

Software: Evolução, Problemas e Mitos

Disciplina: **Engenharia de Software I**
Prof. Me. Fernando Roberto Proença

Evolução do software

- 1950 - 1965
 - ❑ O hardware sofreu contínuas mudanças
 - ❑ O software era uma arte "secundária" para a qual havia poucos métodos sistemáticos
 - ❑ O hardware era de propósito geral
 - ❑ O software era específico para cada aplicação
 - ❑ Não havia documentação

2

Evolução do software

- 1965 - 1975
 - ❑ Multiprogramação e sistemas multiusuários
 - ❑ Técnicas interativas
 - ❑ Sistemas de tempo real
 - ❑ 1ª geração dos Sistemas Gerenciador de Banco de Dados (SGBD)
 - ❑ Produto de software - *software houses*
 - ❑ Bibliotecas de Software
 - ❑ Cresce o número de sistemas baseado em computador
 - ❑ Manutenção quase impossível

3

Evolução do software

- 1975 - hoje
 - ❑ Sistemas distribuídos
 - ❑ Redes locais e globais
 - ❑ Uso generalizado de microprocessadores – produtos inteligentes
 - ❑ Hardware de baixo custo
 - ❑ **Impacto de consumo**

4

Evolução do software

- Quarta era do software de computador
 - ❑ Tecnologias orientadas o objetos
 - ❑ Sistemas especialistas e software de inteligência artificial usados na prática
 - ❑ Software de rede neural artificial
 - ❑ Computação Paralela

5

Evolução do software

- Software cada vez...
 - ❑ Maior
 - ❑ Mais funcionalidades
 - ❑ Mais complexo
 - ❑ Abrangendo mais usuários
 - ❑ Exigindo a participação de equipes em seu desenvolvimento.

6

Exemplos de software

- Ariane 5 – 1996
 - O foguete explodiu 40 segundos após a sua primeira decolagem (prejuízo de U\$ 500 milhões);
 - Erro em uma conversão de valores de tipos de dados
- Sistema de Ambulância de Londres – 1992
 - Morte de pessoas que não foram socorridas em tempo.
 - Problema de Gerência de Software:
 - Responsáveis contrataram uma empresa desconhecida cujo valor cobrado era menor que os cobrados pelas empresas de renome.
 - Colocaram o sistema no ar sem os devidos testes.
 - Não foi feita uma migração correta do sistema antigo para o novo.

7

Características de um Software

- Software é desenvolvido e não produzido no sentido clássico (industrial)
 - custo de software é na engenharia e não na reprodução
- Software não “gasta”
 - custos em sua manutenção
- Software precisa se adaptar a novas tecnologias (“crise do software”)

8

Crise de software

- Refere-se a um conjunto de problemas encontrados no desenvolvimento de software:
 1. **As estimativas de prazo e de custo frequentemente são imprecisas.**
 - “Não dedicamos tempo para coletar dados sobre o processo de desenvolvimento de software”
 - “Sem nenhuma indicação sólida de produtividade, não podemos avaliar com precisão a eficácia de novas ferramentas, métodos ou padrões”.

9

Crise de software

2. **A produtividade das pessoas da área de software não tem acompanhado a demanda por seus serviços.**
 - “Os projetos de desenvolvimento de software normalmente são efetuados apenas com um vago indício das exigências do cliente”.

10

Crise de software

3. **A qualidade de software às vezes é menos que adequada**
 - A insatisfação do cliente com o sistema “concluído” ocorre muito frequentemente
 - Só aí começam a surgir conceitos quantitativos sólidos de garantia de qualidade de software.
 - Geralmente a comunicação entre o cliente e o desenvolvedor de Software é muito fraca.

11

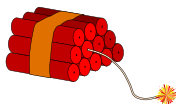
Crise de software

4. **O software existente é muito difícil de manter.**
 - A tarefa de manutenção devora o orçamento destinado ao software.
 - A facilidade de manutenção não foi enfatizada como um critério importante.

12

Crise de software

- Estimativas de prazo e de custo ↑
- Produtividade das pessoas ↓
- Qualidade de software ↓
- Software difícil de manter ↓



13

Crise de software: Causas dos problemas

- Introdução de erros no processo
 - Má especificação
 - Mau projeto
 - Má implementação
 - Testes incompletos ou mal feitos
- Problemas na comunicação homem-máquina
- Problemas na gerência
 - Falta de treinamento em novas técnicas de desenvolvimento
 - O processo (regras de negócio) está evoluindo muito rapidamente em relação ao aprendizado. É necessário reciclar.

14

Crise de software: Causas dos problemas

- O próprio caráter do Software
 - O software é um elemento de sistema lógico e não físico.
 - O sucesso é medido pela qualidade de uma **única entidade** e não pela qualidade de muitas entidades manufaturadas.
 - "O software não se desgasta, mas se deteriora."

15

Crise de software: Causas dos problemas

- Falhas das pessoas responsáveis pelo desenvolvimento de software
 - Gerentes sem nenhuma experiência em software
 - Os profissionais da área de software têm recebido pouco treinamento formal em novas técnicas para o desenvolvimento de software
 - Resistência a mudanças.

16

Crise de software: Problemas

- Complexidade
 - Especificar sistemas é uma atividade bastante complexa.
 - Não se trata apenas de fazer uns "programinhas".
- Insatisfação dos usuários
 - Usuários sentem-se frustrados com sistemas difíceis de serem operados e/ou cujo desenvolvimento se prolonga por vários anos.
 - O usuário precisa de sistemas funcionando de acordo com suas necessidades.

"O valor de um sistema está em atender com precisão às necessidades de seus usuários."

17

Crise de software: Problemas

- Produtividade
 - Costuma estar quase sempre aquém do desejado.
 - Frequentemente, a alocação de recursos e atividades são desbalanceadas.
 - Algumas questões recebem consideração demasiada, enquanto outras são insuficientemente consideradas.
 - Custos, tempo e recursos geralmente são subestimados.

18

Crise de software: Problemas

- Confiabilidade do Sistema
 - Há diversas estatísticas que provam a pouca confiabilidade de boa parte dos sistemas.
 - Depende do uso de métodos que possam garantir uma boa qualidade do produto construído.
 - Não basta que o sistema produza resultados solicitados pelo usuário... É necessário ter um desempenho adequado.

"Não é suficiente que o sistema seja eficiente.
É necessário ainda que ele seja eficaz."

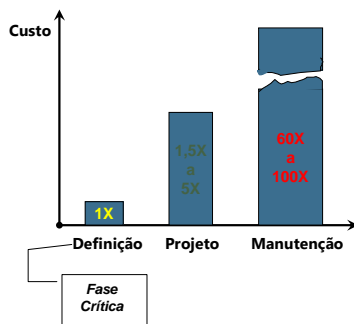
19

Crise de software: Problemas

- Manutenibilidade
 - Facilidade de se modificar um sistema para adaptar-se a circunstâncias novas, inexistentes quando foi implantado.
 - Sistemas recentemente implantados são substituídos por novos, devido ao alto custo para sua manutenção.

20

Crise de software: Problemas



21

Crise de software: Problemas

- Porque leva tanto **tempo** para terminar programas e sistemas?
- Porque os **custos** são tão altos?
- Porque temos dificuldade de **medir** o progresso do desenvolvimento do software?
- Porque não conseguimos detectar todos os **erros** antes de entregar o software aos nossos clientes?
- Porque os usuários estão sempre **insatisfeitos** com o software entregue?

22

Crise de software: Causas dos problemas

- Mitos de Software
 - Propagaram desinformação e confusão
 - Tipos de Mitos de Softwares:
 - Administrativos (ou de Gerência);
 - Profissional (Desenvolvedor/Programador);
 - Cliente.

23

Mitos da Gerência...



24

Mitos do Desenvolvedor...



25

Mitos do Cliente...



26

A Engenharia de Software...

- Abrange um conjunto de três elementos fundamentais:
 1. **Métodos;**
 2. **Ferramentas;** e
 3. **Procedimentos.**

27

Métodos

- Proporcionam os detalhes de **como fazer** para construir **o software**, tais como:
 - Planejamento e estimativa de projeto
 - Análise de requisitos de software e de sistemas
 - Projeto da estrutura de dados
 - Algoritmo e processos do sistema
 - Codificação
 - Teste
 - Manutenção

28

Ferramentas

- Dão suporte automatizado aos métodos.
- Atualmente existem ferramentas para sustentar cada um dos métodos.

29

Procedimentos

- Constituem uma ligação entre os métodos e ferramentas, ou seja:
 - Sequência em que os métodos serão aplicados
 - Produtos que se exige que sejam entregues
 - Controles que ajudam a assegurar a qualidade e coordenar as alterações
 - Marcos de referência que possibilitam administrar o progresso do software.

30

Concluindo...

- A Engenharia de Software consiste:
 - Num conjunto de etapas que envolve **métodos, ferramentas e procedimentos**.
 - Essas etapas são conhecidas como componentes de **Ciclos de Vida de Software** ou **Modelos de Processos de Desenvolvimento de Software**.
 - Alguns ciclos de vida mais conhecidos são: **Ciclo de Vida Clássico, Prototipação, Modelo Espiral, Modelo Evolutivo, Modelo Incremental, Orientado a Reúso e Rational Unified Process (RUP)**.

31

Dúvidas?

Prof. Me. Fernando Roberto Proença

fernando.proenca@uemg.br



32