

Engenharia de Software 1

Viviane Torres da Silva
viviane.silva@ic.uff.br

<http://www.ic.uff.br/~viviane.silva/2010.1/es1>

A Engenharia de Software

O que é Engenharia de Software?

- “Engenharia de Software é a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, operação e manutenção de software” IEEE Std 610.12 (1990)

➤ Por que preciso de Engenharia de Software?

- **Programação** é parte **importante** do processo de Engenharia de Software, **mas não é tudo!**

➤ Precisamos também saber...

- **o que** programar,
- **como** programar,
- se o que foi programado está **certo**,
- 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

Programas do “mundo real”

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

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

➤ Quanto tempo vai levar para ficar pronto?

- 4,5 anos de desenvolvimento

➤ Qual a consequência no caso de defeito?

- 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>

Erro da Engenharia de Software

- 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?
Todas as coisas anteriores dependendo de onde o software esteja sendo utilizado.

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 após o lançamento
- Causa:
 - Software reutilizado sem ser adaptado para o novo hardware
 - Ausência de testes em solo deste software
 - Defeito apresentado em vôo
- 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.

Escopo do Curso



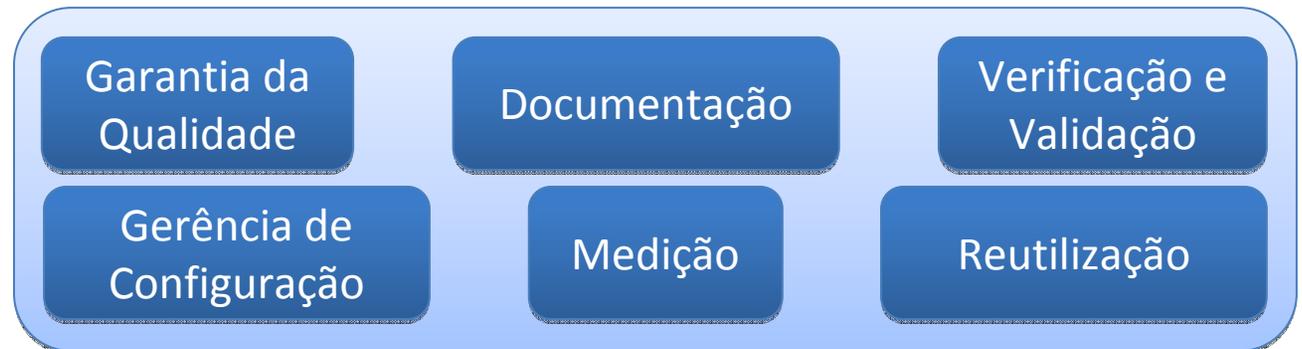
Atividades Gerenciais



Atividades de Desenvolvimento



Atividades de Apoio



Avaliação

- Prova 1 e Prova 2
 - Prova escrita
- Trabalho:
 - Apresentações do trabalho e entrega de trabalho
- Participação em sala
 - Perguntas em sala e resolução da lista

- TrabPar = Trabalho + Participação
 - Nota do trabalho vai de 0 a 9
 - Nota de participação vai de 0 a 1
- Média = (Prova 1 + Prova 2 + TrabPar) /3
 - Aprovado: presença $\geq 75\%$ e Média $\geq 6,0$
 - VS: presença $\geq 75\%$ e $4,0 \leq$ Média $< 6,0$
 - Reprovado: presença $< 75\%$ ou Média $< 4,0$
- Atendimento:
 - Enviar email para agendar

Trabalho

➤ Grupo de 3 participantes

- Todos os participantes devem fazer e apresentar o trabalho
- Definir o grupo na primeira semana de aula (não pode ser alterado)
- Enviar por email (assunto : ESI) com o nome dos participantes
- Projeto + implementação

➤ Objetivo:

- Aplicar as técnicas estudadas em sala

➤ Três apresentações serão feitas durante o curso:

1. Apresentação do tema do trabalho
2. Detalhamento do andamento do trabalho
3. Entregar o resultado do trabalho impresso e relatar as experiências adquiridas

** Atraso na entrega perde 1 ponto por dia

Março

- 9 Apresentação do curso
- 11 Introdução a ES
- 16 Introdução a ES
- 18 Processo Unificado
- 23 OO em Java
- 25 Princípios da Engenharia de Requisitos
- 30 Princípios da Engenharia de Requisitos

Abril

- 1 Não tem aula
- 6 Diagrama de Casos de Uso
- 8 Exercício
- 13 Apresentação Trabalho - Parte I
- 15 Apresentação Trabalho - Parte I
- 20 Diagrama de Classe
- 22 Exercício
- 27 Lista de exercício
- 29 P1

Maio

- 4 Diagrama de Sequência e Revisão de prova
- 6 Diagrama de Atividades e exercício (seqüência e atividades)
- 11 Não tem aula
- 13 Não tem aula
- 18 Diagrama de Estados
- 20 Exercício
- 25 Apresentação Trabalho - Parte II
- 27 Apresentação Trabalho - Parte II

Junho

- 1 Princípios da Programação Orientada a Objetos
- 3 Não tem aula
- 8 Fim de POO e Padrões de Projeto
- 10 Exercício
- 15 Apresentação Trabalho - Final
- 17 Apresentação Trabalho - Final
- 22 Reuso de Software: Frameworks e Linhas de Produto
- 24 Não tem aula
- 29 Lista de exercício

Julho

- 1 P2
- 6 Revisão
- 8 VS

Engenharia de Software 1

Viviane Torres da Silva
viviane.silva@ic.uff.br

<http://www.ic.uff.br/~viviane.silva/2010.1/es1>