



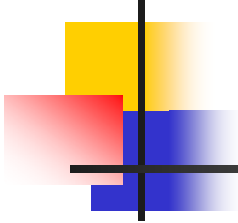
Engenharia de Software

Alessandreia Marta de Oliveira
alessandreia@gmail.com



Mas ...

- O que é Engenharia de Software?
 - “É a aplicação de uma abordagem sistemática, disciplinada e quantificável ao desenvolvimento, operação e manutenção de software” (IEEE Std 610.12, 1990)



Mas eu já sei modelar e programar!

- Por que preciso de Engenharia de Software?
 - Modelar e programar são parte importante do processo de ES, mas não são tudo!
- Precisamos também saber...
 - como estimar um projeto (tamanho, custo, cronograma)
 - como monitorar o andamento de um projeto
 - como testar o software
 - como controlar a evolução do software, etc



Programas da faculdade

- Requisitos estáveis e bem definidos
- Escopo pequeno (1-10 KLOCS)
- Prazos razoáveis
- Equipes pequenas
- Mão de obra gratuita
- Não entra em produção
- Ausência de cliente
- Ausência de manutenção

Programas do “mundo real”

- Fazer software no “mundo real” deve considerar fatores como:

- Escopo
- Custo
- Prazo
- Qualidade



- Em função do tamanho do software, esses fatores se tornam difíceis de garantir!

Cenário 1: Agenda Pessoal

- Objetivo
 - Guardar o nome e o aniversário de até 50 pessoas
- Quanto custa para fazer?
- Quanto tempo vai levar para ficar pronto?
- Qual a consequência no caso de defeito?



Cenário 2: Boeing 777

- Objetivo
 - Controlar todo o hardware do Boeing 777
- Quanto custa para fazer?
- Quanto tempo vai levar para ficar pronto?
- Qual a consequência no caso de defeito?



Cenário 2: Boeing 777

- Tamanho
 - Mais de 4 milhões de linhas de código
 - Linguagem dominante (>99%): Ada
- Documentação
 - De 100 a 10.000 páginas por sub-sistema
 - Total de 79 sub-sistemas integrados
- Duração
 - 4,5 anos de desenvolvimento
- Ampla utilização de ES
- Em operação desde 1995
 - Zero acidentes graves até 2006

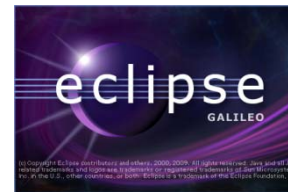


<http://www.stsc.hill.af.mil/crosstalk/1996/01/Boein777.asp>

<http://www.boeing.com/news/techissues/pdf/statsum.pdf>

Outros cenários extremos...

- Toyota Lexus LS460: > 7 MLOCs
- Eclipse Galileo: 24MLOCs
- Windows XP: 40 MLOCs
 - 1800 desenvolvedores
 - 2200 testadores
- SAP: 250 MLOCs
- Debian GNU/Linux 4: 283 MLOCs
 - 1000 desenvolvedores





Mas “quão grande” é isso?

- Uma folha A4 tem em torno de 50 linhas
- Uma pilha de 1000 folhas A4 tem em torno de 10 centímetros de altura
- Assim, 1.000.000 de LOCs
 - Caso impresso, seria uma pilha de 2 m de altura!
- Caso todo o código do Debian GNU/Linux fosse impresso
 - Teria a altura de um prédio de 188 andares!!!

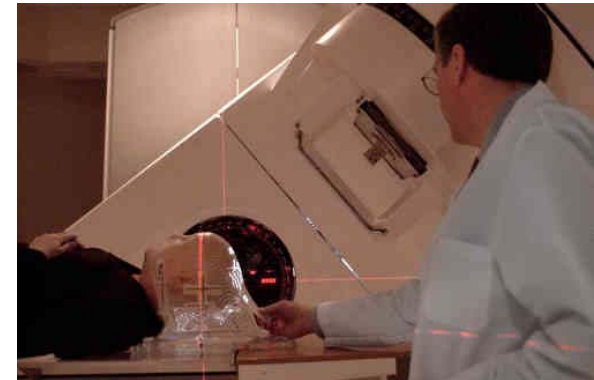


Mas fazer software não é arte?

- Parte arte, parte engenharia...
 - Se o cantor/ator/pintor errar, a audiência fica chateada
 - Se o engenheiro civil errar, o prédio pode cair
 - Se o médico errar, o paciente pode morrer
- Se o desenvolvedor de software errar, o que pode acontecer?

Caso real 1: Therac-25

- Máquina de radioterapia controlada por computador
- Problema:
 - Doses indevidas de radiação emitidas
- Causa:
 - Interface com usuário inapropriada
 - Documentação deficiente
 - Software reutilizado sem ser adaptado para o novo hardware
 - Software de sensores de falha com defeito
- Conseqüências
 - Ao menos 5 mortes entre 1985 e 1987



<http://sunnyday.mit.edu/papers/therac.pdf>

Caso real 2: Ariane 5



- Foguete lançador de satélites
- Problema:
 - O foguete se auto-destruiu 40 segundos após o lançamento
- Causa:
 - Sw reutilizado sem adaptação para o novo hw
 - Ausência de testes em solo deste software
 - Defeito apresentado em voo
- Conseqüências
 - Prejuízo de mais de US\$ 370.000.000,00 em 1996

Dowson, Mark. 1997. The Ariane 5 software failure.
SIGSOFT Softw. Eng. Notes 22, no. 2.