

## Introdução a Eng. De Software II

Donizete C. Bruzarsco

### Problemas relacionados a produção de software

- A sofisticação do hardware ultrapassa nossa habilidade de construir software que utilize todo o potencial do hardware.
- Nossa habilidade para construir novos produtos de software não acompanha a demanda.
- Nossa habilidade de manter os produtos de software existentes estão ameaçadas por projetos ineficientes e recursos inadequados (pessoal mal treinado ou mal alocado).

### Sintomas e causas dos problemas

- mau entendimento das necessidades dos usuários finais
- falta de habilidade para tratar com modificações de requisitos;
- módulos que não "trabalham" juntos;
- software que é difícil de manter ou estender;
- descoberta, de forma tardia, das falhas de projeto;
- baixa qualidade de software;
- desempenho inaceitável;
- os membros da equipe não conseguem reconstruir pois não sabem quem modificou o que, quando, onde e porque;
- um processo "irresponsável" de construir- entregar

### Que resultaram de ...

- gerenciamento adhoc dos requisitos
- comunicação ambígua e imprecisa;
- arquiteturas frágeis;
- grande complexidade;
- inconsistência detectada em requisitos, projetos e implementação;
- testes insuficientes;
- avaliação subjetiva do estado do projeto;
- falha em atacar/avaliar riscos;
- propagação incontrolada de modificações;
- automação insuficiente.

### Tamanho e complexidade

- Fatores que influenciam no tamanho:
  - Contexto (escopo);
  - Número de interfaces com outros sistemas;
  - Número e tipo de usuários;
  - Número de entidades (dados) e atributos.

### Engenharia de Software

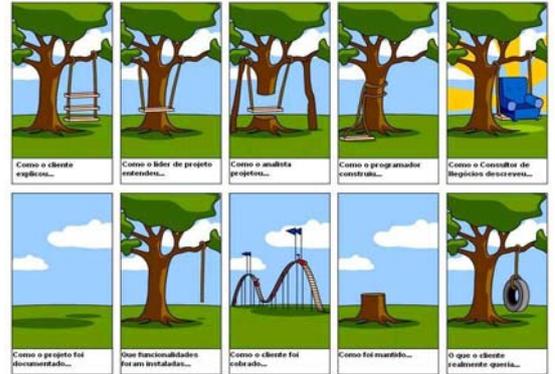
- O estabelecimento e uso de **sólidos princípios de engenharia** para que se possa obter **economicamente** um software que seja **confiável** e que **funcione** efetivamente em máquinas reais [Fritz Bauer]

## Tamanho e complexidade

### Fatores que influenciam a complexidade:

- Complexidade do processo de negócio que precisa ser automatizado;
- Complexidade das regras de negócio;
- Volume de transações para o processo por unidade de tempo;
- Localização geográfica dos usuários;
- Complexidade das relações que existem entre as várias entidades que usam o sistema;
- Tipo e nível de maturidade que é requerido para executar o sistema.

## Análise, Projeto e Programação de Sistema



## Qualidade de Produtos de Software – ISO 9126 – NBR 13596

| Característica                             | Sub-característica  | Pergunta chave para a subcaracterística   |
|--|---------------------|---|
| Funcionalidade (satisfaz as necessidades?) | Adequação           | Propõe-se a fazer o que é apropriado?     |
|  | Acurácia            | Faz o que foi proposto de forma correta?  |
|  | Interoperabilidade  | Interage com os sistemas especificados?   |
|  | Conformidade        | Está de acordo com as normas, leis, etc.? |
|  | Segurança de acesso | Evita acesso não autorizado aos dados?    |

| Característica                     | Sub-característica  | Pergunta chave para a subcaracterística      |
|------------------------------------|---------------------|--|
| Confiabilidade (é imune a falhas?) | Maturidade          | Com que frequência apresenta falhas?         |
|                                    | Tolerância a falhas | Ocorrendo falhas, como ele reage?            |
|                                    | Recuperabilidade    | É capaz de recuperar dados em caso de falha? |
| Usabilidade (é fácil de usar?)     | Intelegibilidade    | É fácil entender o conceito e a aplicação?   |
|                                    | Apreensibilidade    | É fácil aprender a usar?                     |
|                                    | Operacionalidade    | É fácil de operar e controlar?               |

| Característica                           | Sub-característica | Pergunta chave para a subcaracterística               |
|--|--------------------|---|
| Eficiência (é rápido e "enxuto"?)        | Tempo              | Qual é o tempo de resposta, a velocidade de execução? |
|  | Recursos           | Quanto recurso usa? Durante quanto tempo?             |
| Manutenibilidade (é fácil de modificar?) | Analísabilidade    | É fácil de encontrar uma falha, quando ocorre?        |
|  | Modificabilidade   | É fácil modificar e adaptar?                          |
|  | Estabilidade       | Há grande risco quando se faz alterações?             |

## Engenharia de Software

↴ Abrange um conjunto de três elementos fundamentais:

- Métodos
- Ferramentas
- Procedimentos

**Métodos:** proporcionam os detalhes de como fazer para construir o software

**FERRAMENTAS:** dão suporte automatizado aos métodos

**PROCEDIMENTOS:** constituem o elo de ligação entre os métodos e ferramentas

## Processo de Software Manutenção

**FASE DE MANUTENÇÃO:** concentra-se nas "mudanças" que ocorrerão depois que o software for liberado para uso operacional

- ↕ Correção
- ↕ Adaptação
- ↕ Melhoramento funcional

## Modelos de Processo de Desenvolvimento de Software

☒ Modelo de ciclo de vida

Descrições abstratas do processo de desenvolvimento e modificação, tipicamente, mostrando os principais estágios de desenvolvimento e manutenção de um software executável.

☒ Alguns ciclos de vida mais conhecidos são:

- Ciclo de vida clássico
- Prototipação
- Modelo Espiral
- Técnicas de 4a. geração

## O que foi feito para atacar os problemas da crise de software

- 🌐 Programação estruturada
- 🌐 Análise estruturada
- 🌐 Sistemas de gerenciamento de banco de dados
- 🌐 Linguagens de 4a. Geração
- 🌐 Verificação, validação e teste de software
- 🌐 Métodos formais (sequenciais, concorrentes)
- 🌐 Análise da complexidade de algoritmos
- 🌐 Tipos abstratos de dados
- 🌐 Análise orientada a objetos
- 🌐 Modelagem e automação de processos
- 🌐 Ferramentas CASE

## Bons Exemplos de Aplicativos de software

- Sistemas telefônicos
- Sistemas bancários
- Sistemas de reservas de passagens aéreas
- Sistemas administrativos: folha de pagamento, contabilidade, etc.

## CONCLUSÃO

**Engenharia de software** pode ser vista como uma abordagem de desenvolvimento de software elaborada com disciplina e métodos bem definidos.