

Engenharia de Software

Introdução

Prof. Sanderson Macedo

Aula 01

Porque precisamos de Engenharia
de software?

Vamos ver alguns casos

London Ambulance System

Despacho de ambulâncias
em Londres, 1992

London Ambulance System

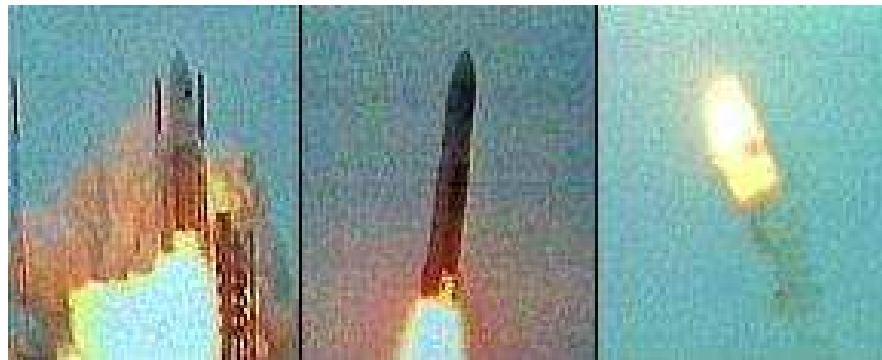
- Morte de pessoas que não foram socorridas em tempo.
- Problema de Gerência de Software;
- Responsáveis contrataram uma empresa desconhecida cujo valor cobrado era menor que os cobrados pelas empresas de renome.

London Ambulance System

- Colocaram o sistema no ar sem os devidos testes.
- Não foi feita uma migração correta do sistema antigo para o novo.

Ariane 5

Foguete de lançamento, 1996



Ariane 5

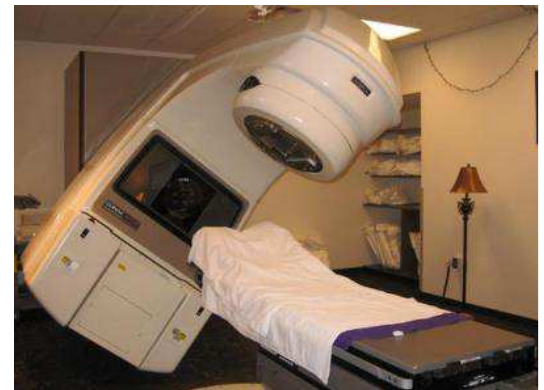
- O foguete explodiu 40 segundos após a sua primeira decolagem.
- Prejuízo de U\$ 500 milhões
- Foi aproveitado um pacote de software de navegação do Ariane 4 que não tinha erros.

Ariane 5

- No Ariane 5, o tal módulo só deveria ter sido utilizado antes do lançamento, entretanto foi utilizado durante o voo.
- **Falta de Teste e de Reengenharia**

Therac-25

Máquina de radioterapia com
software controlador 1985



Therac-25

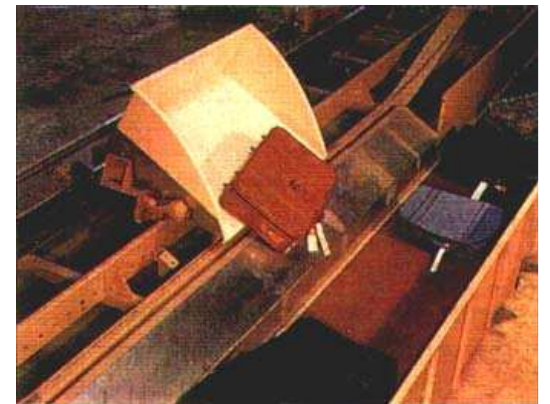
- O controle de segurança feito pelo hardware em máquinas anteriores foi removido e passou a ser feito pelo software.
- O software falhou na tarefa de, ao mesmo tempo, manter serviços essenciais: o feixe de elétrons e o dispositivo que controla a concentração do feixe em níveis seguros;

Therac-25

- Diversas mortes em decorrência de queimaduras.
- O programador não tinha experiência em programação concorrente.

Denver International Airport

Sistema de triagem/controlado de
bagagem



Denver International Airport

- Atrasou a inauguração do aeroporto. Custo do sistema: US\$ 193 milhões
- Inauguração estava prevista para Out/1993.
- Em Julho/1994 o sistema ainda não estava funcionando e causava prejuízos de US\$ 1,1 milhão/dia.

Denver International Airport

- No começo de 1995 um controle MANUAL de bagagem foi instalado para que o aeroporto pudesse ser inaugurado (com atraso de mais de um ano)
- Problema de planejamento e gerenciamento.

USS Vincennes

Derrubada de avião do Iran A320



USS Vincennes

- Desastre – USS Vincennes derrubou um A320 em 1988
- Motivo – BUG! – confundiu um A320 com um F-14.
- Prejuízo – \$107 milhões de dólares
- **290 mortes**

GPS Waze

Erro de localização por GPS



GPS Waze

- O casal ia do Rio para Niterói e colocou o endereço de destino no aplicativo de navegação por GPS Waze. Ago/2015.
- Foram levados à Rua Quintino Bocaiúva, dentro da favela do Caramujo.

GPS Waze

- O casal teria se deparado com criminosos armados, e o automóvel foi alvo de disparos.
- O motorista tentou fugir, acabou entrando em uma rua saída e foi alvo novamente de diversos tiros antes de conseguir deixar a região;

GPS Waze

- A mulher morreu;



O software está em crise?

A crise do software

- O termo “crise do software” surgiu no fim da década de 1960 –início dos anos 1970;
- Alto custo de manutenção de sistemas;
- Alto custo de novos projetos que falhavam;

O que é falhar?

O que é falhar?

- Falhar = Não cumprimento de prazos;
- Falhar = Orçamento estourado;
- Falhar = Não satisfação dos requisitos;
- Falhar = Produto de baixa qualidade;
- Falhar = Produtos não gerenciáveis e difíceis de manter e evoluir;

“A maior causa da crise do software é que as máquinas tornaram-se várias ordens de magnitude mais potentes!

Enquanto não havia máquinas, programar não era um problema;
Quando só havia computadores fracos, o problema pequeno.

Agora que temos computadores gigantescos, programar tornou-se um problema gigantesco.”

Dijkstra, 1971.



*A Crise é do software ou
a crise é dos
desenvolvedores de
software?*

A crise dos desenvolvedores

- Parece existir uma desorientação em relação sobre como planejar e conduzir o processo de desenvolvimento de software.

A crise dos desenvolvedores

- Muitos desenvolvedores concordam que não utilizam um processo adequado e que deveriam investir em algum.
- E assim segue a indústria de software, década após década.

A engenharia de software

Em meio a crise do software a Engenharia de Software surgiu propondo abordagens padronizadas para o desenvolvimento de software.

O que é software?

O Software

1º instruções (programa de computador) que, quando executadas, produzem a função e o desempenho **desejado**;

2º estruturas de dados que permitem a manipulação da informação;

3º documentos que descrevem a operação e o uso dos programas.

***Áreas de conhecimento da
engenharia de software.***

Áreas de conhecimento

Segundo o [SWEBOK](#) (Corpo de Conhecimento da Engenharia de Software), versão 2004, as áreas de conhecimento da **Engenharia de Software** são:

- Requisitos de software
- Projeto de software
- Construção de software
- Teste de software

Áreas de conhecimento

- Manutenção de software
- Gerência de configuração de software
- Gerência de engenharia de software
- Processos de Engenharia de Software
- Ferramentas e Métodos de Engenharia de Software
- Qualidade de software

Os Mitos do Software

Os mitos do software

Alguns mitos do software definidos são:

- Mitos administrativos;
- Mitos do cliente;
- Mitos do profissional;

Mito Administrativo

1. “A empresa deve produzir com qualidade, pois tem ferramentas e computadores de última geração.” (Seu futuro chefe, 2016)

- Computadores e ferramentas boas são necessários. Mas comprar uma ferramenta não lhe fará instantaneamente em um arquiteto.

Mito Administrativo

2. “Se o projeto estiver atrasado, sempre é possível adicionar mais programadores para cumprir o cronograma.” (Seu futuro chefe, 2016)

- Desenvolvimento de software é algo complexo.
- O simples ato de adicionar pessoas ao time pode gerar mais atrasos.
- Imagine construir um programa de 20 mil linhas de código com apenas **um minuto de prazo**.
- Bastaria contratar 20 mil programadores.

Mito Administrativo

2. “Se o projeto estiver atrasado, sempre é possível adicionar mais programadores para cumprir o cronograma.” (Seu futuro chefe, 2016)

- Desenvolvimento de software é algo complexo.
- O simples ato de adicionar pessoas ao time pode gerar mais atrasos.
- Imagine construir um programa de 20 mil linhas de código com apenas **um minuto de prazo**.
- Bastaria contratar 20 mil programadores.

Os Mitos do cliente

Mito do cliente

1. “Uma declaração geral de objetivos é suficiente para iniciar a fase de programação. Os detalhes podem ser adicionados depois.” (Seu futuro cliente, 2015)

- Esperar que a especificação esteja 100% completa e correta é utópico. No entanto não se deve conformar-se.
- Poucos detalhes significa retrabalho.
- Técnicas mais sofisticadas de análise de requisitos e uma equipe bem treinada poderão ajudar a construir especificações melhores em menos tempo.

Mito do cliente

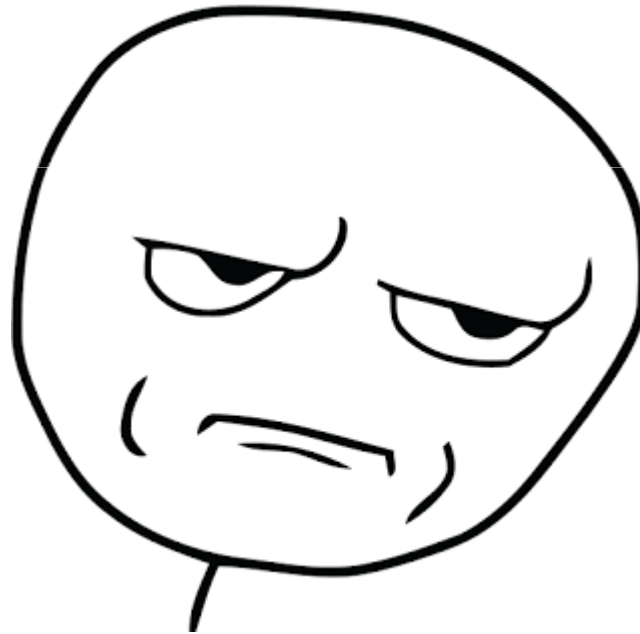
2. “Os requisitos mudam com frequência, mas sempre é possível acomodá-los, pois o software é flexível. Código é fácil de mudar!” (Seu futuro cliente, 2015)

- “Software não é um edifício, mas é difícil alterá-lo. Manutenção implica esforço e custo (tempo e recursos)”.

Mito do cliente

3. *“Eu sei do que preciso.”*
(*Seu futuro cliente*, 2016)

- Sério???



Os Mitos do programador

Mito do programador

*1. “Assim que o programa for colocado em operação, nosso trabalho terminou.”
(VOCÊ, 2016)*

- Alguns estudos apontam que mais da metade do esforço aplicado com um sistema de software ocorre após a sua implantação.

Mito do programador

2. “Enquanto o programa não estiver funcionando, não será possível avaliar sua qualidade”

- O programa é apenas um dos artefatos produzidos (sim, certamente o mais importante).
- A qualidade dos requisitos, modelos, casos de uso, protótipos, fazem parte do processo de desenvolvimento e influenciam diretamente no produto final.

Mito do programador

3. “Se eu esquecer de algo, posso consertar depois.”

- Quanto mais complexo fica o sistema, mais custosa fica a manutenção.
- Nota mental: Conserte agora!