

Engenharia de Software: Visão Geral

Seiji Isotani, Rafaela V. Rocha

sisotani@icmc.usp.br

rafaela.vilela@gmail.com

PAE: Armando M. Toda

armando.toda@gmail.com

Introdução

“O mundo de hoje não
poderia viver sem o
software”

Sommerville, 2011

Introdução

O que é **software**?



Introdução

- Conceito de **software**
- Artigo: ***The Five Orders of Ignorance***

Phillip G. Armour. 2000. The five orders of ignorance. *Commun. ACM* 43, 10, October 2000, 17-20.



The Five Orders of Ignorance

- Conceito de **software**
- *Software é um meio de armazenamento de **conhecimento***
- *Mantém o conhecimento **ativo!***

The Five Orders of Ignorance

- ***Conhecimento ativo:***
 - ***DNA***
 - ***Cérebro***
 - ***Livros***
 - ***....***

The Five Orders of Ignorance

- Software = conhecimento
 - Produzir software -> **processo de adquirir conhecimento**
 - Fácil produzir software
 - Difícil produzir software **correto!**

The Five Orders of Ignorance

- Produzir software -> atividade de adquirir conhecimento
- Como adquirir conhecimento?
- Qual é o nível de conhecimento necessário?
 - Necessário reconhecer o nível de ignorância...

The Five Orders of Ignorance

- **Nível 0 – ausência de ignorância**
 - Conheço e posso provar
 - Nível desejado de ignorância
 - Impossível estar no **Nível 0** para todo tipo de conhecimento existente
 - Desenv. software: conheço e incorporo ao software, sei a resposta.

The Five Orders of Ignorance

- **Nível 1 – falta de conhecimento**
 - Sei o que não sei!
 - Ex: Não sei pilotar avião...
 - Saber identificar que está no Nível 1 aumenta a chance de ir para o Nível 0
 - Desenv. software: conheço as variáveis envolvidas mas não o seu valor, tenho a pergunta

The Five Orders of Ignorance

- **Nível 2 – falta de ciência (*awareness*)**
 - Não sei que desconheço!
 - Não trata-se somente de ser ignorante sobre algo, eu sou totalmente alheio do fato, i.e, não sei o suficiente para saber que não sei o suficiente

The Five Orders of Ignorance

- **Nível 2 (cont) – falta de ciência (*awareness*)**
 - Desenv. software: problema real do desenvolvimento de software
 - Quando se inicia um projeto sabemos que não sabemos muito sobre o projeto
 - Como estimar algo que não sabemos o que não sabemos a respeito?

The Five Orders of Ignorance

- **Nível 3 – perda de processo**
 - Não sei uma maneira de encontrar o que não sei que eu não sei
 - O ideal seria mudar coisas que eu não sei que desconheço para:
 - Coisas que eu conheço
 - Ou pelo menos
 - Coisas que eu sei que desconheço
 - Mas nesse nível não sei como fazer isso (process)

The Five Orders of Ignorance

- **Nível 3 (cont) – perda de processo**
 - Desenv. software = existem metodologias para o desenvolvimento de software, mas elas por si só não respondem essa questão
 - Não é o seu papel!
 - Não tenho como resolver minha falta de conhecimento no tempo disponível

The Five Orders of Ignorance

- **Nível 4 – meta ignorância**
 - Não conheço os cinco níveis de ignorância!!!
 - Não estamos mais no nível 4... 😊

Introdução

- Lei das “**consequências não pretendidas**” (Pressman, 2006)
 - Ex: isopor, fusão, microondas, facebook...
 - Na década de 50 não era esperado que o software tomasse as proporções atuais
 - Quais as consequências disso?

Introdução

- Software presente no dia a dia
 - Tudo o que nos rodeia, desde produtos eletrônicos a serviços de transportes, médicos, telecomunicações, militar, industrial e financeiro, entretenimento e muitos outros, possui embutido algum tipo de software.
 - Esse panorama dificilmente poderia ter sido previsto nos anos 50

Pressman, 2010

Introdução

- Software presente no dia a dia
 - Quais as consequências disso?
 - Agilidade
 - Portabilidade
 - Compartilhamento de informações
 - Tolerância a falhas
 - Corretude
 - Software resolvendo diferentes problemas...

Tipos de software

Software básico

Software de tempo real

Software comercial (gerencial, operacional)

Software científico ou de engenharia

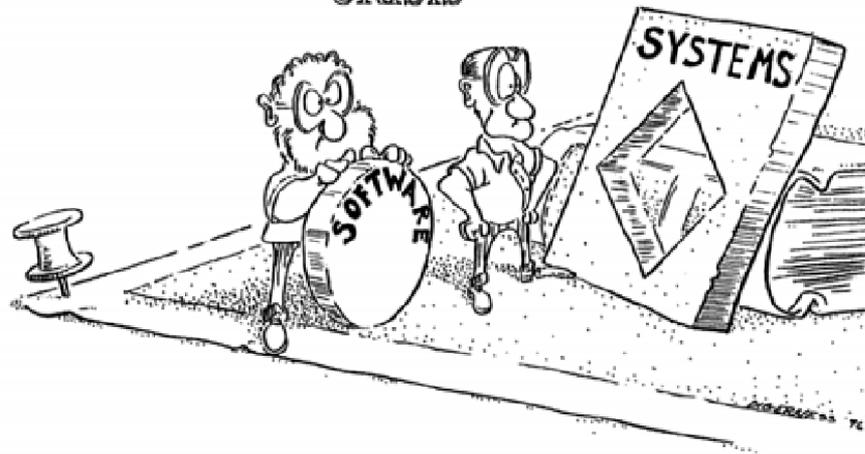
Software embutido ou embarcado

Software para inteligência artificial

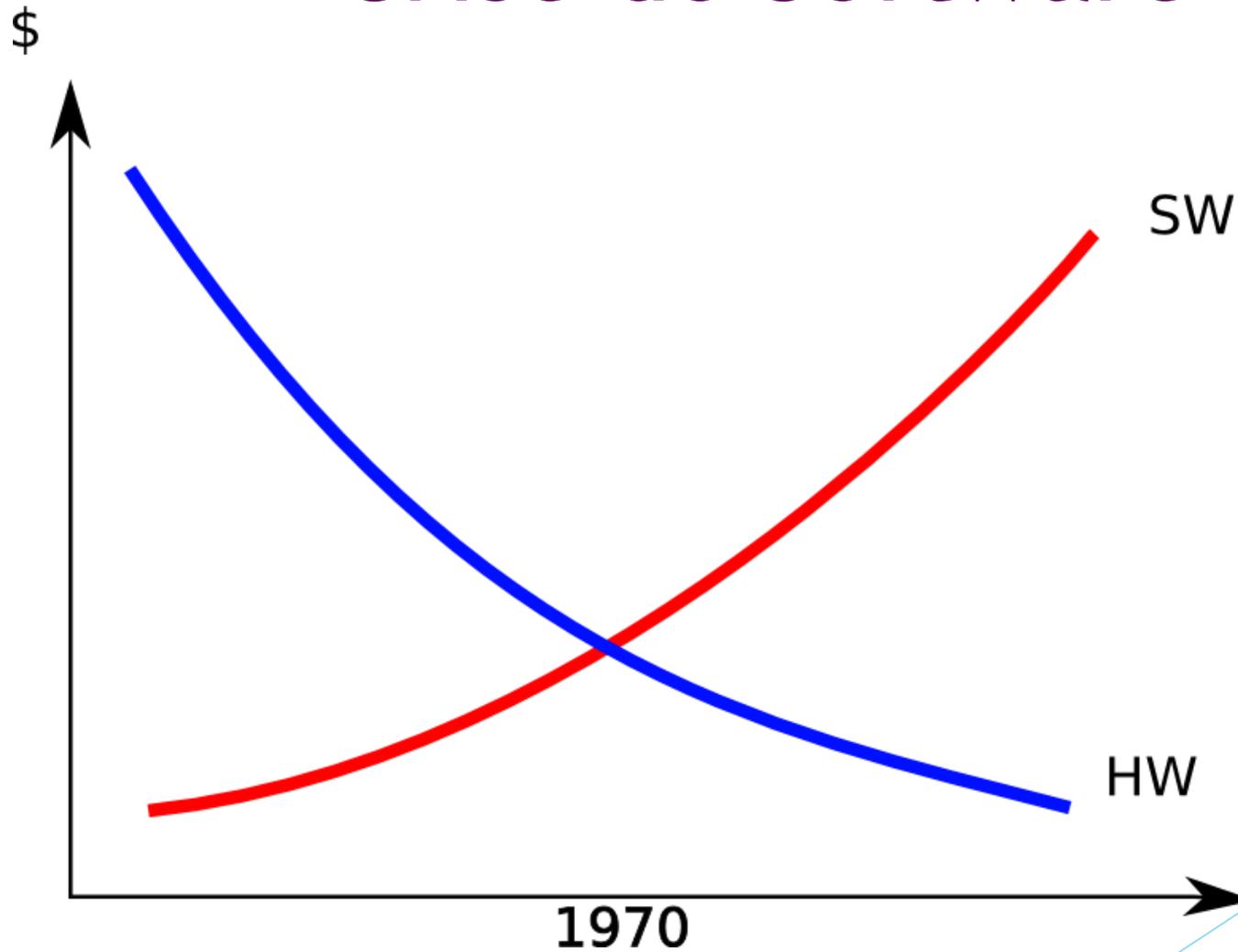
Jogos

Crise do Software

THE SOFTWARE
CRISIS



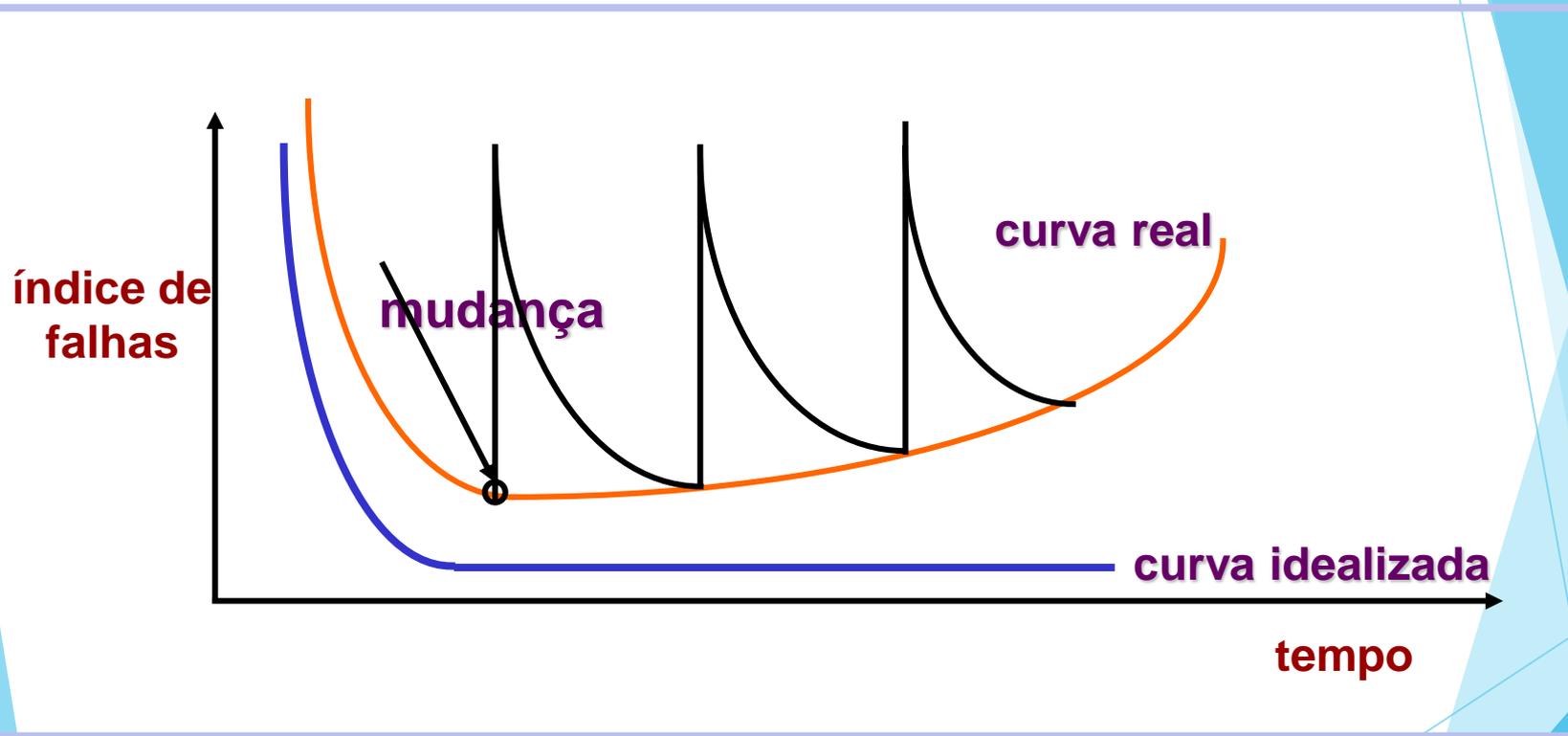
Crise do Software



Curva de falhas para o Hardware



Curva de falhas do Software



Evolução do Software

Crise do software

A quarta era

Os primeiros anos

- ORIENTAÇÃO BATCH
- DISTRIBUIÇÃO LIMITADA
- SOFTWARE CUSTOMIZADO

A segunda era

- MULTIUSUÁRIO
- TEMPO REAL
- BANCO DE DADOS
- PRODUTOS DE SOFTWARE

A terceira era

- SISTEMAS DISTRIBUÍDOS
- INTELIGÊNCIA EMBUTIDA
- HARDWARE DE BAIXO CUSTO
- IMPACTO DE CONSUMO

- SISTEMAS DE DESKTOP PODEROSOS
- TECNOLOGIAS ORIENTADAS A OBJETOS
- SISTEMAS ESPECIALISTAS
- REDES NEURAIS ARTIFICIAIS
- COMPUTAÇÃO PARALELA

1950

1960

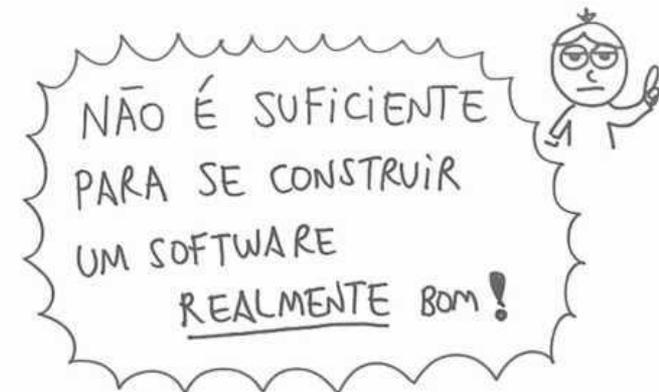
1970

1980

2000

Causas da crise de software

- ▶ Projetos que estouram o cronograma
- ▶ Projetos que estouram o orçamento
- ▶ Baixa qualidade
- ▶ Clientes insatisfeitos
- ▶ Produtos difíceis de manter....
- ▶ e
- ▶ Mitos...



Principais mitos

- ▶ Atraso no projeto X alocação de pessoas
- ▶ Mudanças se acomodam facilmente depois do software pronto
- ▶ Detalhes do problema podem ser passados durante o projeto



Por que surgiu a Engenharia de Software?



Definição de Engenharia de Software



Engenharia de Software

Conjunto de **processos, métodos, técnicas** e **ferramentas** que ajudam a produzir software com maior qualidade e com menor custo.

Incluir **processo de desenvolvimento** e atividades de **gerenciamento de projetos**

O Processo de Software

- Elementos fundamentais:
 - Métodos
 - Ferramentas
 - Procedimentos
- Para projetar, construir e manter grandes sistemas de software de forma profissional.

O Processo de Software

- ▶ **MÉTODOS:** prescrevem os detalhes sobre como fazer para construir o software.

Engenharia de requisitos

Análise de requisitos

Projeto de software

Codificação

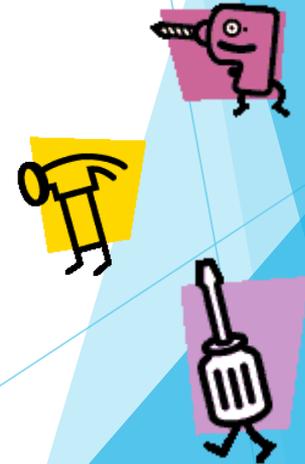
Teste

Manutenção

...

O Processo de Software

- ▶ **FERRAMENTAS:** dão suporte automatizado aos métodos.
 - ▶ Existem atualmente ferramentas para **apoiar** cada um dos métodos.
 - ▶ **CASE** (Computer Aided Software Engineering)
 - ▶ Quando as ferramentas são integradas é estabelecido um sistema de suporte ao desenvolvimento de software.



O Processo de Software

- ▶ **PROCEDIMENTOS:** constituem o elo de ligação entre os métodos e ferramentas.
 - ▶ **Sequência** em que os métodos são aplicados.
 - ▶ **Produtos** que se exige que sejam entregues.
 - ▶ **Controles** que ajudam assegurar a qualidade e coordenar as alterações.
 - ▶ **Marcos de referência** que possibilitam administrar o progresso do software.

Evolução do Software

2000 – atual

Sistemas Web

Web services

Software como serviço (cloud computing)

Aplicativos para smartphone (mobile applications)

Internet das coisas

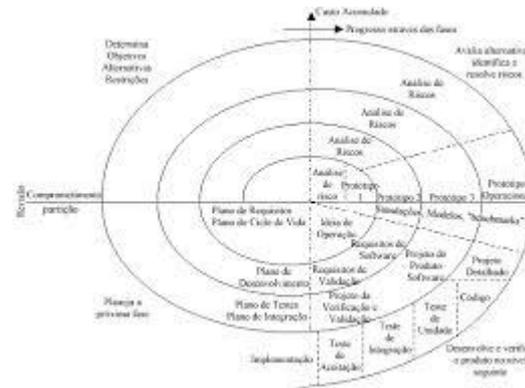
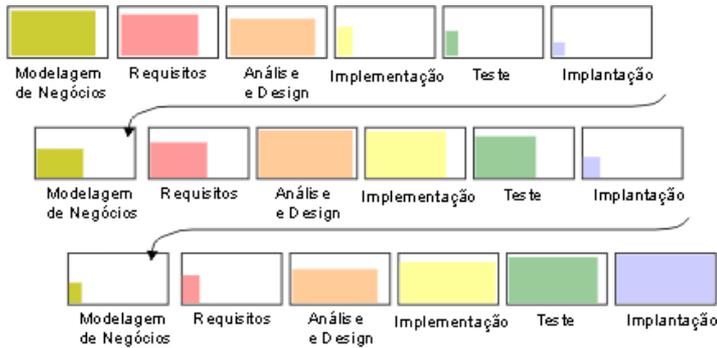
Sistema-de-sistemas



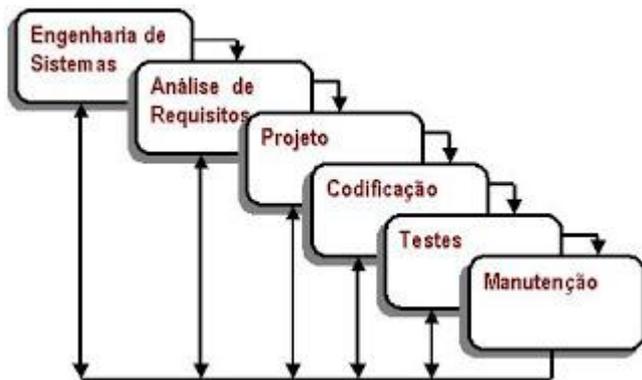
Desafios Atuais da Engenharia de Software

- ◆ **Sistemas legados**
 - ◆ **Heterogeneidade**
 - ◆ **Entrega dos sistemas**





Processo de desenvolvimento de Software



Processo de Software

Atividades fundamentais:

1. Especificação do software
2. Projeto e implementação do software
3. Validação do software
4. Evolução do software

Processo genérico de Engenharia de Software

Uma Visão Genérica: 3 Fases

1. Definição - "*o que*"

- Engenharia do Sistema
- Planejamento do Projeto
- Engenharia de Requisitos

2. Desenvolvimento - "*como*"

- Projeto
- Geração do Código
- Teste

3. Manutenção



Atividades Guarda-Chuva

- Controle de Projeto
- Revisões Técnicas Formais
- Garantia de Qualidade
- Gerenciamento de Configuração
- Gestão de Reutilização
- Medição
- Gestão de Risco

Processo de Software

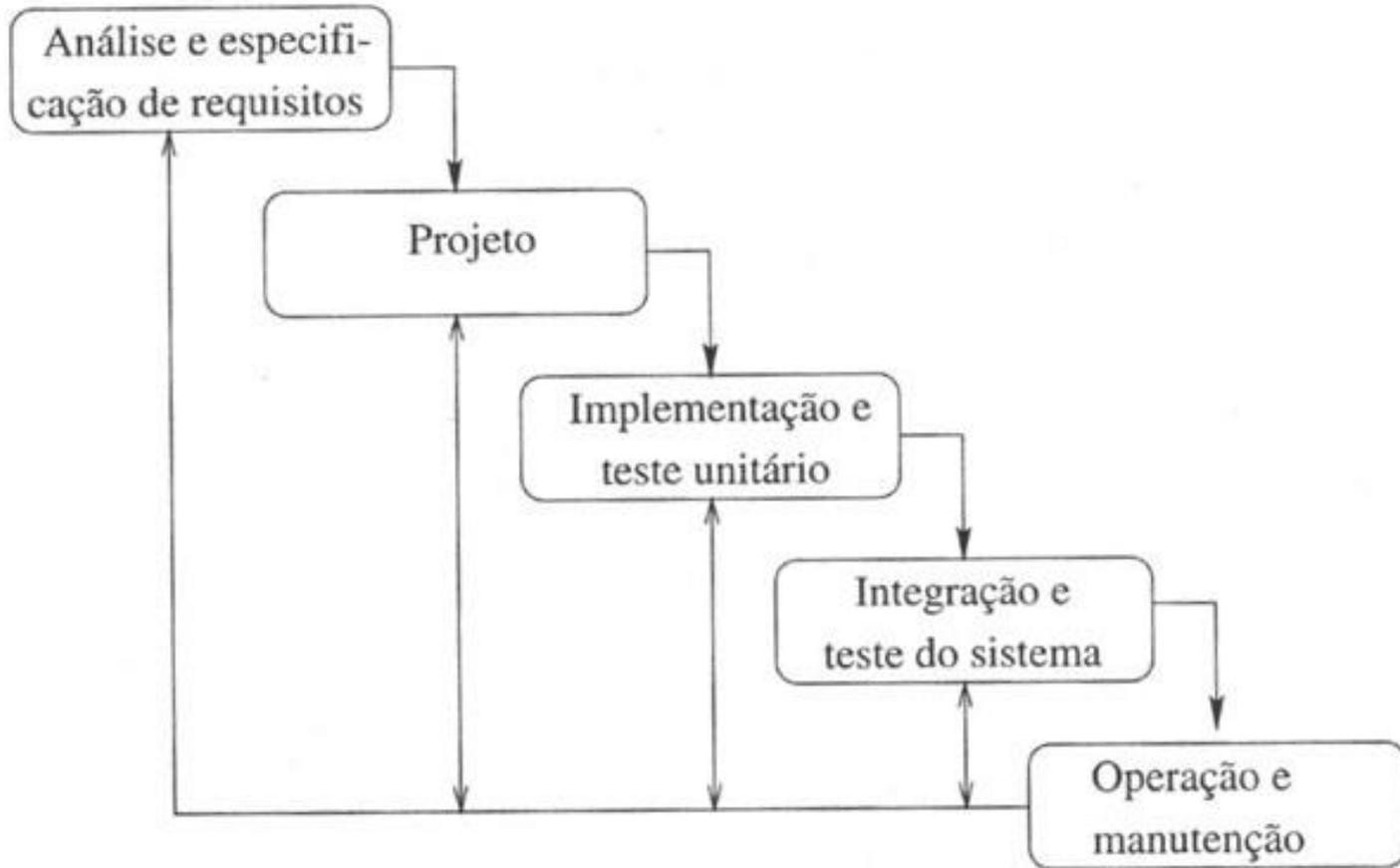
- ◆ Diferentes processos de desenvolvimento
- ◆ Escolhido com base:
 - na natureza do projeto e da aplicação
 - nos métodos e ferramentas a serem utilizados
 - nos controles e produtos que precisam ser entregues

Processos de Software

1. Modelo Cascata
2. Prototipação
3. Modelo Incremental
4. Modelo Espiral
5. Modelo RAD
6. Modelo RUP
7. Modelo Ágil
8. Vários outros



Modelo Cascata



Tarefa - próxima aula

- ▶ Ler Capítulo 1
- ▶ Exercícios sobre o capítulo na próxima aula (15 min)