

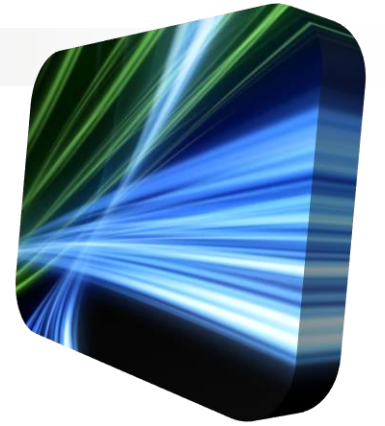


Engenharia de Software

Introdução

José Osvano da Silva, PMP

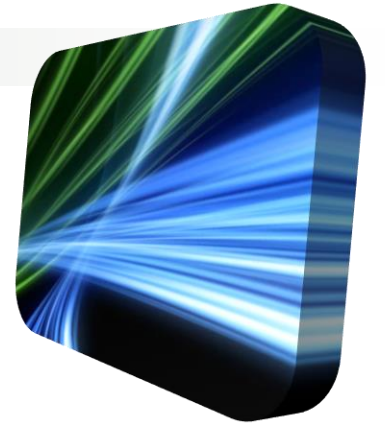
Sumário



- O que é Engenharia de Software?
- Programa de Faculdade
- Programa do “mundo real”
- Cenários
- Elementos da Engenharia de Software
- Processo de Qualidade
- Exercício de Fixação
- Referências

O que é Engenharia de Software?

- “Engenharia de Software é a aplicação de uma abordagem **sistemática**, **disciplinada** e **quantificável** ao desenvolvimento, operação e manutenção de software”.



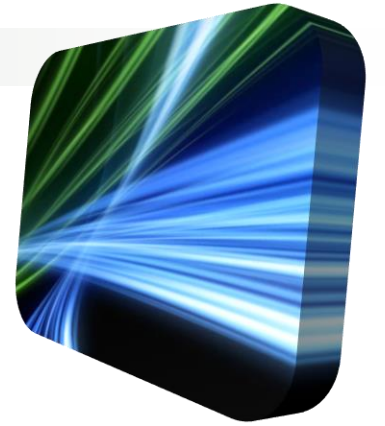
IEEE Std 610.12 (1990)

Mas eu já sei modelar e programar!



- **Por que preciso de Engenharia de Software?**
 - **Modelar e programar** são parte **importante** do processo de Engenharia de Software, **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 de 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

* Acrônimo para Kilo Lines of Code (em português, mil linhas de código)
Engenharia de Software

Programas do “mundo real”



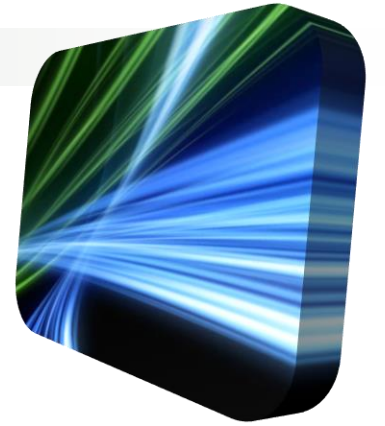
- Fazer software no “mundo real” deve considerar fatores como:

- ☐ 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



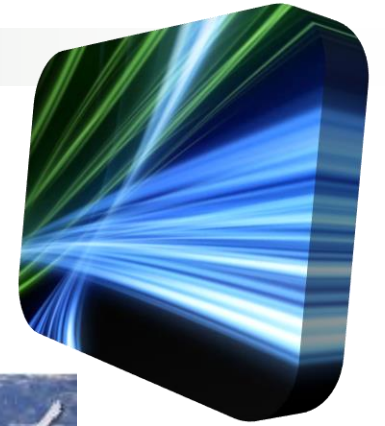
- Objetivo
 - Controlar todo o hardware do Boeing 777



- Quanto custa para fazer?
- Quanto tempo vai levar para ficar pronto?
- Qual a consequência no caso de defeito?

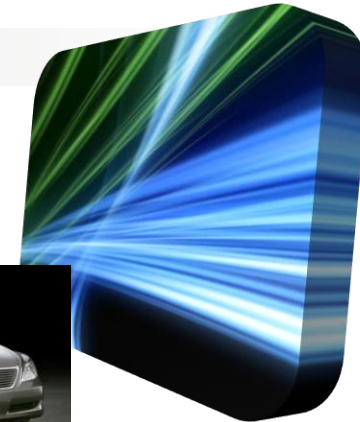
Cenário 2: Boeing 777

- 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 Engenharia de Software
- Em operação desde 1995
 - Zero acidentes graves até 2013
 - <http://www.stsc.hill.af.mil/crosstalk/1996/01/Boein777.asp>
 - <http://www.boeing.com/news/techissues/pdf/statsum.pdf>

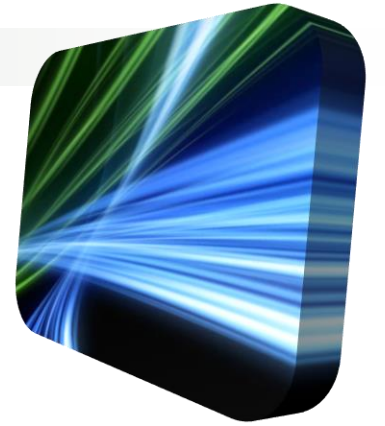


Outros Cenários Extremos...

- 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



Mais quanto grande é isso?



- Assuma que uma folha A4 tem em torno de 50 linhas.
- Assuma que 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 metros de altura!
- Caso todo o código do Debian GNU/Linux fosse impresso, teria a altura de um prédio de 188 andares!!!

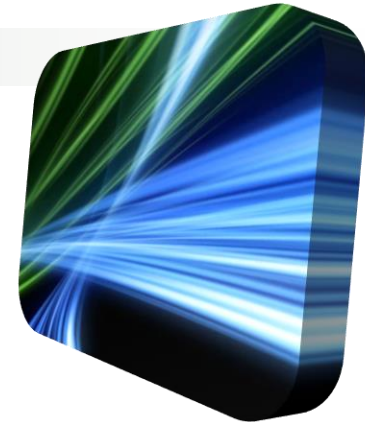
Mas fazer software não é arte?



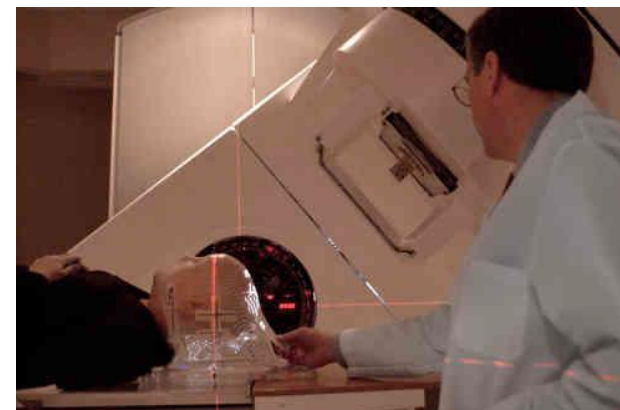
- Parte arte, parte engenharia...
 - Se o cantor/ator/pintor errar, a audiência fica chateada
 - Se o engenheiro civil errar o prédio pode cair
 - Se o médico errar o paciente pode morrer

- Se o desenvolvedor de software errar, o que pode acontecer?

Caso real 1: Therac-25?

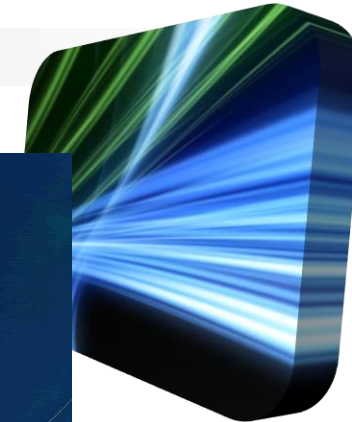


- Máquina de radioterapia controlada por computador
- Problema:
 - Doses indevidas de radiação emitidas
- Causa:
 - Interface com usuário inapropriada
 - Documentação deficiente
 - Software reutilizado sem ser adaptado para o novo hardware
 - Software de sensores de falha com defeito
- Conseqüências
 - 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:
 - Software reutilizado sem ser adaptado para o novo hardware
 - Ausência de testes em solo deste software
 - Defeito apresentado em voo
- Conseqüências
 - Prejuízo de mais de US\$ 370.000.000,00 em 1996



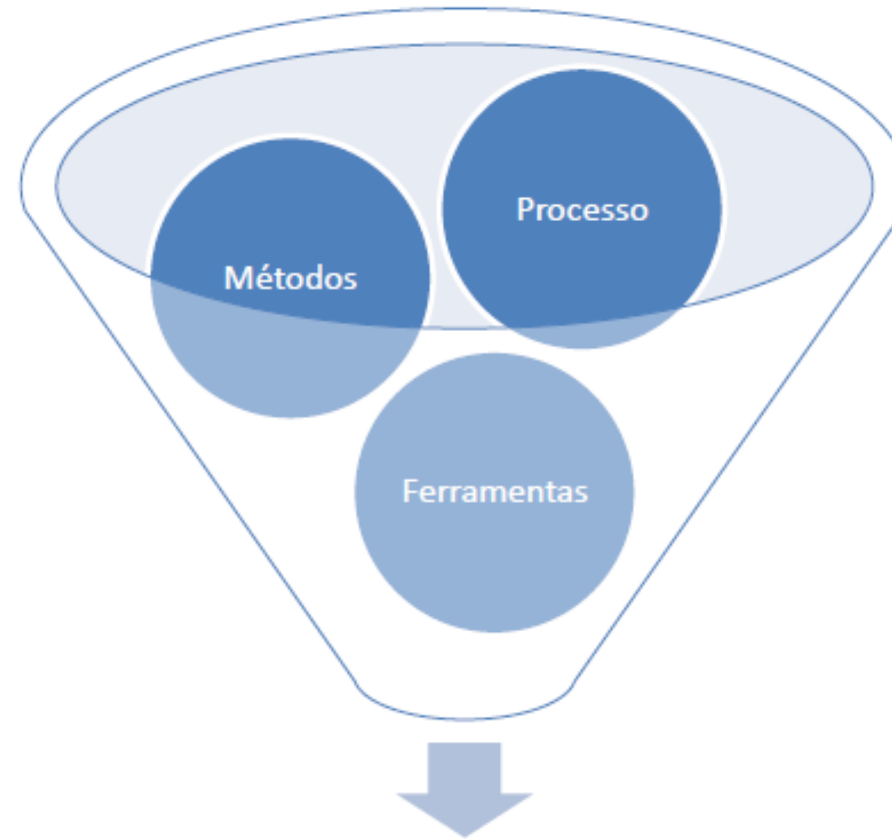
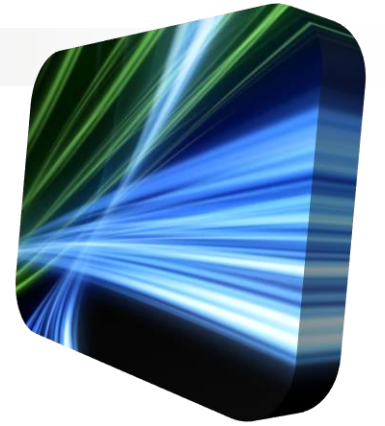
- Dowson, Mark. 1997. The Ariane 5 software failure. *SIGSOFT Softw. Eng. Notes* 22, no. 2.

Motivação extra para estudar?

- Diversos concursos e oportunidades de emprego exigem conhecimento de Engenharia de Software
- Alguns exemplos:



Elementos da Engenharia de Software



Engenharia de Software

Elementos da Engenharia de Software



■ Processo

- Define os passos gerais para o desenvolvimento e manutenção do software
- Serve como uma estrutura de encadeamento de métodos e ferramentas

■ Métodos

- São os “how to’s” de como fazer um passo específico do processo

■ Ferramentas

- Automatizam o processo e os métodos

Elementos da Engenharia de Software



- Cuidado com o “desenvolvimento guiado por ferramentas”
 - É importante usar a ferramenta certa para o problema
 - O problema não deve ser adaptado para a ferramenta disponível



“Para quem tem um martelo, tudo parece prego”

Elementos da Engenharia de Software



1. Coloque em uma panela funda o leite condensado, a margarina e o chocolate em pó.
2. Cozinhe [no fogão] em fogo médio e mexa sem parar com uma colher de pau.
3. Cozinhe até que o brigadeiro comece a desgrudar da panela.
4. Deixe esfriar bem, então unte as mãos com margarina, faça as bolinhas e envolva-as em chocolate granulado.

**O que é
processo,
método ou
ferramenta?**

Elementos da Engenharia de Software



1. Coloque em uma panela funda o leite condensado, a margarina e o chocolate em pó.
2. Cozinhe [no fogão] em fogo médio e mexa sem parar com uma colher de pau.
3. Cozinhe até que o brigadeiro comece a desgrudar da panela.
4. Deixe esfriar bem, então unte as mãos com margarina, faça as bolinhas e envolva-as em chocolate granulado.



método

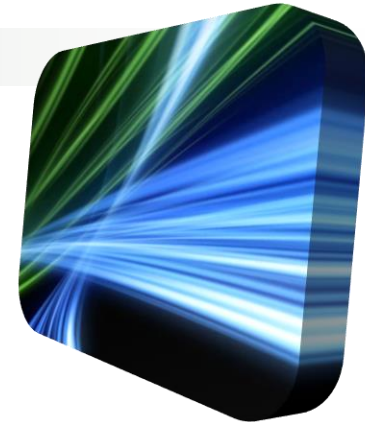


ferramenta

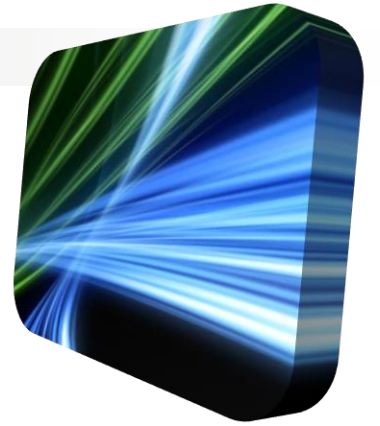
processo

O Supermercado de ES

- ES fornece um conjunto de métodos para produzir software de qualidade
- Pense como em um supermercado...
 - Em função do problema, se escolhe o processo, os métodos e as ferramentas
- Cuidado
 - Menos do que o necessário pode levar a desordem
 - Mais do que o necessário pode emperrar o projeto



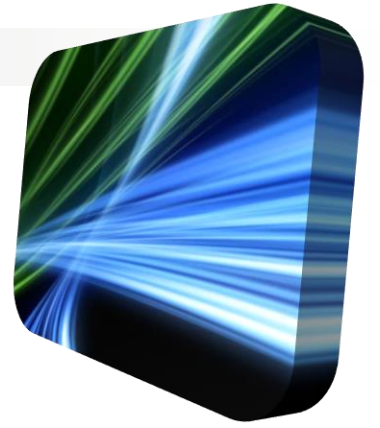
Processos implícitos x explícitos



- Lembrem-se: Processos sempre existem, seja de forma implícita ou explícita!
 - Processos implícitos são difíceis de serem seguidos, em especial por novatos
 - Processos explícitos estabelecem as regras de forma clara



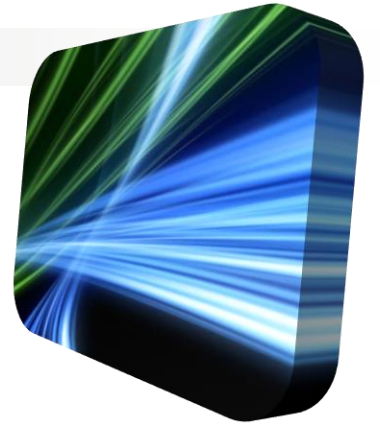
Processo de Qualidade



- Última palavra para medir a qualidade de um processo:
Satisfação do Cliente

- Outros indicadores importantes
 - Qualidade dos produtos gerados
 - Custo real do projeto
 - Duração real do projeto

Exercício de Fixação



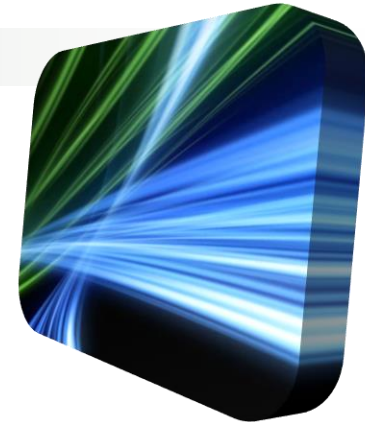
- Como Criar um App com Zero Experiência?
 - Identifique o que Método, Processo e Ferramenta;
 - Envie o resultado no portal;
 - Avise que terminou no chat da aula.

Referências



- Craig Larman, 2007, “**Utilizando UML e Padrões**”, 3ª ed.
- PRESSMAN, Roger; Maxim, Bruce. **Engenharia de software**. Porto Alegre: AMGH, 2016 (e-book).
- SOMMERVILLE, Ian. **Engenharia de software**. Tradução de Kalinka Oliveira e Ivan Bosnic. 9.ed. São Paulo: Pearson, 2011.

Dúvidas



José Osvano da Silva
joseosvano@unipac.br