



**UNIVERSIDADE FEDERAL DE RORAIMA**  
**CENTRO DE CIÊNCIA E TECNOLOGIA**  
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**  
**DCC402 – Engenharia de Software I (2023.1)**  
**Prof. Thais Oliveira Almeida**

---

# AULA 4:

## DETERIORAÇÃO DE SOFTWARE

---

# Desenvolvimento de Software x Hardware

---

- ❖ Custos do software são concentrados na engenharia.
- ❖ *Software se desgasta?*
- ❖ Software não desgasta, mas se deteriora.
- ❖ *Fase de fabricação de um hardware pode produzir problemas de qualidade.*
- ❖ Quando um hardware falha, pode substituir por um sobressalente.

# Atividades

---

## Levantamento de requisitos

- Explorar os conceitos
- Elicitar as necessidades do cliente

## Análise (especificação)

- Analisar os requisitos do cliente
- Documentar os requisitos
- Planejar o desenvolvimento

# Atividades

---

## Projeto

- Projeto arquitetural e projeto detalhado

## Implementação

- Codificação

## Manutenção pós-entrega

- Corretiva, aperfeiçoamento, adaptativa

# Manutenção (Evolução)

---

## Entendimento clássico: (IEEE 1990)

- Alterações depois da entrega e instalação.
- Se a falha ou requisito é descoberto durante o desenvolvimento é tratado como parte do desenvolvimento, caso contrário é manutenção.

## Entendimento moderno: (ISO/IEC 1995 e IEEE 1998)

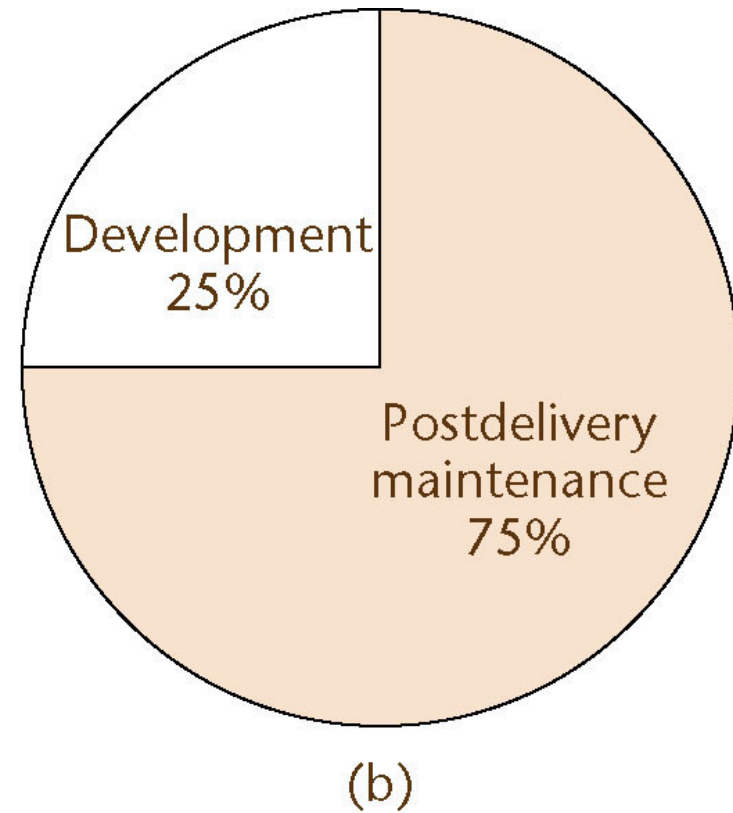
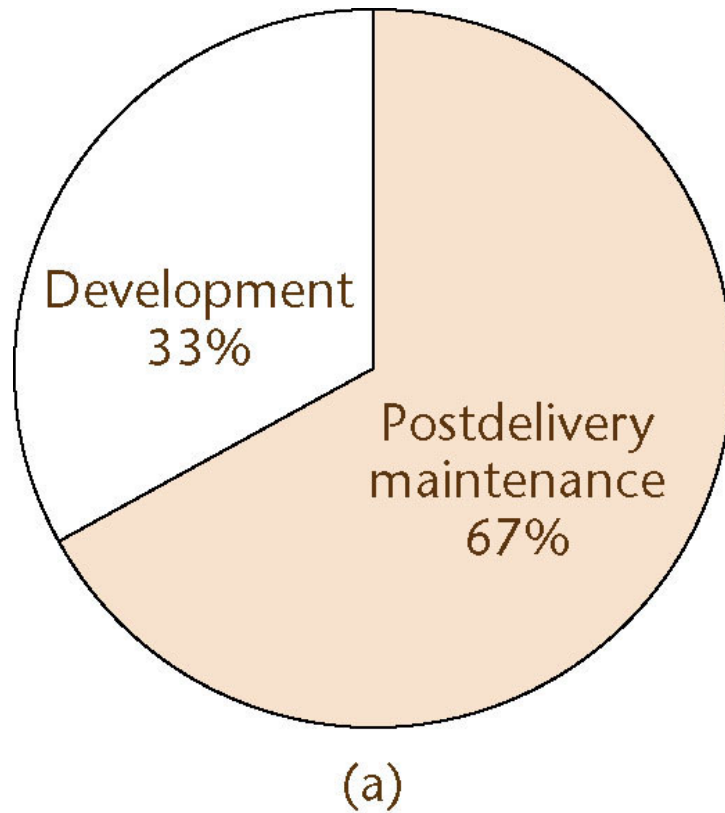
- O processo que ocorre quando um artefato de software é modificado por conta de um problema ou por necessidade de melhoria ou adaptação.
- (Ocorre sempre o software é modificado, independente da fase).

Software é um modelo da realidade, que se altera constantemente.

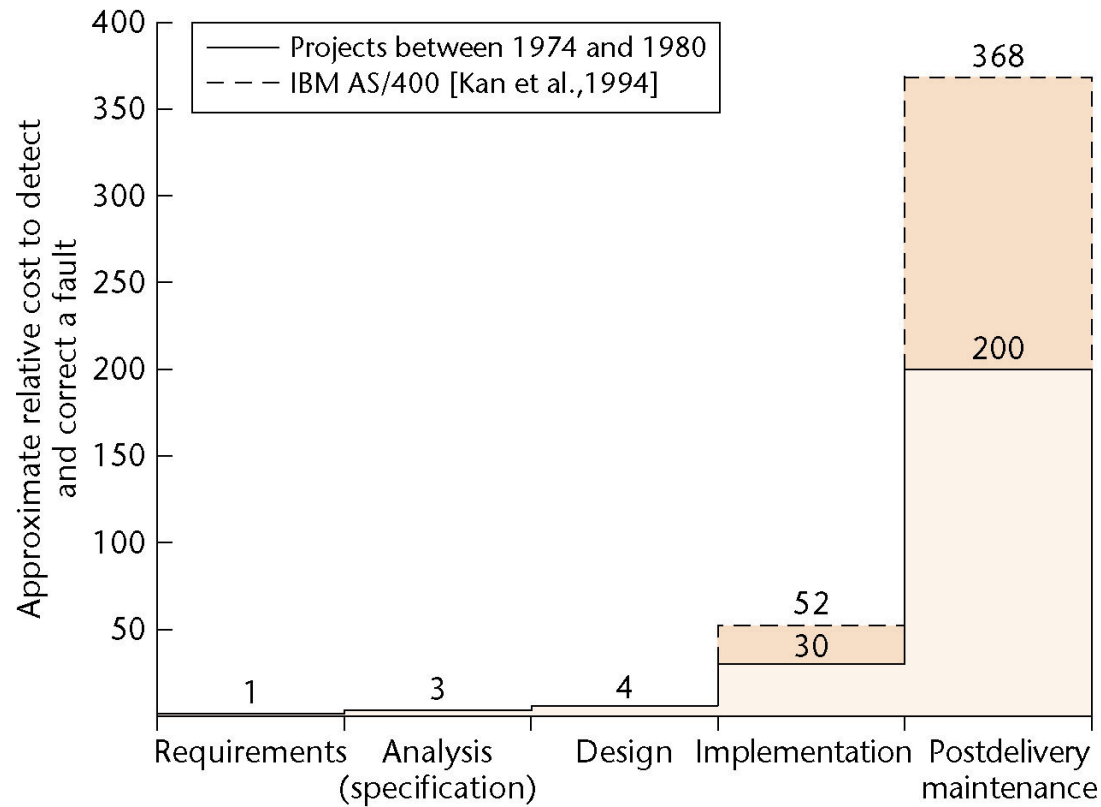
**50% da manutenção pós-entrega é para correção!**

# Desenvolvimento x Manutenção

---



# Custo de Uma Falha



# Falhas

---

- ❖ Entre 60% a 70% das falhas são nos requisitos, análise ou projeto (Boehm, 1970).
- ❖ Example: Jet Propulsion Laboratory inspections (1992).
  - ❖ 1.9 falhas por página de especificação.
  - ❖ 0.9 por página de projeto.
  - ❖ 0.3 por página de código.



# Falhas

---

- ❖ Corrigir uma falha em fases mais avançadas do desenvolvimento demanda:
  - ❖ Alterar código e documentação.
  - ❖ Executar teste de regressão.
  - ❖ Reinstalar o produto nos clientes.

# Testes

---

- ❖ Engano => Imperfeição no código => Resulta em um falha => Manifesta um erro [IEEE 610.12, 1990]. Defeito é uma palavra genérica para imperfeição, falha ou erro.
- ❖ Verificação no final de cada fase (pode ser tarde demais).
- ❖ Validação no final do projeto (mais tarde ainda).

# Testes

---

- ❖ Atividades contínuas de testes devem ser feitas durante o projeto.
  - ❖ Teste de unidade;
  - ❖ Teste de integração;
  - ❖ Teste de aceitação.
- ❖ Grupo de qualidade.
- ❖ Grupo de teste.

# Documentação

---

- ❖ Não deve ser feita somente após o final do desenvolvimento.
  - ❖ Rotatividade de membros da equipe.
  - ❖ Para iniciar uma fase é necessário que a anterior esteja documentada.
  - ❖ Documentação facilita o teste.
  - ❖ Manutenção.
  
- ❖ Problema: atualização da documentação.

# Os Envolvidos no Processo

---

❖ **Stakeholders** (qualquer pessoa ou organização que tenha interesse, ou seja afetado pelo projeto, de forma direta ou indireta, positiva ou negativamente).

- Gerente de Projeto.
- Analista de Sistema.
- Programador.
- Patrocinador.
- Cliente (usuário).

# Stakeholders Internos

---

- ❖ São os que estão **dentro** do ambiente da empresa. Alguns exemplos:
  - ❖ Gestores da empresa;
  - ❖ Colaboradores;
  - ❖ Acionistas.

# Stakeholders Externos

---

- ❖ São os que estão **fora** do ambiente da empresa, mas que interagem com ela de alguma forma. Alguns exemplos:
  - ❖ Fornecedores;
  - ❖ Concorrentes;
  - ❖ Clientes;
  - ❖ Familiares dos clientes;
  - ❖ Governo;
  - ❖ ONG's;
  - ❖ Mídia;
  - ❖ Sindicatos;
  - ❖ Meio ambiente.

# Padronizações

---

ISO – International Organization for Standardization.

- ❖ Além de ser um acrônimo, é derivado da palavra grega que significa “igual”.
- ❖ Rede de institutos nacionais de padronização de 147, com sede em Genebra, Suíça.