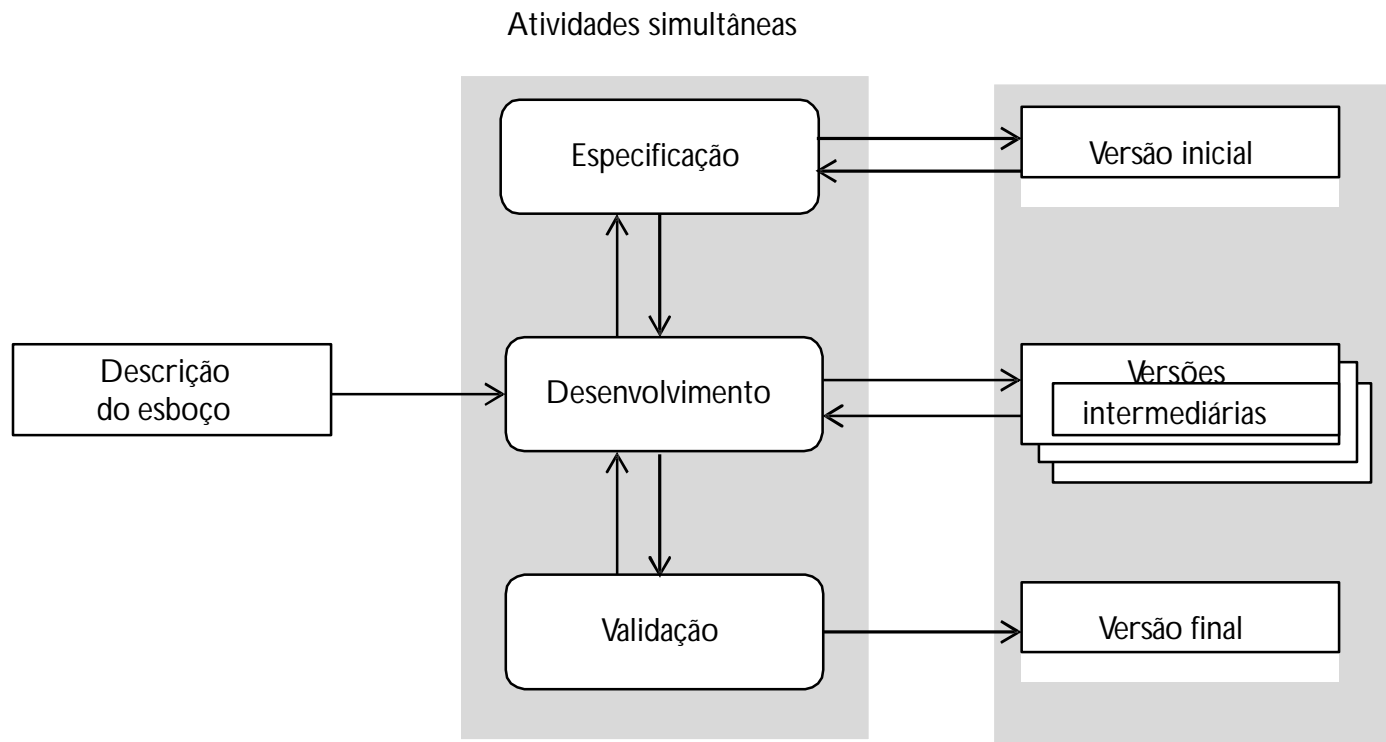
# Modelos de Processo de Desenvolvimento de Software

## Modelo em V

- ✓ Neste modelo, o modelo em cascata é colocado em forma de "V". Do lado esquerdo do V ficam da análise de requisitos até o projeto, a codificação fica no vértice e os testes, desenvolvimento, implantação e manutenção, à direita.
- A característica principal desse modelo, que o diferencia do modelo em cascata, é a ênfase dada à verificação e validação: cada fase do lado esquerdo gera um plano de teste a ser executado no lado direito.
- Da mesma forma que o modelo em cascata, o cliente só recebe a primeira versão do software no final do ciclo, mas apresenta menos risco, devido ao planejamento prévio dos testes nas fases de análise e projeto.

# Modelos de Processo de Desenvolvimento de Software

## Desenvolvimento Incremental



# Modelos de Processo de Desenvolvimento de Software

## Desenvolvimento Incremental

- ✓ O desenvolvimento incremental tem três vantagens importantes quando comparado ao modelo em cascata:
- O custo de acomodar as mudanças nos requisitos do cliente é reduzido. A quantidade de análise e documentação a ser refeita é muito menor do que o necessário no modelo em cascata.
- É mais fácil obter *feedback* dos clientes sobre o desenvolvimento que foi feito. Os clientes podem fazer comentários sobre as demonstrações do software e ver o quanto foi implementado. Os clientes têm dificuldade em avaliar a evolução por meio de documentos de projeto de software.

# Modelos de Processo de Desenvolvimento de Software

## **Desenvolvimento Incremental (cont.)**

- É possível obter entrega e implementação rápida de um software útil ao cliente, mesmo se toda a funcionalidade não for incluída. Os clientes podem usar e obter ganhos a partir do software inicial antes do que é possível com um processo em cascata.

# Modelos de Processo de Desenvolvimento de Software

- ✓ Do ponto de vista do gerenciamento, a abordagem incremental tem dois problemas:
- O processo não é visível. Os gerentes precisam de entregas regulares para mensurar o progresso. Se os sistemas são desenvolvidos com rapidez, não é economicamente viável produzir documentos que reflitam cada uma das versões do sistema.
- A estrutura do sistema tende a se degradar com a adição dos novos incrementos. A menos que tempo e dinheiro sejam dispendidos em refatoração para melhoria do software, as constantes mudanças tendem a corromper sua estrutura. Incorporar futuras mudanças do software torna-se cada vez mais difícil e oneroso.



# Modelos de Processo de Desenvolvimento de Software

## Modelo Evolutivo



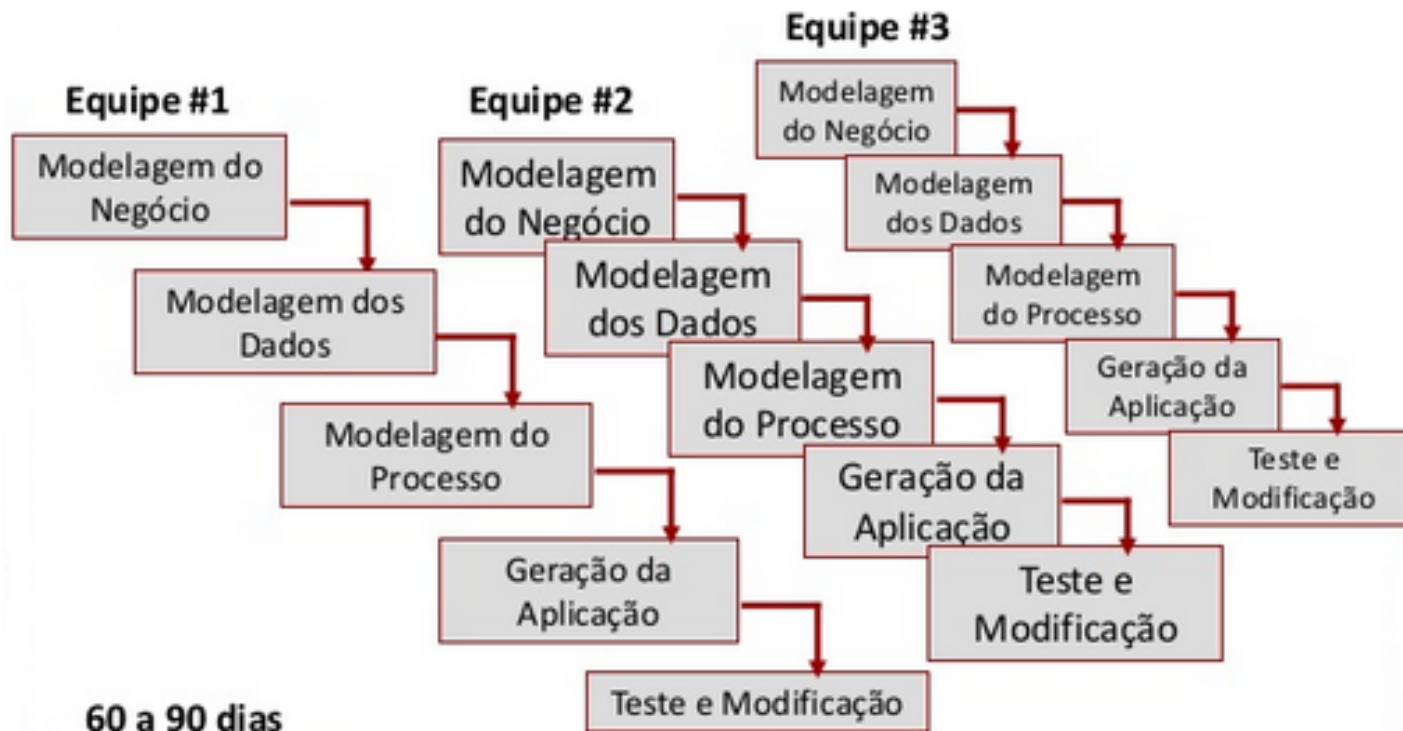
# Modelos de Processo de Desenvolvimento de Software

## **Modelo Evolutivo**

- ✓ Os requisitos são adquiridos em paralelo à evolução do sistema. O modelo evolutivo parte do princípio que o cliente não expõe todos os requisitos, ou os requisitos não são tão bem conhecidos, ou os requisitos ainda estão sofrendo mudanças.
- A análise é feita em cima dos requisitos conseguidos até então, e a primeira versão é entregue ao cliente.
- O cliente usa o software no seu ambiente operacional, e como feedback, esclarece o que não foi bem entendido e dá mais informações sobre o que precisa e sobre o que deseja (ou seja, mais requisitos).
- A partir deste feedback, nova análise, projeto e desenvolvimento são realizados, e uma segunda versão do software é entregue ao cliente que, novamente, retorna com mais feedbacks.

# Modelos de Processo de Desenvolvimento de Software

## RAD – “Rapid Application Development”



# Modelos de Processo de Desenvolvimento de Software

## **RAD – “Rapid Application Development”**

- ✓ Uma evolução da “prototipagem rápida”, destaca-se pelo desenvolvimento rápido da aplicação. O ciclo de vida é extremamente comprimido, de forma a encontrarem-se exemplos, de duração de 60 e 90 dias. É ideal para clientes buscando lançar soluções pioneiras no mercado.
- Ciclo de vida incremental, iterativo, onde é preferível que os requisitos tenham escopo restrito.
- A diferença principal do modelo evolutivo é o forte paralelismo das atividades, requerendo, assim, módulos bastante independentes. Aqui os incrementos são desenvolvidos ao mesmo tempo, por equipes diferentes.

# Modelos de Processo de Desenvolvimento de Software

## Prototipagem



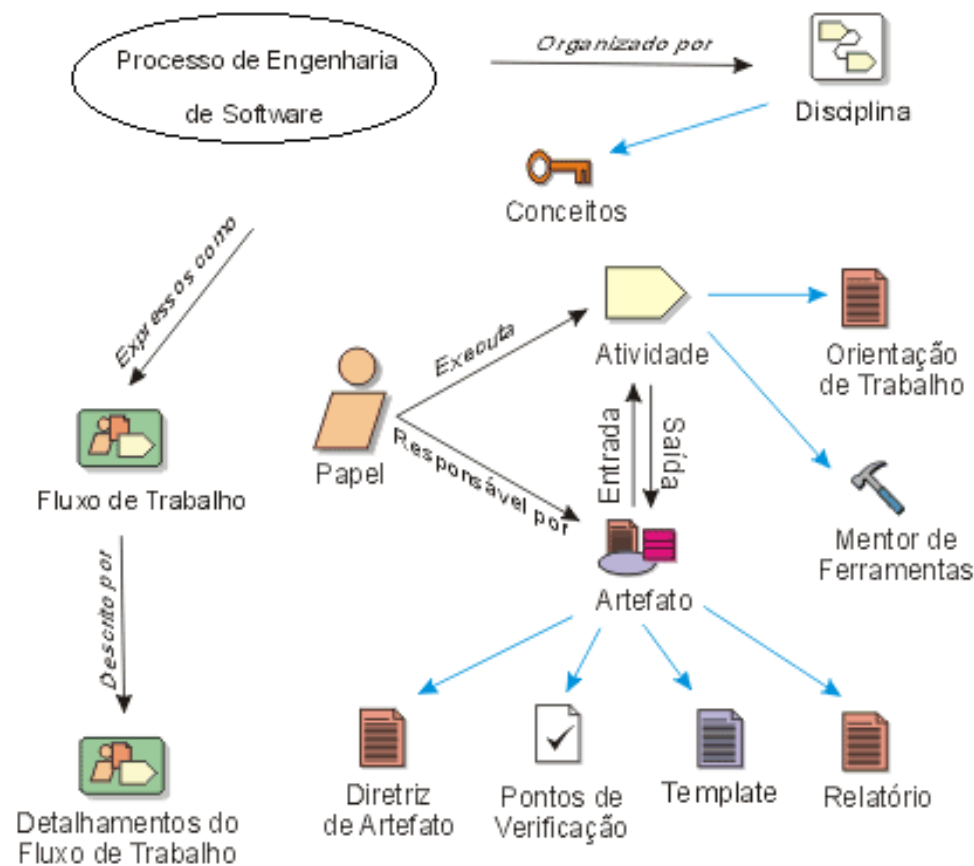
# Modelos de Processo de Desenvolvimento de Software

## Prototipagem

- ✓ É a construção de um exemplar do que foi entendido dos requisitos capturados do cliente. Pode ser considerado um ciclo de vida ou pode ser usado como ferramenta em outros ciclos de vida.
- No ciclo de vida de prototipagem, não é exigido um conhecimento aprofundado dos requisitos num primeiro momento. Isso é bastante útil quando os requisitos não são totalmente conhecidos, são muitos complexos ou confusos.
- A geração de protótipos pode ser facilitada por ferramentas geradoras de telas, de relatórios, poupando esforço de programação e diminuindo o tempo de entrega.
- Cada protótipo tem uma finalidade diferente. Um protótipo pode servir para esclarecer dúvidas sobre uma rotina, demonstrar a aparência das telas, conteúdo de tabelas, formato de relatórios.

# Modelos de Processo de Desenvolvimento de Software

## Modelo de Ciclo de Vida Associado ao RUP



# Modelos de Processo de Desenvolvimento de Software

## Modelo de Ciclo de Vida Associado ao RUP

- ✓ Derivado da UML e do Processo Unificado de Desenvolvimento de Software, o RUP, *Rational Unified Process*, é um modelo de processo iterativo e incremental, dividido em fases, orientado a casos de uso.
- O objetivo do RUP é produzir software com qualidade que satisfaça as necessidades dos clientes dentro de um prazo e orçamento estabelecidos.
- A iteração no RUP tem por objetivo minimizar os riscos. A iteração pode acontecer dentro de cada fase, gerando incrementos, ou em todo o processo.



# Modelos de Processo de Desenvolvimento de Software

## Modelo de Ciclo de Vida Associado ao RUP

- ✓ O RUP possui quatro fases de negócio. O nome de cada fase revela o que será entregue por ela:
- **Concepção:** define o escopo do projeto, ou “business case”; onde é julgado se o projeto deve ir adiante ou ser cancelado.
- **Elaboração:** elabora modelo de requisitos, arquitetura do sistema, plano de desenvolvimento para o software e identificar os riscos.
- **Construção:** constrói o software e a documentação associada.
- **Transição:** finaliza produto, define-se plano de entrega e entrega a versão operacional documentada para o cliente.

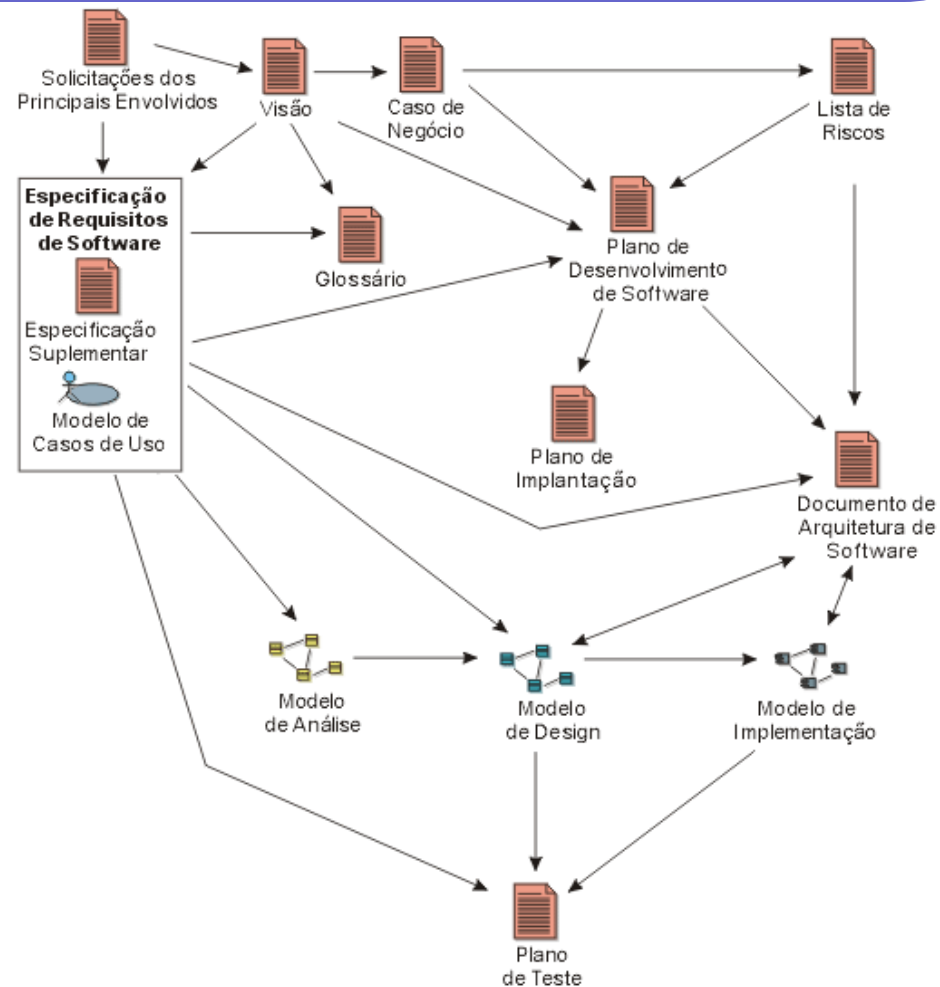
# Modelos de Processo de Desenvolvimento de Software

## **Modelo de Ciclo de Vida Associado ao RUP**

- ✓ Para descrever as atividades (codificação de uma classe, integração de sistemas etc.) o RUP usa guidelines que descrevem técnicas e heurísticas; e de “Mentores de Ferramentas”, que explicam o uso da ferramenta para executar a atividade.
- ✓ Os artefatos de cada fase (documentos, modelos, códigos etc.) são criados, juntamente com templates e exemplos, para melhor entendimento da equipe e do cliente

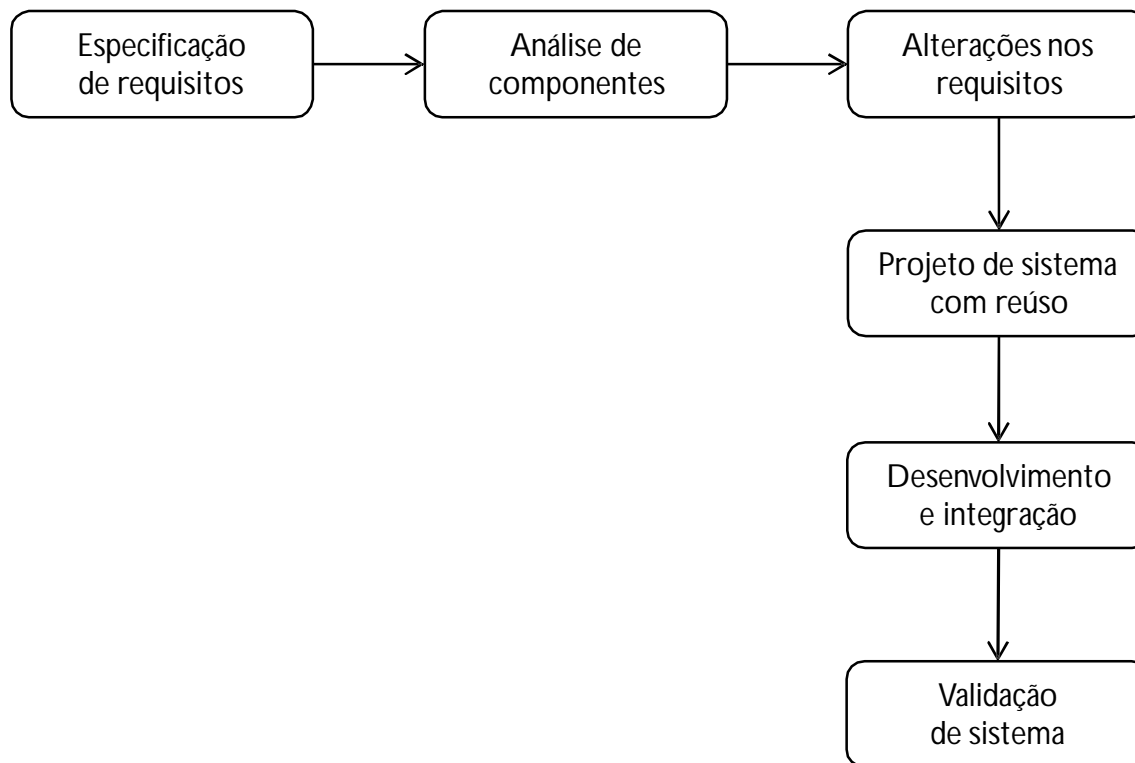
# Modelos de Processo de Desenvolvimento de Software

Os principais artefatos do RUP e o fluxo de informações existente entre eles.



# Modelos de Processo de Desenvolvimento de Software

## Engenharia de software orientada a reuso



# Modelos de Processo de Desenvolvimento de Software

## **Engenharia de software orientada a reúso**

- ✓ Embora o estágio de especificação de requisitos iniciais e o estágio de validação sejam comparáveis a outros processos de software, os estágios intermediários em um processo orientado a reúso são diferentes.
- ✓ Os estágios são:
  - 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.

# Modelos de Processo de Desenvolvimento de Software

## Engenharia de software orientada a reuso (cont.)

- **Modificação de requisitos:** Os requisitos são analisados usando-se informações sobre os componentes que foram descobertos.
- Em seguida, os requisitos são 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.

# Modelos de Processo de Desenvolvimento de Software

## Engenharia de software orientada a reuso (cont.)

- **Projeto do sistema com reuso:** 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 reuso. Alguns softwares novos podem ser necessários, se componentes reusáveis não estiverem disponíveis.
- **Desenvolvimento e integração:** Softwares que não podem ser adquiridos externamente são desenvolvidos, e os componentes e sistemas COTS (Commercial Off-The-Shelf) 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.

# Modelos de Processo de Desenvolvimento de Software

## Engenharia de software orientada a reuso

Existem três tipos de componentes de software que podem ser usados em um processo orientado a reuso:

1. *Web services* desenvolvidos de acordo com os padrões de serviço e que estão disponíveis para invocação remota.
2. Coleções de objetos que são desenvolvidas como um pacote a ser integrado com um framework de componentes, como .NET ou J2EE.
3. Sistemas de software *stand-alone* configurados para uso em um ambiente particular.



# Modelos de Processo de Desenvolvimento de Software

## **Engenharia de software orientada a reúso**

### ✓ **Vantagens:**

- Reduz a quantidade de software a ser desenvolvido, reduzindo assim os custos e riscos.
- Proporciona a entrega mais rápida do software.

### ✓ **Desvantagens**

- Novas versões dos componentes reusáveis não estão sob o controle da organização que os está utilizando.