



# Processo de Software

Aula 1 de 32

Prof. Dsc. Nauber Gois

## Introdução a Processo de Software



# Apresentação



**Francisco Nauber Bernardo Gois**  
**Doutor em Informática Aplicada**  
**Professor da Universidade Federal do Ceará**

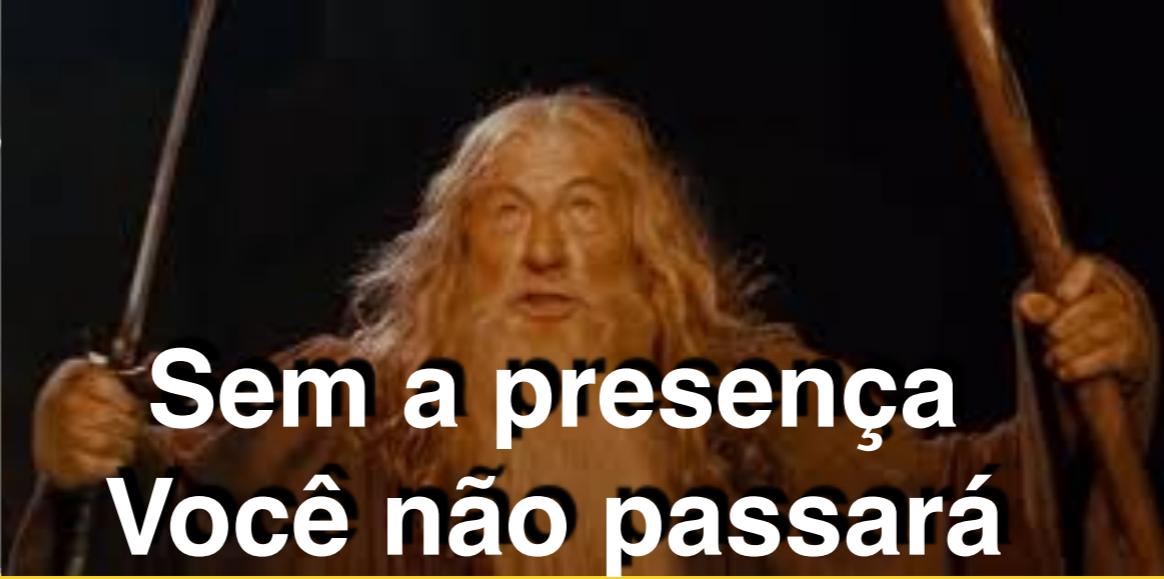
**Curso de Manutenção de Software -Aula 1 de 32**  
**Dúvidas: [naubergois@ufc.br](mailto:naubergois@ufc.br)**

**Jovem Padawan  
procure na aula**

**ao telefone  
não falar**



**Procure  
não  
conversar  
durante a  
aula**



**Para melhor  
desempenho na  
aula**

**Buscar  
aprendizado  
ao invés de  
pontos**

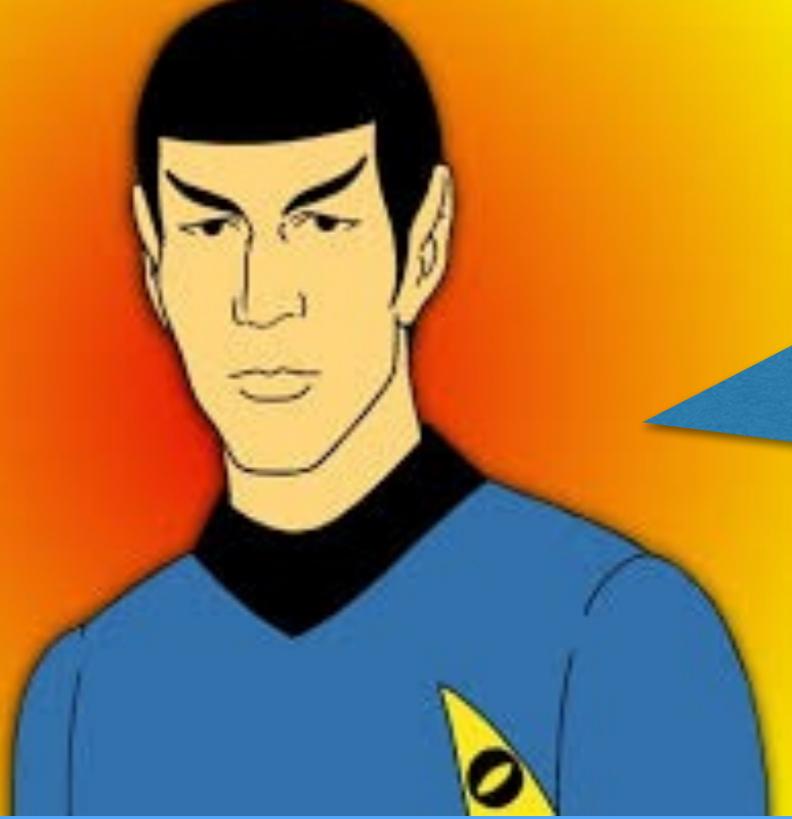


**Não  
teremos  
pontuação fora dos  
trabalhos e provas  
da disciplina**



**Cuidado  
com o  
Horário**



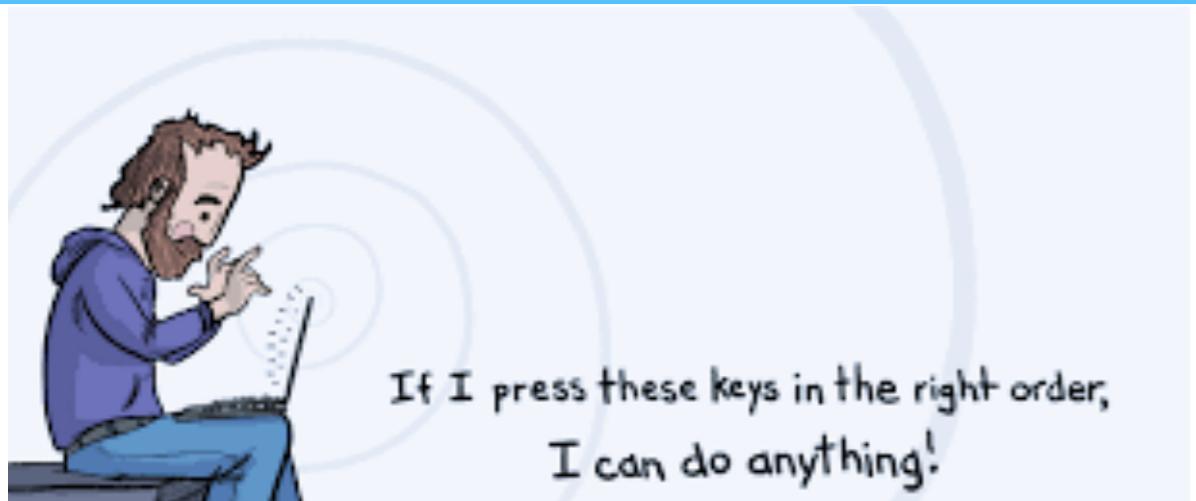


**Não teremos  
pontos após a prova  
não adianta pedir**

**Um cadeira longa  
e prospera**



# Metodologia de Avaliação



**Prova Prática - Desenvolver Software escolhido pelo Aluno aplicando os conceitos de manutenibilidade e reuso apresentados na cadeira (10 pontos).**



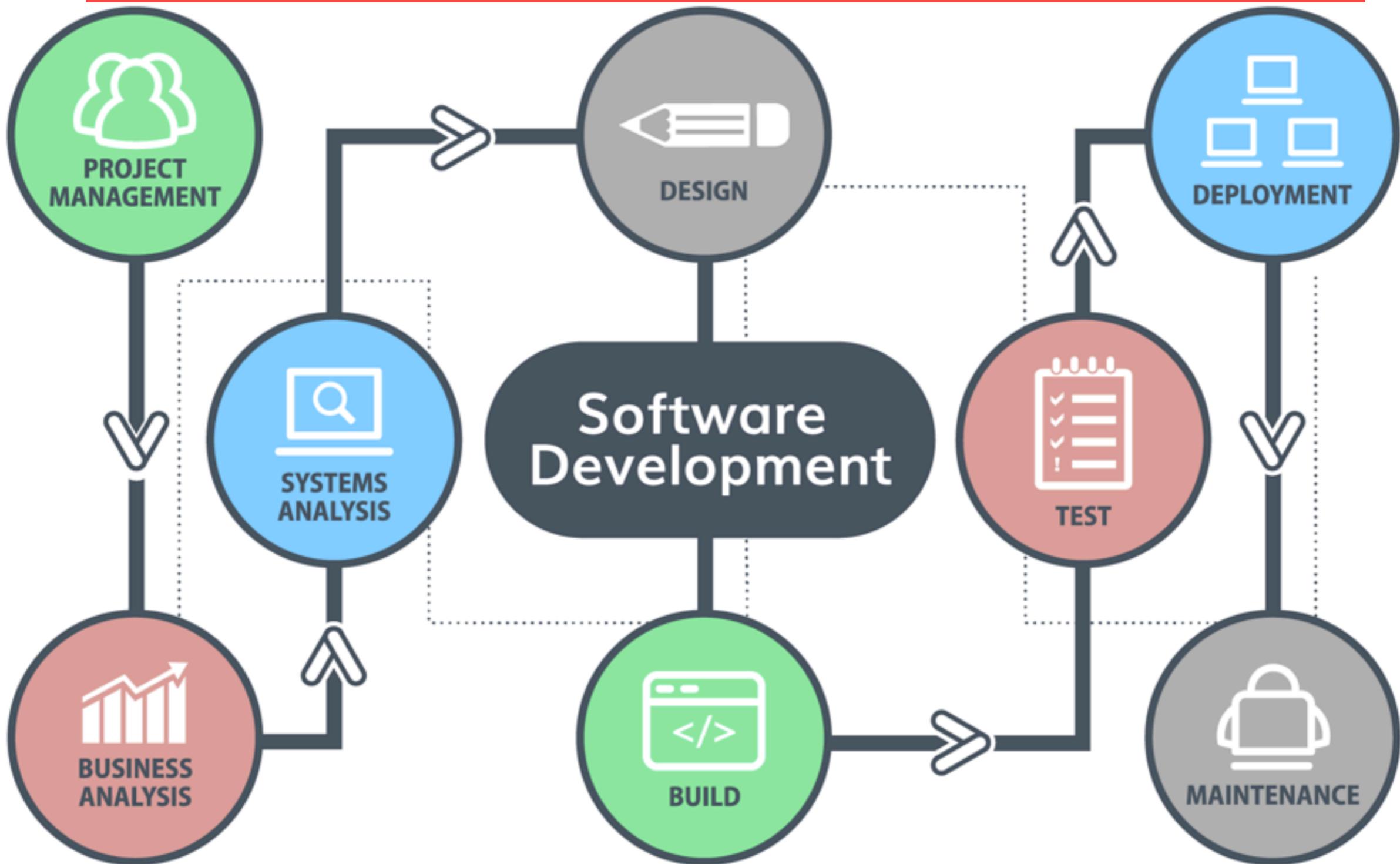
**Prova Teórica - Escrever artigo de 3 páginas em assunto acordado previamente com o professor (10 pontos).**

**Curso de Manutenção de Software -Aula 1 de 32  
Dúvidas: [naubergois@ufc.br](mailto:naubergois@ufc.br)**

# **Após essa aula o aluno deve :**

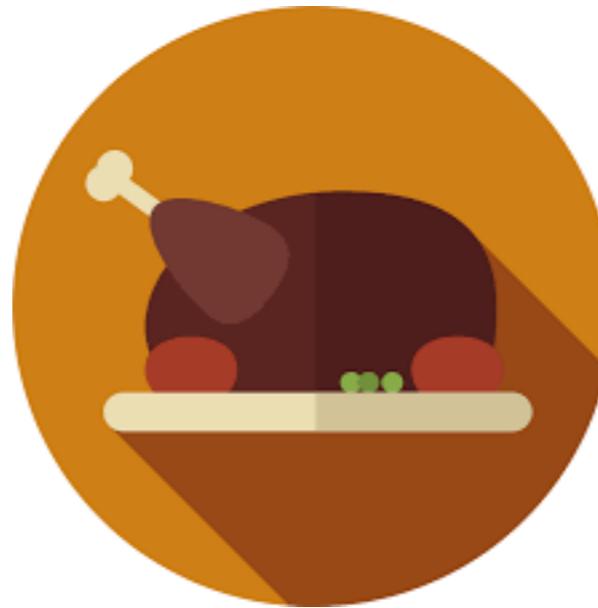
- **Aprender o conceito de Manutenibilidade**
- **Distinguir Coesão e Acoplamento**
- **Entender o conceito de Modularização**
- **Entender o que são Interfaces**
- **Entender o conceito de Inversão de Controle e Injeção de Dependência**

# Introdução ao Processo de Software



# Introdução a Manutenção de Software

## Processo



Curso de Processo de Software -Aula 1 de 32

Dúvidas: [naubergois@ufc.br](mailto:naubergois@ufc.br)

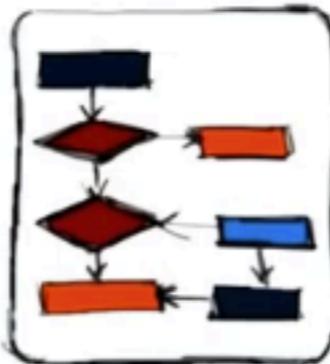
# Processo de Software

- Um conjunto estruturado de atividades necessárias para o desenvolvimento de um sistema de software;
- Atividades variam com a organização e o tipo de sistema sendo desenvolvido;

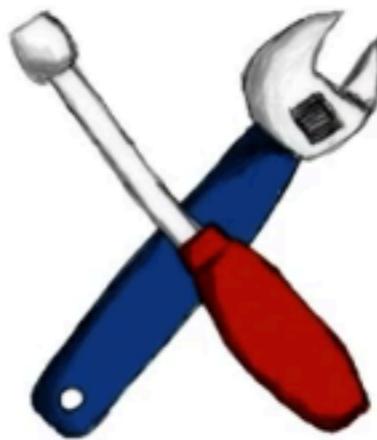
# DISCIPLINE OF SOFTWARE ENGINEERING



Methodologies



Techniques



Tools



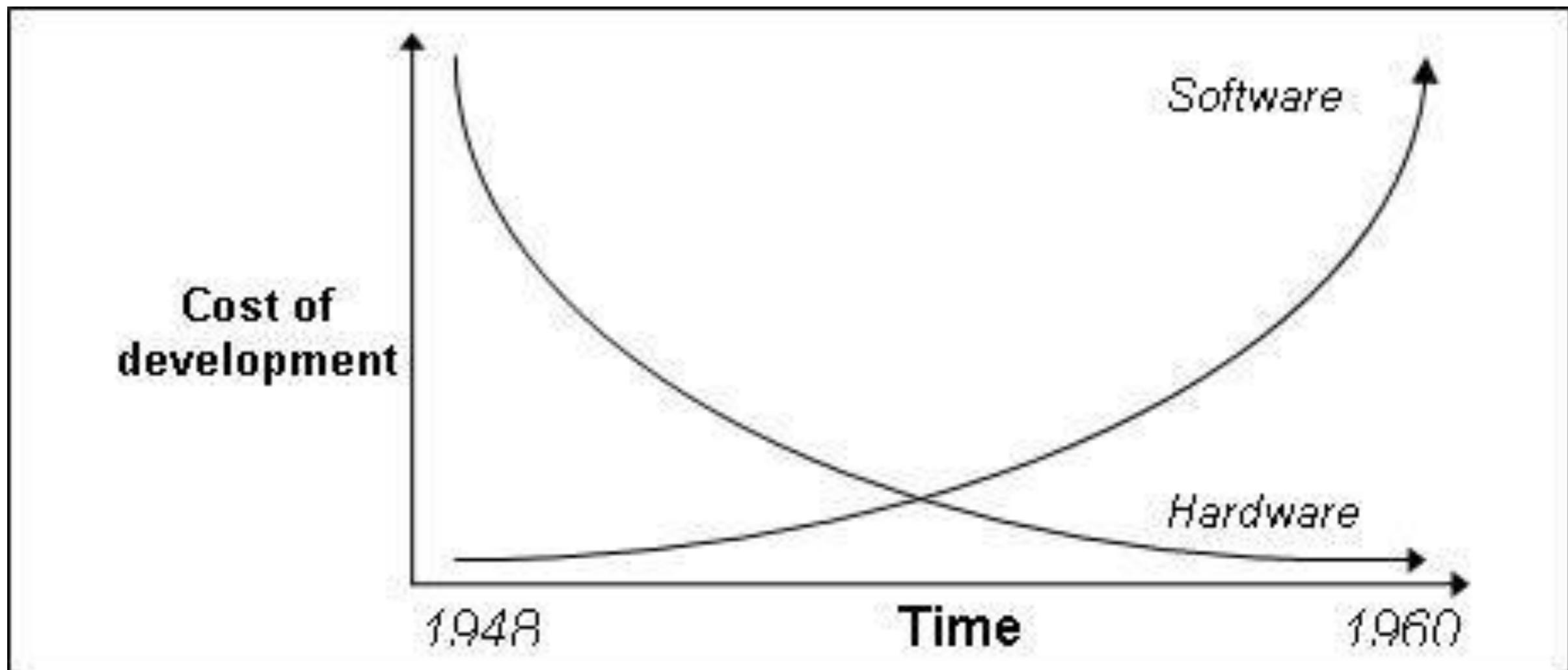
High Quality software that Works and Fits Budget



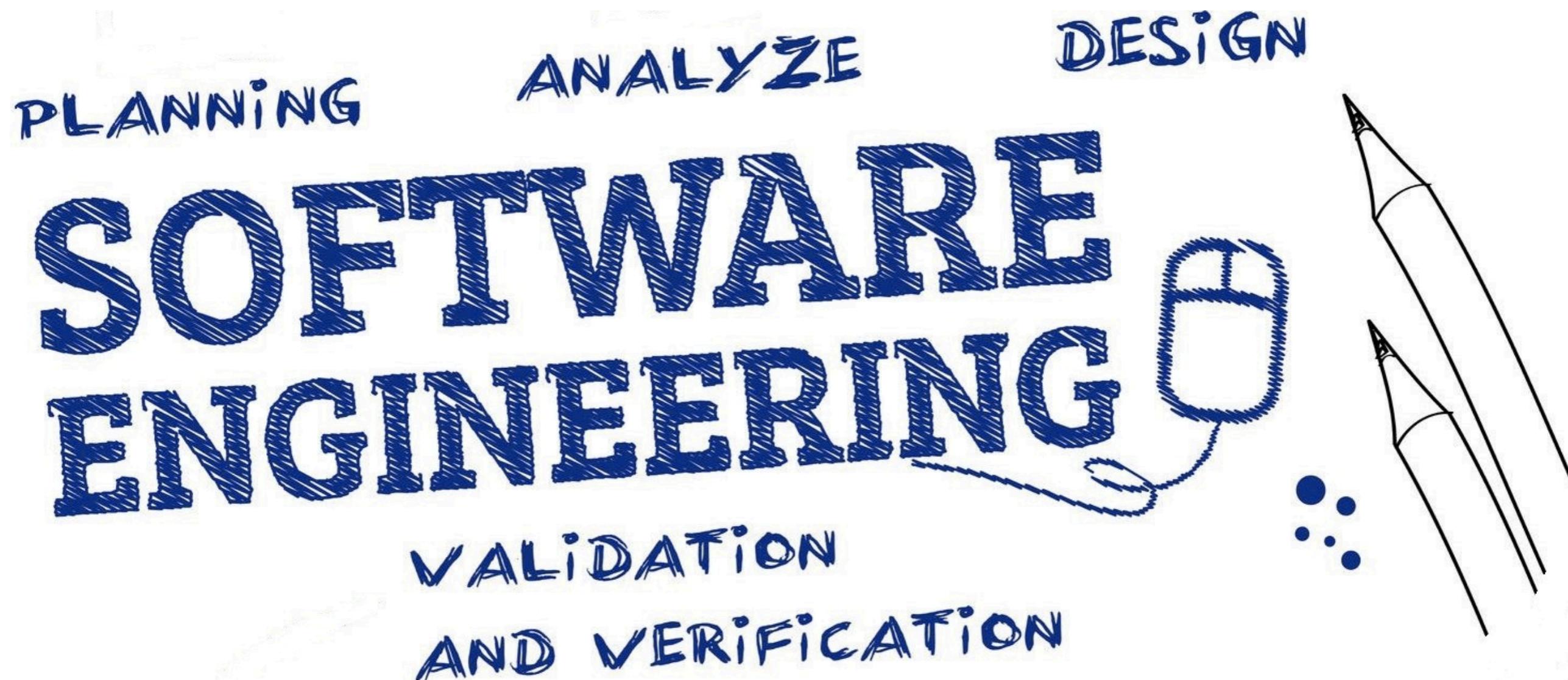
# WHAT IS SOFTWARE ENGINEERING ? WHY DO WE NEED IT?

<https://youtu.be/gBd-ct58DCI>

# Crise no Desenvolvimento de Software



# Qual a importância da Engenharia de Software



Curso de Processo de Software -Aula 1 de 32

Dúvidas: [naubergois@ufc.br](mailto:naubergois@ufc.br)

Então, software é um produto do engenheiro de software, como um hardware é um produto de um engenheiro eletrônico ? O que diferencia estes produtos?

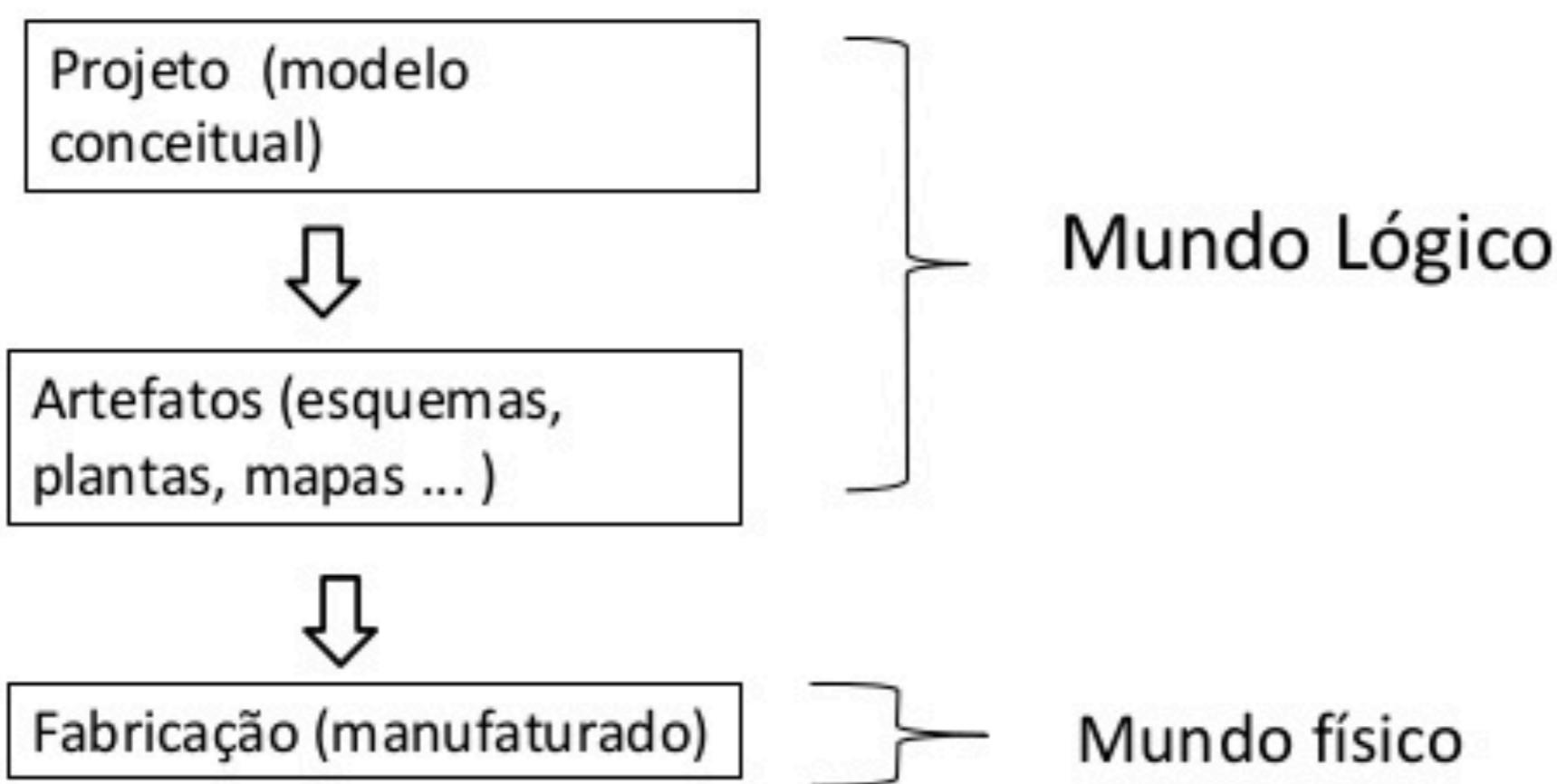


Software é lógico.

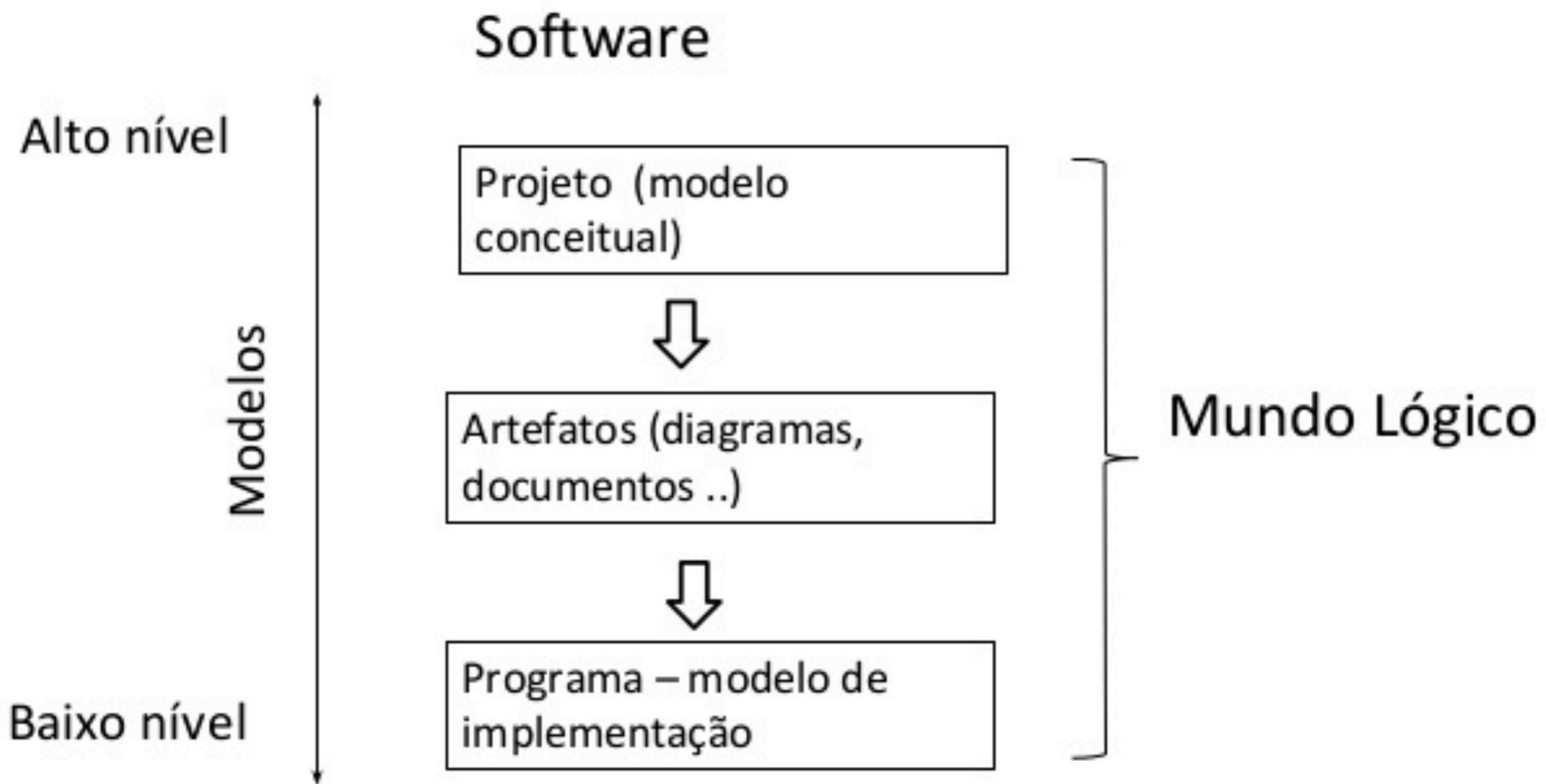
Hardware é físico.

1. Desenvolvido ou projetado por engenharia,  
não manufaturado no sentido clássico.

### Hardware - manufaturado

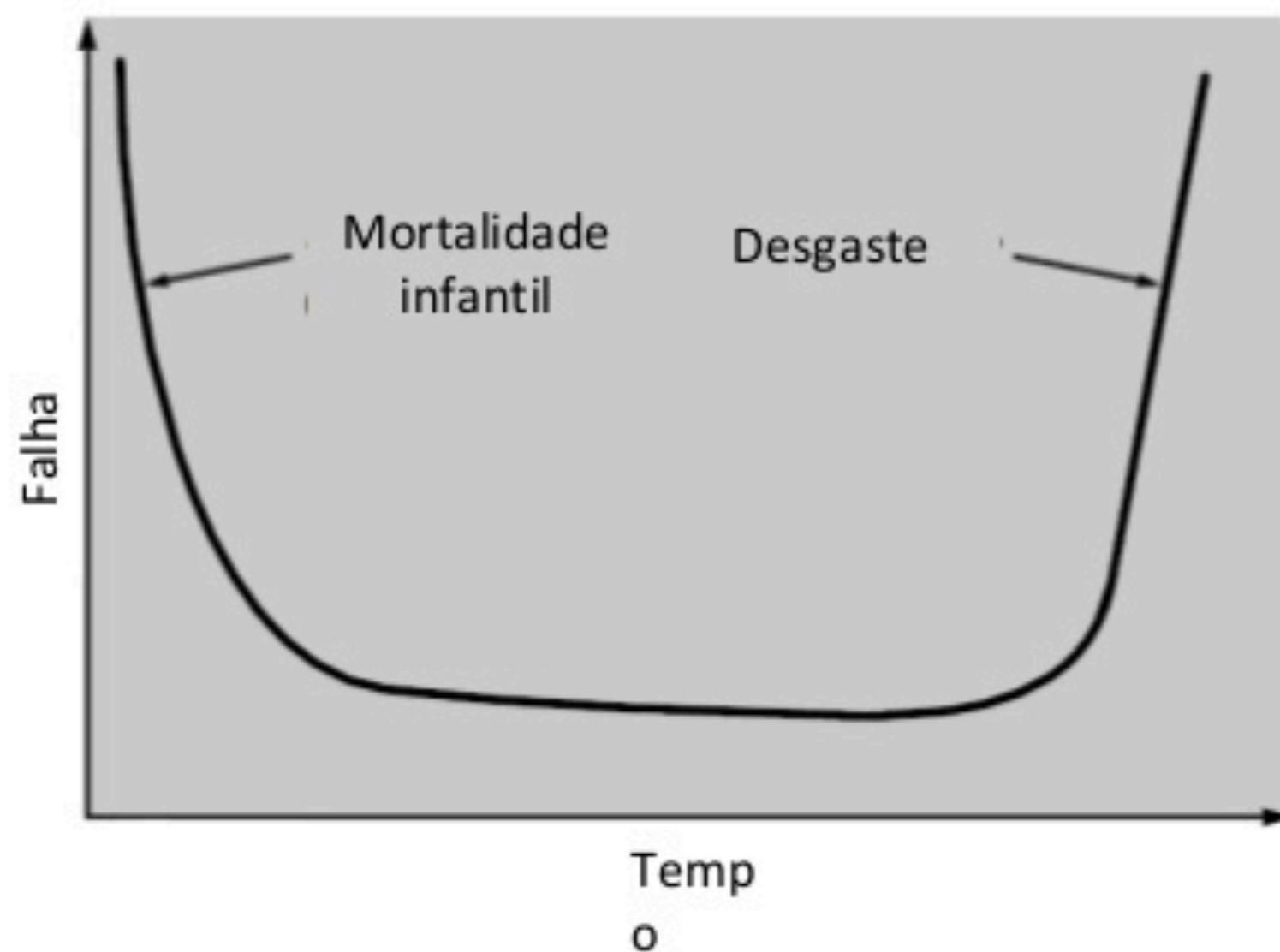


1. Desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico.



# Curva de Falha

## Curva de falha do hardware



**Mortalidade infantil**

Associada a falhas de fabricação e ou projeto.

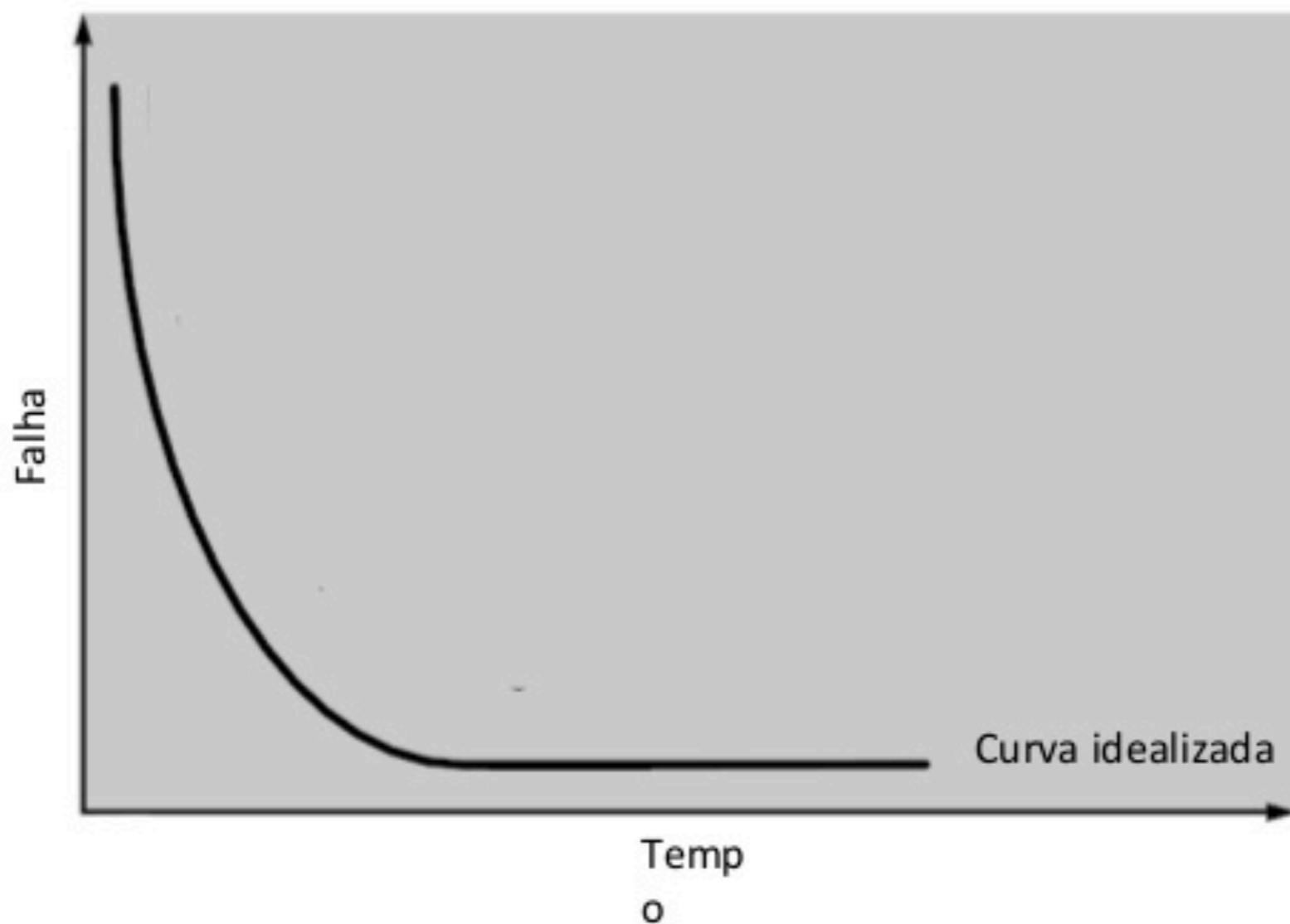
**Desgaste**

Males ambientais, poeiras, vibrações.

Todo hardware tem um tempo de vida.

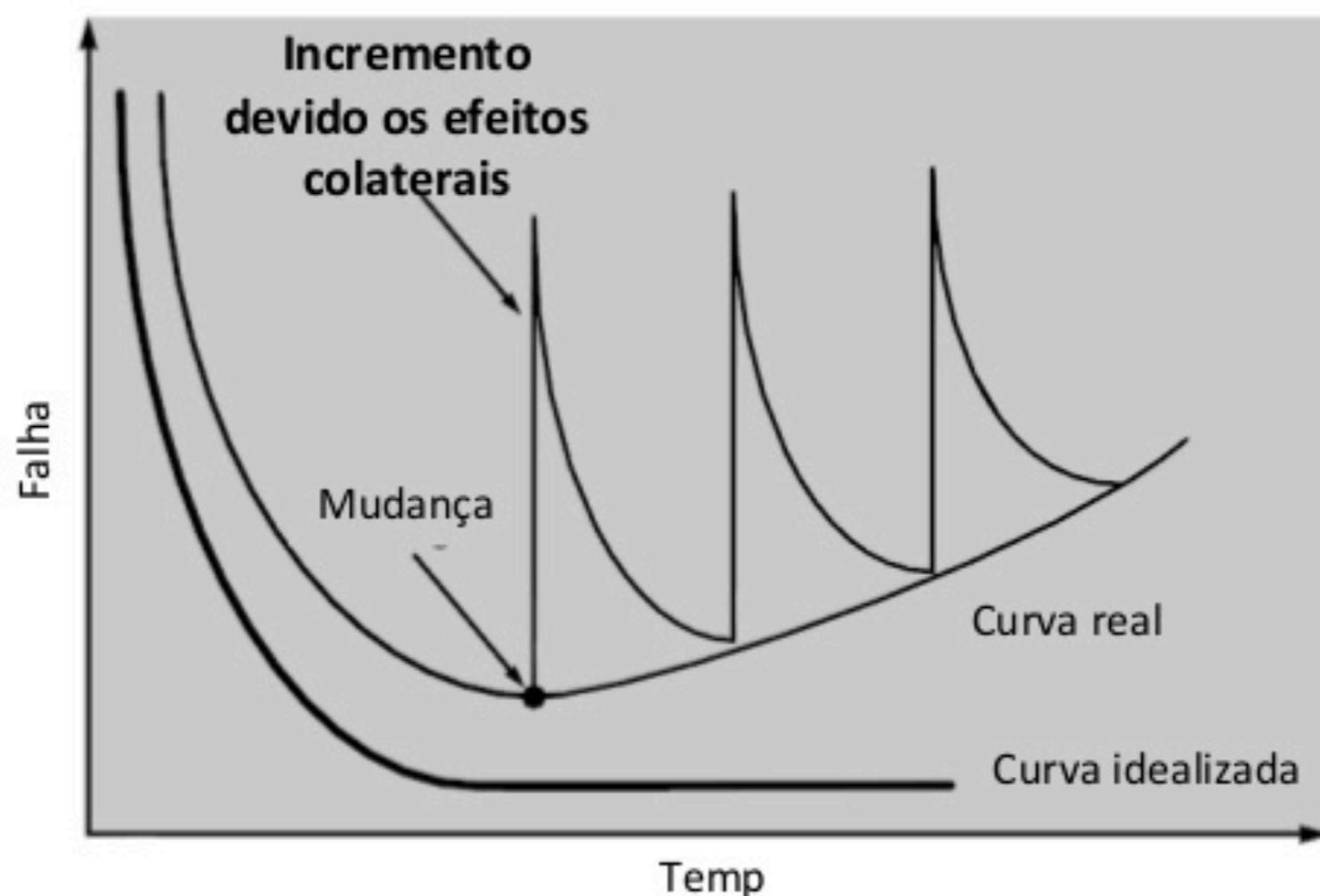
# Curva de Falha

## Curva de falha do software



# Curva de Falha

## Curva de falha do software



# Evolução do Software

## Evolução do Software

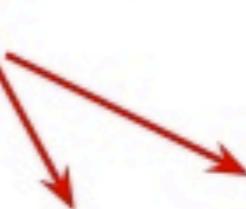
### Crise do software

#### Os primeiros anos

- ORIENTAÇÃO BATCH
- DISTRIBUÍÇÃO 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

#### A quarta era

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

1950

1960

1970

1980

2000

# Crise do Software

## A crise do software

### + Complexidade

Aumento crescente por sistemas de Informação

Sistemas mais e mais sofisticados exigem mais recursos (humanos e máquinas)

### - Confiabilidade

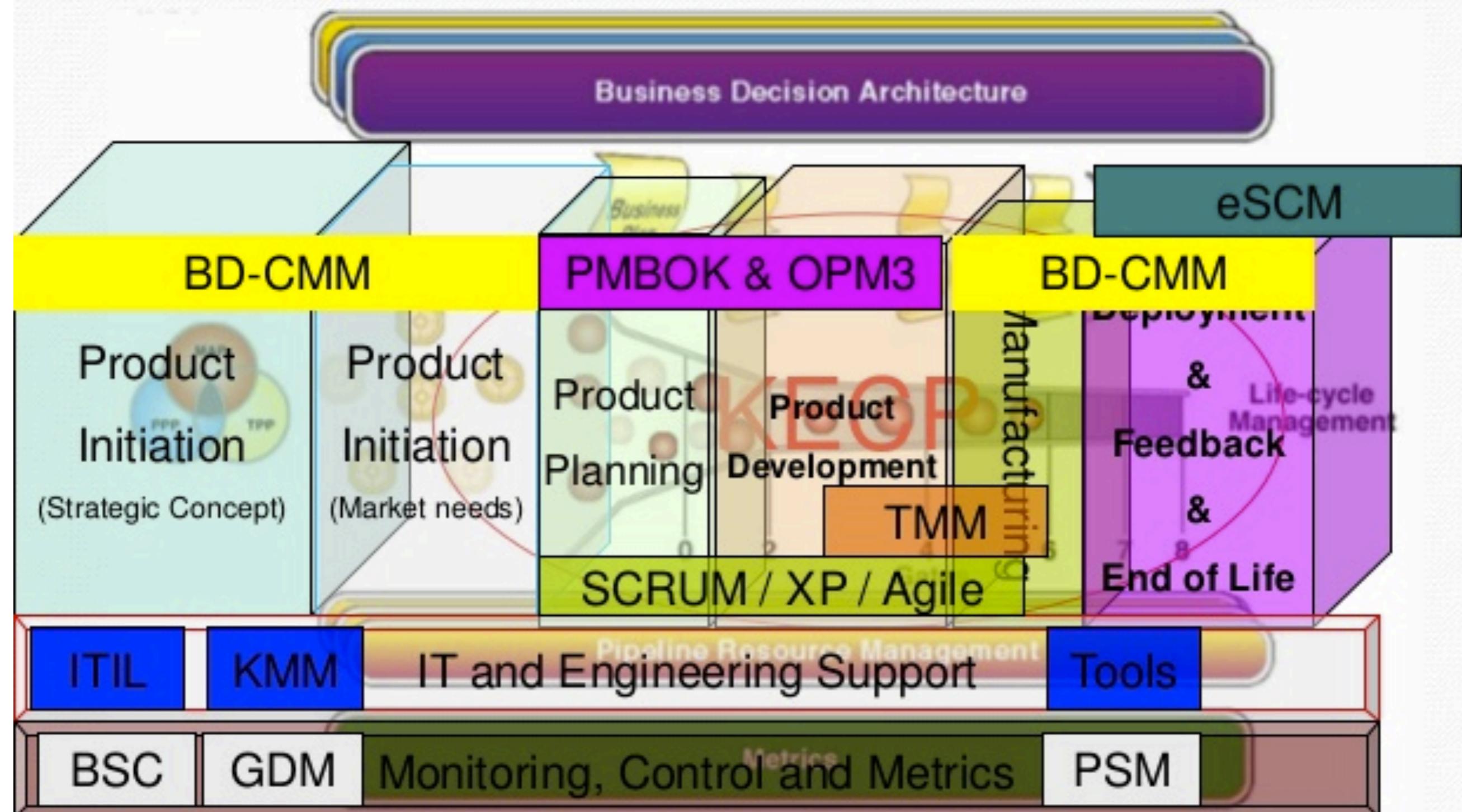
Mais dependência do software nos procedimentos normais do cotidiano

Sistemas devem ser mais e mais seguros.

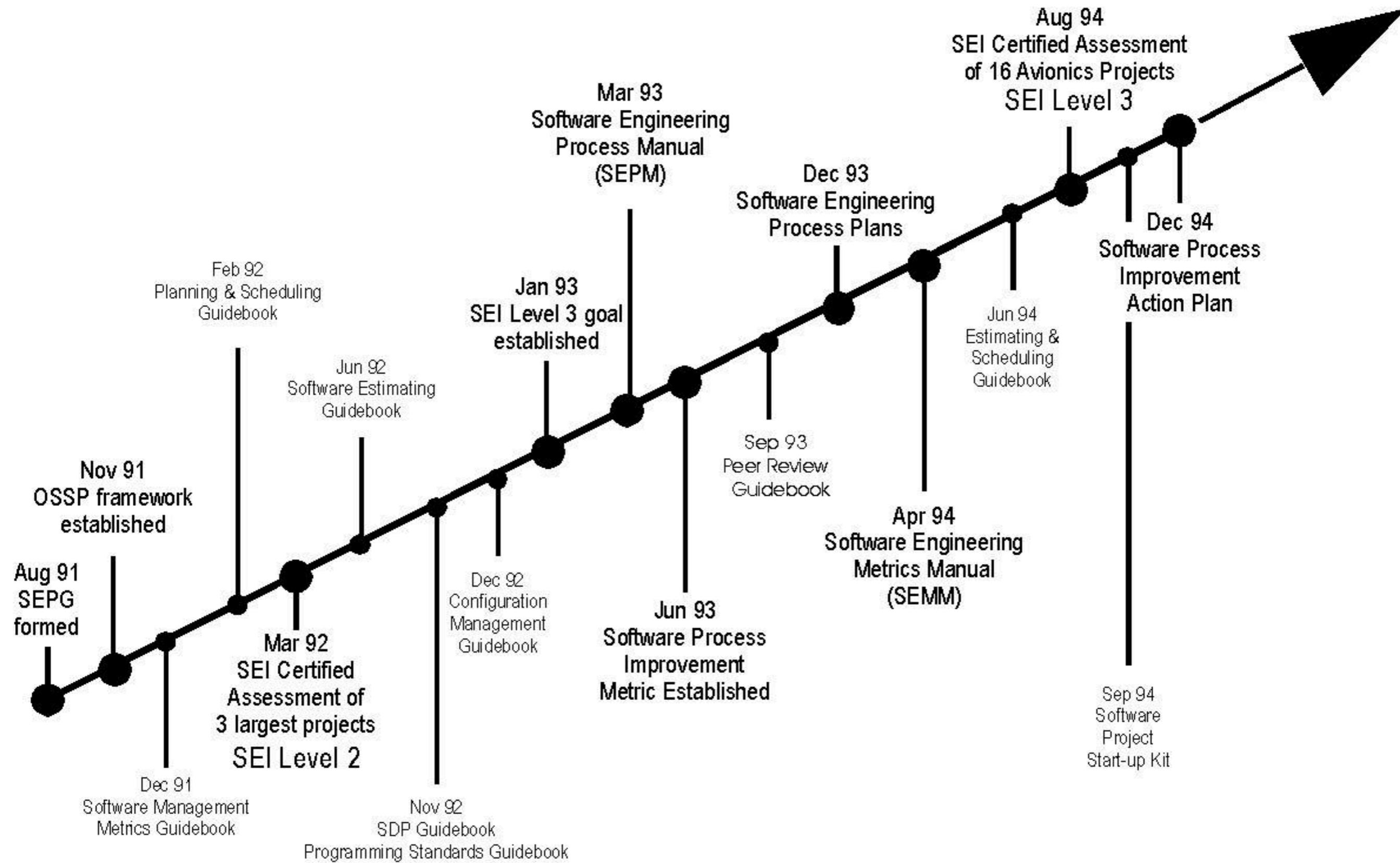
# Processos de Software

## The Expected Coverage

P-CMM / SSE-CMM



## SOFTWARE PROCESS IMPROVEMENT HISTORY AT MDA (ST. LOUIS)



# Modelos de Ciclo de Vida de Software



## Modelos de Ciclo de Vida de Software

- São a progressão dos projetos, desenvolvimento, manutenção e substituição de software;
- Descrições abstratas do processo de desenvolvimento de software;

# Modelos de Ciclo de Vida de Software



## Life-Cycle Model

- It specifies the various phases/workflows of the software process, such as the requirements, analysis (specification), design, implementation, and postdelivery maintenance, and the order in which they are to be carried out.

# Principais Modelos de Ciclo de Vida



## Principais Modelos

- Modelo clássico (ou em cascata)
- Prototipagem (ou Prototipação)
- Modelo espiral
- Modelo Incremental
- Modelo RAD

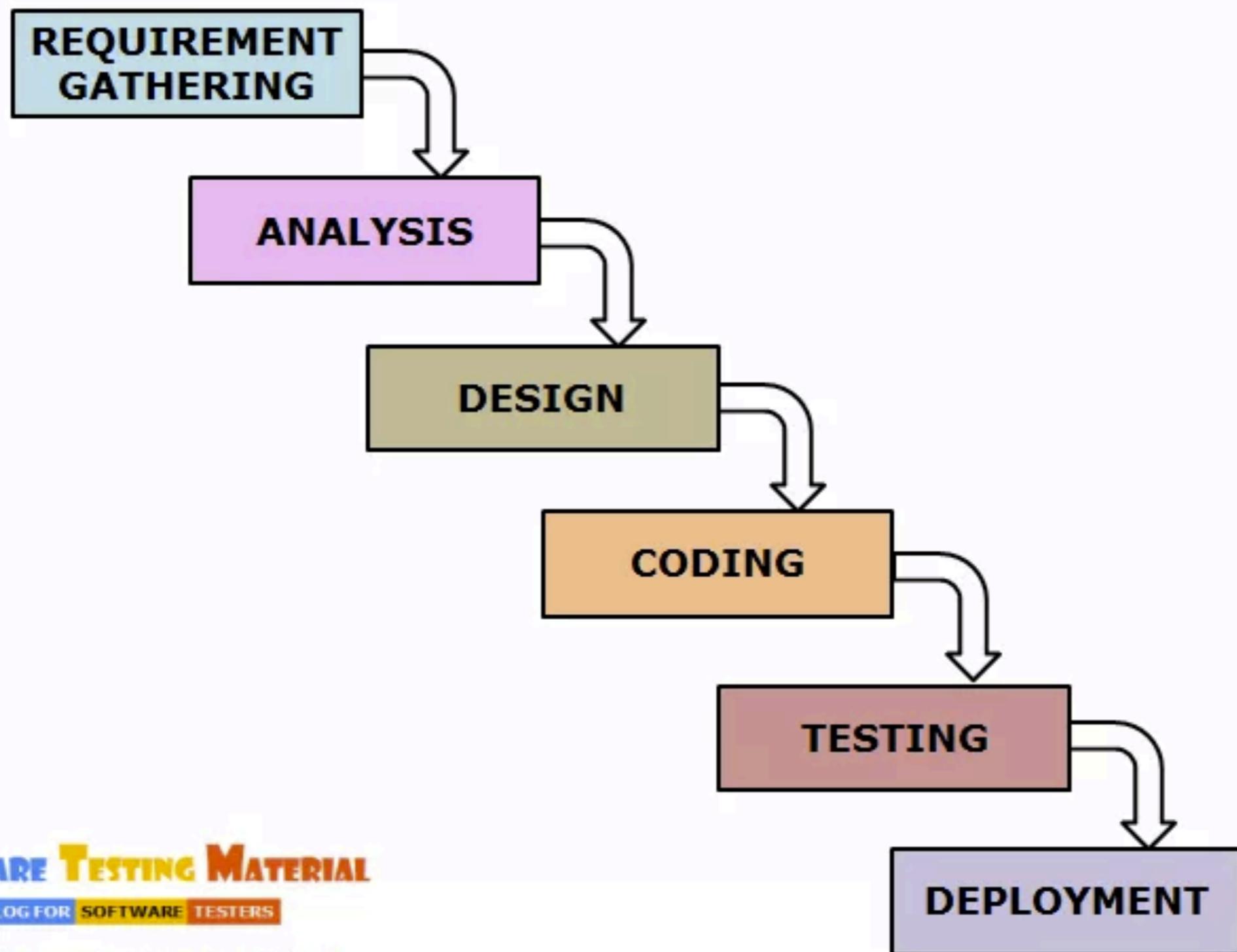
# Principais Modelos de Ciclo de Vida



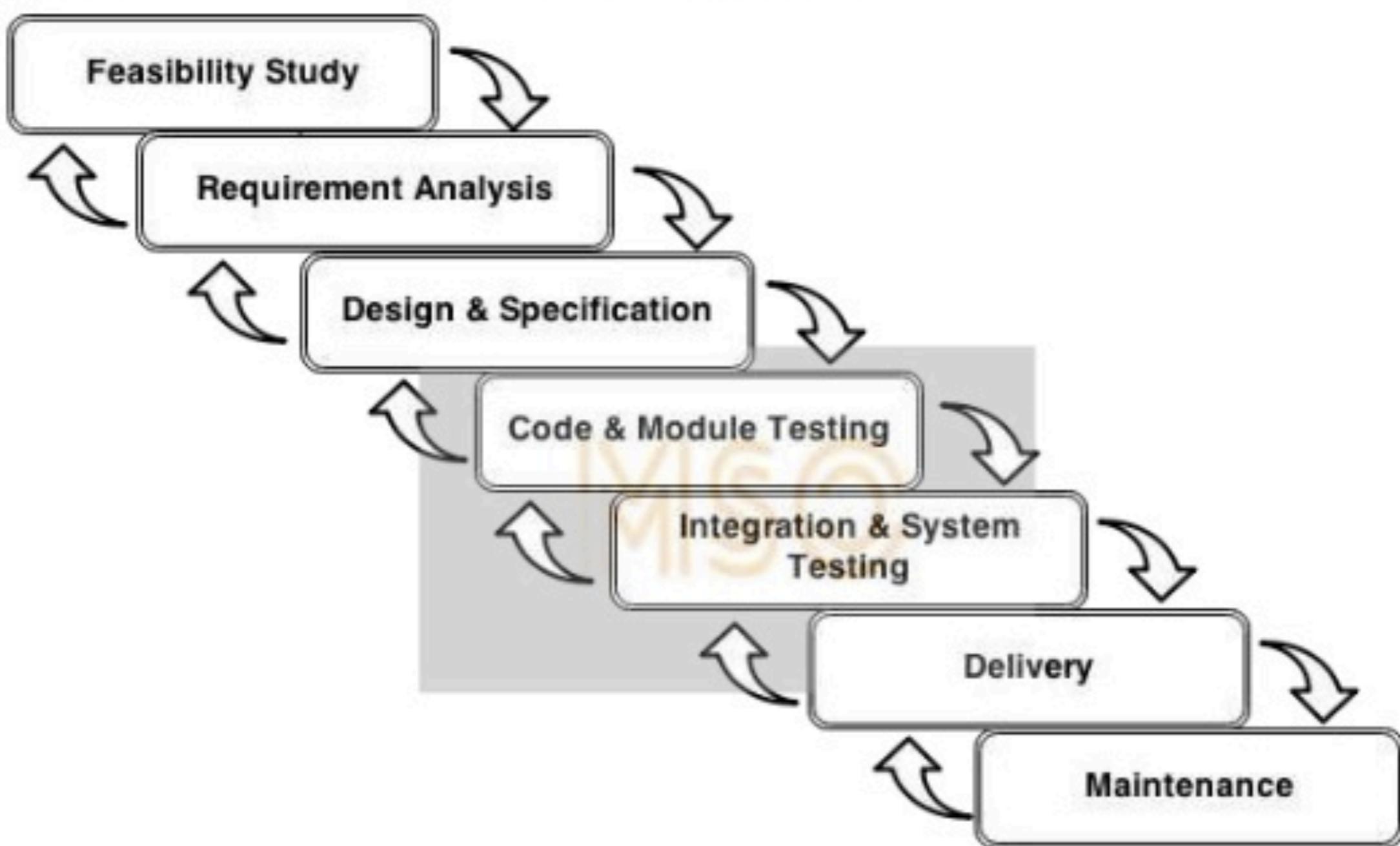
## Modelo Cascata

- Este é o modelo mais antigo mas ainda o mais usado;
- Segue uma sequência linear;

# Waterfall Model - SDLC



## FIG 2 : WATERFALL MODEL



In the waterfall model "With feedback the process is structured as a cascade of phases where the output of one phase constitute the input to the next one.

# Modelo Cascata

## Vantagens:

- Oferece uma maneira de tornar o processo mais visível;
- Facilita o planejamento;

# Modelo Cascata

## Desvantagens:

- Projetos reais raramente seguem o fluxo sequencial;
- Os requisitos se alteram durante o projeto;

# Principais Modelos de Ciclo de Vida



## Prototipação

- O objetivo é entender os requisitos do usuário;
- Possibilita que o desenvolvedor crie um modelo (protótipo) do software que deve ser construído, podendo ser no papel ou executável;

# Principais Modelos de Ciclo de Vida - Prototipação



## Vantagens:

- Protótipos contribuem para melhorar especificação dos requisitos;
- Partes do protótipo podem vir a ser usadas no desenvolvimento do sistema final;

# Principais Modelos de Ciclo de Vida - Prototipação



## Desvantagens:

- Custo elevado;
- Atraso no desenvolvimento ;