

# Apresentação do Curso de Engenharia de Software 2 2009/2

Leonardo Gresta Paulino Murta  
leomurta@ic.uff.br

# Apresentações

- Quem sou eu?
  - Leonardo Murta
  - <http://www.ic.uff.br/~leomurta>
- Quem são vocês?
  - Nome?
  - Período?
  - Estágio? Projeto de Aplicação? Iniciação Científica?
  - O que achou de Engenharia de Software 1?
  - Expectativas para Engenharia de Software 2?

# Relembrando, 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 2?**
  - **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

# Programas do “mundo real”

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

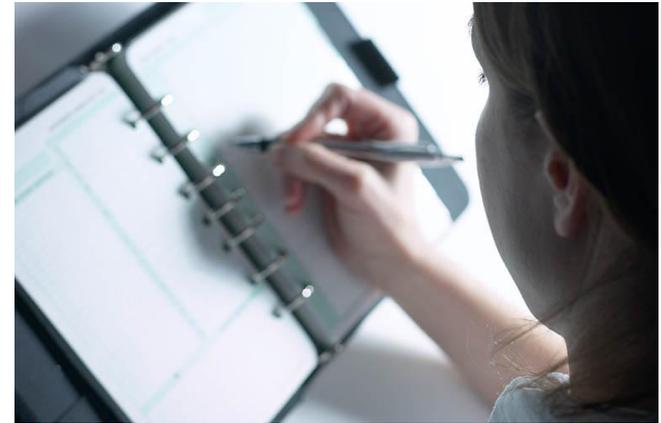
- Escopo
- Custo
- Prazo
- Qualidade



- 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 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 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:



# Como será no curso de ES2?

Leo, só apresente  
Métodos Clássicos,  
pois só eles prestam!



Leo, só apresente  
Métodos Ágeis, pois  
só eles prestam!



# Como será no curso de ES2?

- Veremos **sem preconceito** técnicas clássicas e ágeis de Engenharia de Software
- Utilizaremos o que considerarmos melhor para cada situação
- Mas... o processo base que utilizaremos é iterativo, incremental e ágil

# Escopo do curso



Atividades  
Gerenciais

Planejamento  
de Projetos

Monitoração  
e Controle

Melhoria de  
Processos

Gerência  
de Riscos



Atividades de  
Desenvolvimento

Levantamento  
de Requisitos

Análise de  
Requisitos

Projeto

Codificação



Atividades de  
Apoio

Garantia da  
Qualidade

Medição  
e Análise

Verificação,  
Validação e Testes

Gerência de  
Configuração

Reutilização

# Avaliação

$Avaliação_1 = Prova Escrita$

$Avaliação_2 = Prova Escrita$

$Avaliação_3 = \frac{(Participação + 2 \times Seminários + 3 \times Trabalho)}{6}$

$Média = \frac{(Avaliação_1 + Avaliação_2 + Avaliação_3)}{3}$

# Avaliação

- APROVADO

*Presença*  $\geq 75\%$

**E**

*Média*  $\geq 6$

- VERIFICAÇÃO SUPLEMENTAR

*Presença*  $\geq 75\%$

**E**

$4 \leq \textit{Média} < 6$

Será aprovado na VS se tirar nota maior ou igual a 6

- REPROVADO

*Presença*  $< 75\%$

**OU**

*Média*  $< 4$

# Segundo o Regulamento dos Cursos de Graduação...

- Presença
  - 75% das aulas (Art. 80, §14)
  - Nenhuma falta será abonada (Art. 80, §15)
- Segunda Chamada
  - Não será permitida a Avaliação de Aprendizagem em Caráter Excepcional (i.e., 2ª chamada)
  - Com exceção dos casos citados no Art. 87 (**congressos, competições** ou **serviço militar**), de acordo com os procedimentos do Art. 88 (**aviso na coordenação do curso com 30 dias de antecedência**).

# Grupos

- Atividades em grupo
  - Participação em aula (aprox. 15% da Avaliação<sub>3</sub>)
  - Seminários (aprox. 35% da Avaliação<sub>3</sub>)
  - Trabalho (50% da Avaliação<sub>3</sub>)
- Deve ser o mesmo durante todo o curso
  - 1 “gerente” por grupo, que delega funções aos demais membros do grupo
  - Total de 6 membros por grupo
- Definir na primeira semana de aula
  - Enviar por e-mail (assunto: ES2 - Grupo) a matrícula e o nome completo dos participantes

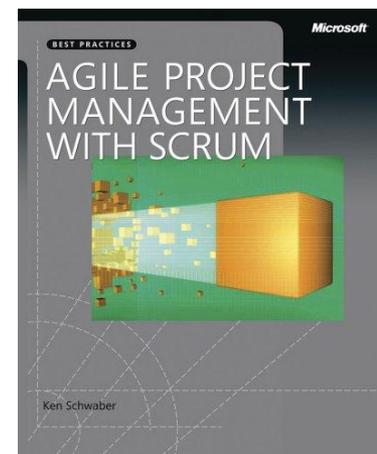
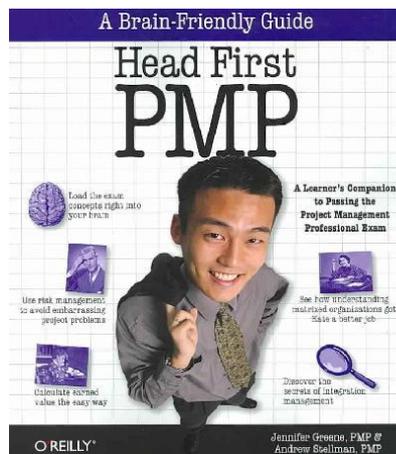
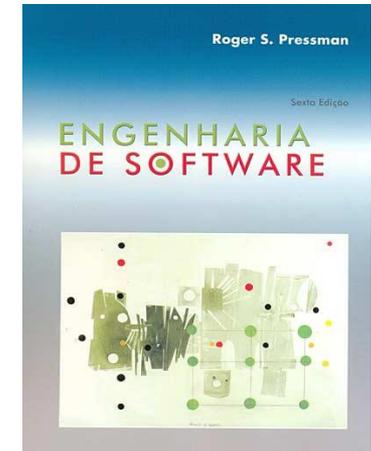
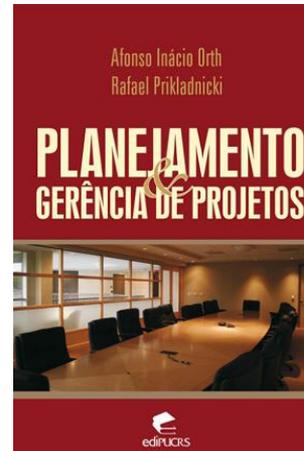
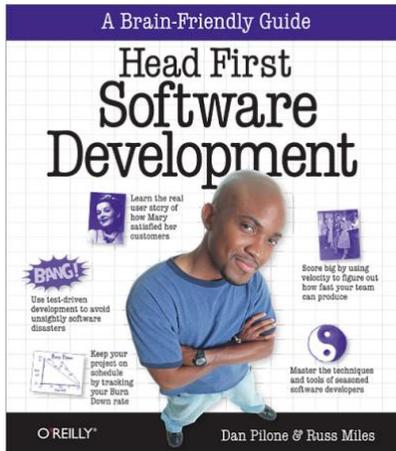
# Trabalho

- Objetivo: Fazer um **jogo de xadrez** onde seja possível jogar contra o computador
- Se enxerguem como uma pequena *software house*, e me enxerguem como cliente
- Será avaliado o produto final e como esse produto foi desenvolvido (processos e técnicas aplicados)
- Atraso na entrega do trabalho terá 1 ponto de multa por dia

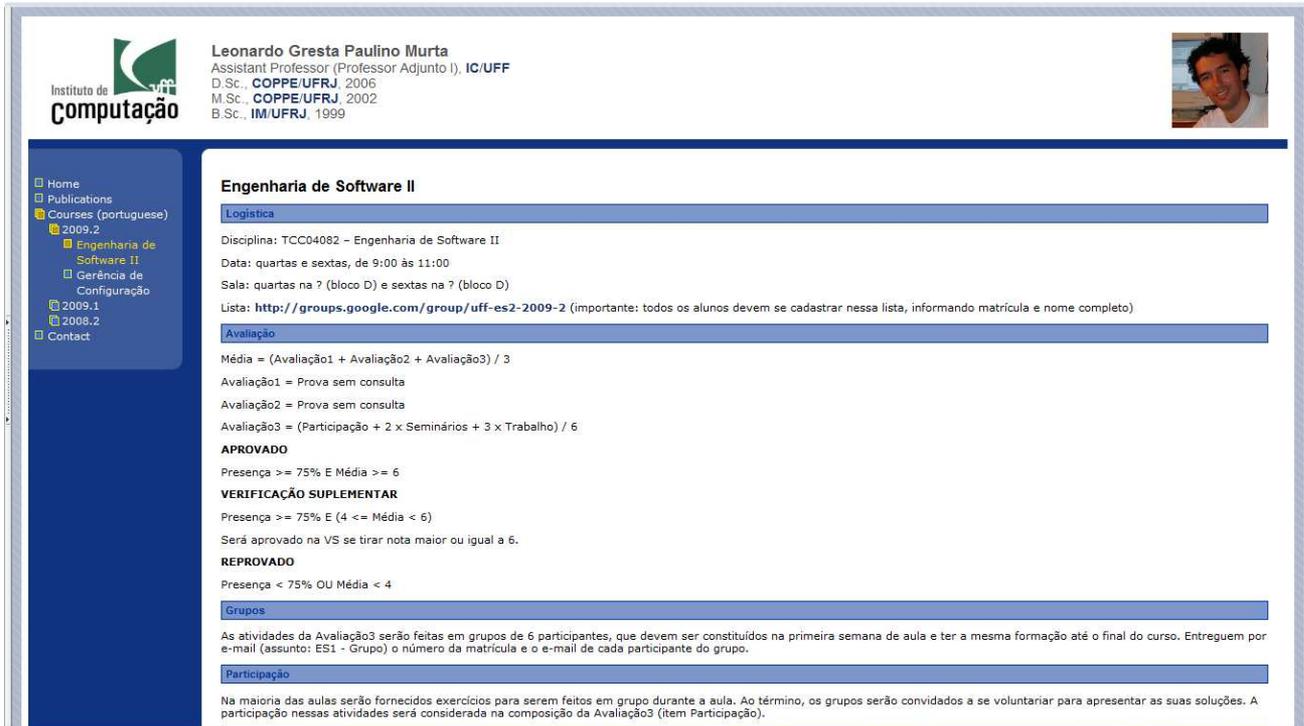
# Trabalho

- Três seminários serão feitos durante o curso
- 1º seminário
  - Papéis dos membros da equipe
  - Estimativas de esforço, custo e cronograma de desenvolvimento
  - Versão parcial do produto
- 2º seminário
  - Monitoração e controle do projeto
  - Repositório de gerência de configuração
  - Versão parcial do produto
- 3º seminário
  - Monitoração e controle do projeto
  - Casos de Teste e resultados da sua execução
  - Versão final do produto
  - Relatório final impresso

# Bibliografia básica do curso



# Página do curso



**Leonardo Gresta Paulino Murta**  
 Assistant Professor (Professor Adjunto I), IC/UFF  
 D.Sc., COPPE/UF RJ, 2006  
 M.Sc., COPPE/UF RJ, 2002  
 B.Sc., IM/UF RJ, 1999

**Engenharia de Software II**

**Logística**  
 Disciplina: TCC04082 – Engenharia de Software II  
 Data: quartas e sextas, de 9:00 às 11:00  
 Sala: quartas na ? (bloco D) e sextas na ? (bloco D)  
 Lista: <http://groups.google.com/group/uff-es2-2009-2> (importante: todos os alunos devem se cadastrar nessa lista, informando matrícula e nome completo)

**Avaliação**  
 $Média = (Avaliação1 + Avaliação2 + Avaliação3) / 3$   
 Avaliação1 = Prova sem consulta  
 Avaliação2 = Prova sem consulta  
 $Avaliação3 = (Participação + 2 \times Seminários + 3 \times Trabalho) / 5$

**APROVADO**  
 Presença  $\geq 75\%$  E Média  $\geq 6$

**VERIFICAÇÃO SUPLEMENTAR**  
 Presença  $\geq 75\%$  E  $(4 \leq Média < 6)$   
 Será aprovado na VS se tirar nota maior ou igual a 6.

**REPROVADO**  
 Presença  $< 75\%$  OU Média  $< 4$

**Grupos**  
 As atividades da Avaliação3 serão feitas em grupos de 6 participantes, que devem ser constituídos na primeira semana de aula e ter a mesma formação até o final do curso. Entreguem por e-mail (assunto: ES1 - Grupo) o número da matrícula e o e-mail de cada participante do grupo.

**Participação**  
 Na maioria das aulas serão fornecidos exercícios para serem feitos em grupo durante a aula. Ao término, os grupos serão convidados a se voluntariar para apresentar as suas soluções. A participação nessas atividades será considerada na composição da Avaliação3 (item Participação).

Leiam as regras do curso no site e tragam as dúvidas na próxima aula!!!

<http://www.ic.uff.br/~leomurta>

(dica: monitorem com <http://www.changedetection.com>)

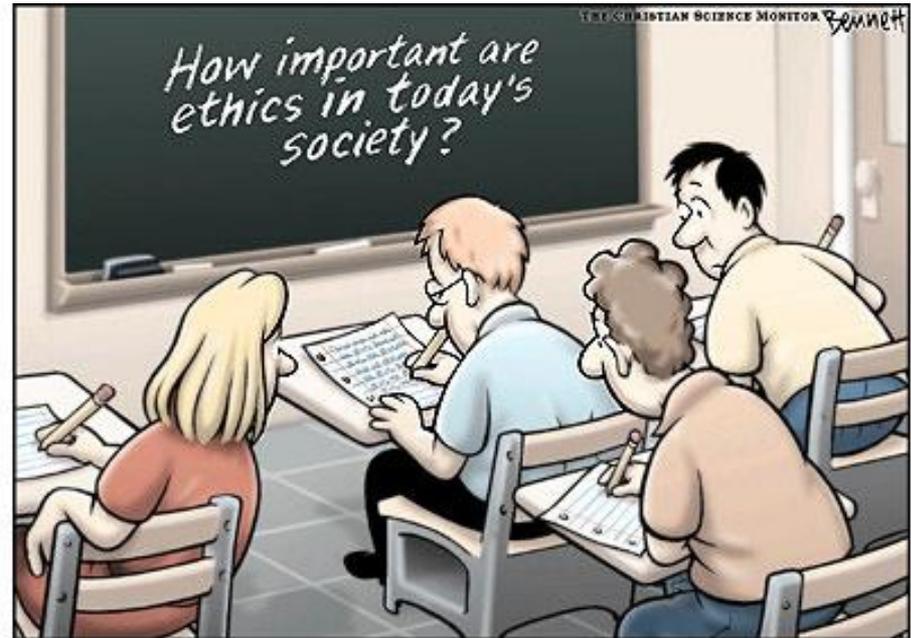
**Importante:** cadastrem-se em <http://groups.google.com/group/uff-es2-2009-2>

# Datas importantes

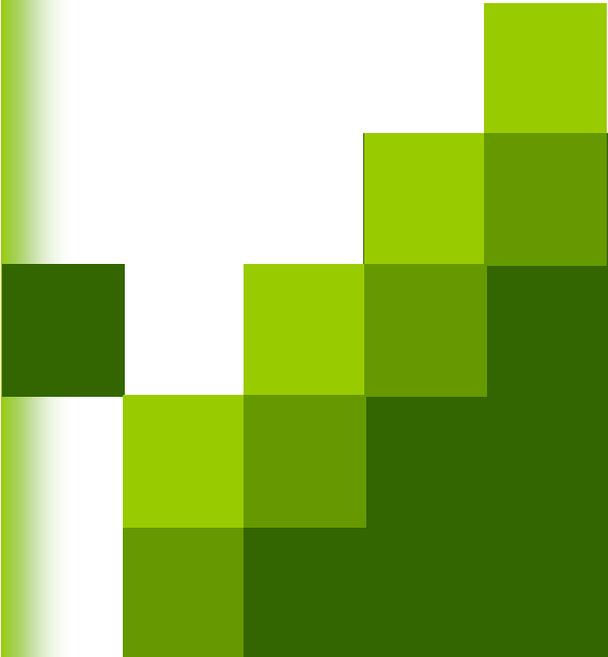
- 1ª rodada dos Seminários
  - 16/9 e 18/9
- 2ª rodada dos Seminários
  - 28/10 e 30/10
- Rodada final dos Seminários
  - 27/11 e 02/12
- Entrega do trabalho
  - 27/11
- 1ª Prova
  - 30/9
- 2ª Prova
  - 04/12
- Verificação Suplementar
  - 11/12

# Fair Play!

- Não colar ou dar cola em provas
- Não plagiar o trabalho
- Não trapacear nas leituras e listas de exercício
- Não sobrecarregar os colegas do grupo
- Não assinar presença por colegas
- Dar crédito apropriado quando usar trabalhos de terceiros



<http://www.claybennett.com/pages/ethics.html>



# Apresentação do Curso de Engenharia de Software 2 2009/2

Leonardo Gresta Paulino Murta  
leomurta@ic.uff.br