

Resumo – Engenharia de Software

1. Visão Geral do Software

- Conjunto de **programas, dados e documentos** (impressos e virtuais).
 - **Programas**: instruções que produzem funções e desempenho desejados.
 - **Estruturas de dados**: permitem manipulação da informação.
 - **Documentos**: descrevem operação e uso.
 - **Importância**: *Software é mais crucial que hardware para o sucesso de sistemas* (Pressman, 2016).
-

2. Evolução do Software

- **1ª Era (1950–1965)**: batch, software customizados, poucos profissionais.
 - **2ª Era (1965–1975)**: multiusuário, tempo real, bancos de dados, produtos de software.
 - **3ª Era (1975–1988)**: sistemas distribuídos, inteligência embutida, hardware barato.
 - **4ª Era (1988–2000)**: desktops poderosos, orientação a objetos, IA, redes neurais, computação paralela.
 - **5ª Era (2000–atual)**: realidade virtual, computação móvel, explosão da internet, software livre.
-

3. Importância do Software

- Antes (1950–1988): foco em **hardware barato e eficiente**.
 - Hoje: **melhorar qualidade e reduzir custos do software**.
 - Impacta todos os aspectos da vida.
-

4. Características do Software

- **Desenvolvido**, não manufaturado.
- **Custos concentrados no trabalho de engenharia**.
- **Não desgasta** fisicamente (falhas = erros de projeto ou implementação).
- Manutenção é **mais complexa que no hardware**.
- **Curva de falhas**:
 - Hardware: falhas aumentam com o tempo (desgaste).
 - Software: falhas só mudam quando há **alterações** (mudanças de requisitos, manutenção).
- **Situação atual**:
 - Antes: software sob medida.
 - Agora: **reuso e orientação a objetos** para reduzir custos.

5. Aplicações de Software

- **Básico** (sistemas operacionais, compiladores).
- **Tempo real** (controladores de processos, aviões).
- **Comercial** (webapps, ERPs).
- **Científico/Engenharia** (simulações, CAD).
- **Embutido/Embarcado** (firmware, IoT).
- **Aberto** (open source).
- **IA** (redes neurais, aprendizado de máquina).

6. Desenvolvimento de Software

- Anos 50–60: **sem planejamento**, desenvolvimento ad hoc.
- Problemas que causaram **crise do software**:
 - Estimativas ruins de tempo e custo.
 - Falta de habilidades técnicas.
 - Problemas de manutenção.
 - **Baixa qualidade** do produto final.

7. Crise do Software e Mitos

- **Administrativos**:
 - “Um manual resolve tudo”.
 - “Computador moderno resolve problemas”.
 - “Adicionar mais programadores recupera atraso” (*Mito da Horda de Mongóis*).
- **Do Cliente**:
 - “Declaração geral é suficiente para começar”.
 - “Mudanças são fáceis porque software é flexível”.
- **Do Profissional**:
 - “Se funciona, está pronto”.
 - “Quanto antes codificar, mais rápido termino”.
 - “Qualidade só pode ser avaliada com o software rodando”.
 - “Somente o programa funcionando é entregue”.
 - “Engenharia de software atrasa o projeto”.

8. Engenharia de Software – Conceito

- **Abordagem sistemática e disciplinada** para desenvolvimento, operação e manutenção do software.
- Baseia-se em **princípios de engenharia** para garantir confiabilidade e eficiência.

- **Elementos fundamentais:**
 - **Métodos** (como fazer).
 - **Ferramentas** (apoio automatizado).
 - **Procedimentos** (ligam métodos e ferramentas).
 - **Paradigmas:** diferentes modelos de processo.
-

9. Processo de Software

- Define **fases, atividades e tarefas**.
 - **Três fases básicas:**
 - **Definição:** o que será feito (levantamento de requisitos).
 - **Construção:** implementação e testes.
 - **Manutenção:** correção, aperfeiçoamento e adaptação.
 - **Correção de falhas.**
 - **Aperfeiçoamento** (novos recursos).
 - **Adaptação** (novos ambientes).
-

Exemplo Prático de Erros Clássicos

- Cliente diz: “Quero um sistema de vendas simples”.
 - Desenvolvedor começa a codificar sem detalhar.
 - Depois cliente pede relatórios, integração com estoque → **mudanças complexas e caras.**
 - Boa prática: aplicar **engenharia de requisitos** antes da codificação.
-

✓ Checklist para Prova

- ✓ Definição de software e sua importância.
- ✓ Evolução (5 eras e características principais).
- ✓ Diferenças entre software e hardware.
- ✓ Curva de falhas (hardware x software).
- ✓ Aplicações do software (exemplos).
- ✓ Crise do software e mitos (cliente, administrador, profissional).
- ✓ Conceito e elementos da engenharia de software.
- ✓ Fases do processo de software.