

# Apresentação da Disciplina de Gerência de Projetos e Manutenção de Software

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br

# Apresentações

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

# 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 Gerência de Projetos e Manutenção 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 controlar a evolução do software,
  - etc.

# Programas de faculdade

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

# Programas do “mundo real”

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

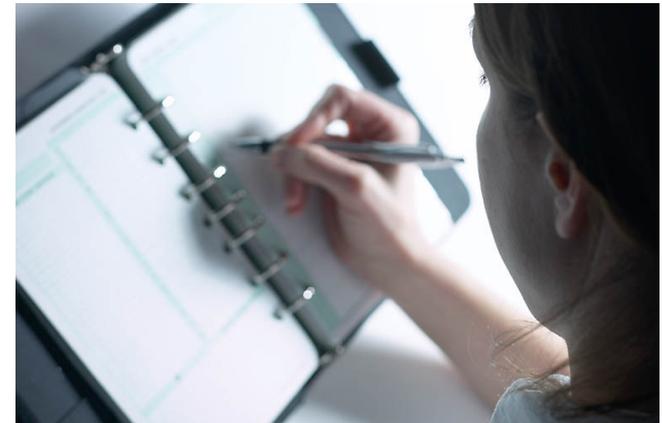
- Escopo
- Custo
- Prazo
- Qualidade



- 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>

# 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: Aeroporto de Denver

- Sistema de despacho de bagagem do aeroporto de Denver
- Problema:
  - O sistema nunca funcionou adequadamente
- Causa:
  - Arquitetura extremamente complexa
  - Mudança constante nos requisitos
  - Erros de estimativa de custo e prazo
  - Desprezo aos conselhos de especialistas
  - Intolerância a falhas
- Consequências
  - Entrega de algo muito menor do que o planejado, que foi desativado 10 anos depois
  - Atraso de 16 meses para entregar o aeroporto
  - Prejuízo de US\$ 560 milhões e custo mensal de manutenção de US\$ 1 milhão



[http://calleam.com/WTPF/?page\\_id=2086](http://calleam.com/WTPF/?page_id=2086)

# Caso real 2: 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 3: 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 deste software em solo
  - Defeito apresentado em voo
- Consequências
  - Prejuízo de mais de US\$ 370 milhões



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 o curso?

Só os Métodos  
Clássicos prestam!



Só os Métodos  
Ágeis prestam!



# Como será o curso?

- 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 será iterativo e incremental

# ES na UFF

Atividades  
Gerenciais



Planejamento  
de Projetos

Monitoramento  
e Controle

Gerência de  
Configuração

Gerência  
de Riscos

Atividades  
de Análise e  
Projeto



Levantamento  
de Requisitos

Modelagem



Arquitetura

Projeto

Reutilização

Atividades  
de Apoio



Garantia da  
Qualidade

Medição  
e Análise

Melhoria de  
Processos

Verificação,  
Validação e Testes

# Ementa da disciplina

- Planejamento de projetos
- Gerenciamento ágil
- Monitoramento e controle
- Gerência de riscos
- Gerência de configuração de software
- Engenharia reversa, refatoração e reengenharia

# Avaliação

$$Média = \frac{Prova_1 + Prova_2 + Trabalho}{3}$$

# 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*  $< 4$

# Trabalho

- Fazer um **jogo War** ou **jogo de Xadrez** onde seja possível jogar contra o computador, usando as técnicas estudadas durante o curso.
- Se enxerguem como uma pequena *software house*
  - Grupo de 6 participantes
- Será avaliado tanto o produto quanto como esse produto foi desenvolvido (processos e técnicas aplicados)

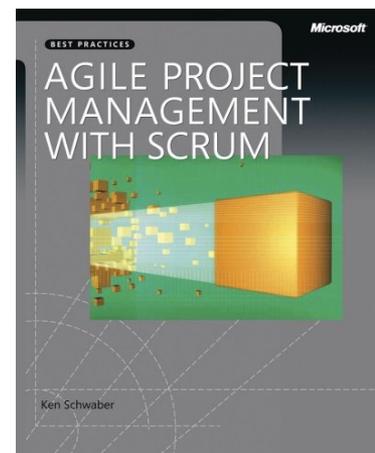
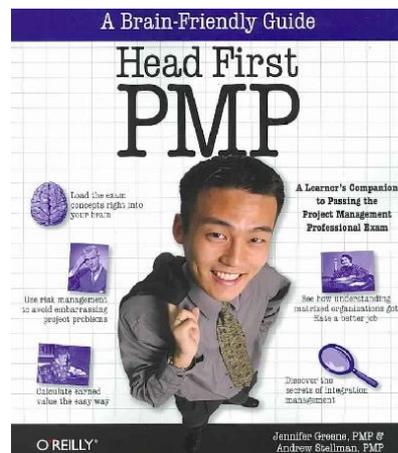
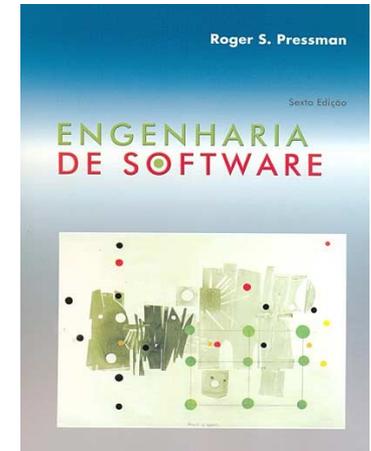
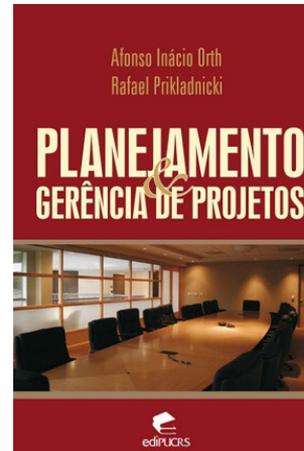
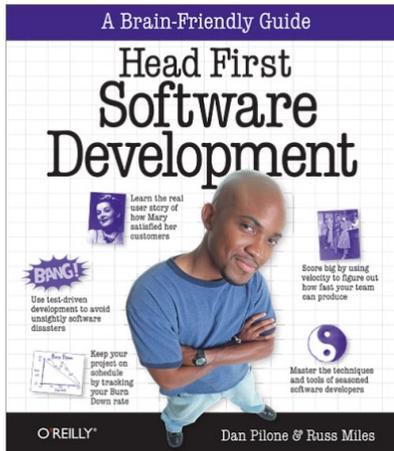
# Trabalho

- Duas apresentações serão feitas durante o curso
- 1ª apresentação e entrega
  - Escopo do produto
  - Escopo do projeto
  - Estimativas de esforço e custo
  - Orçamento
  - Cronograma
  - Análise de riscos
  - Monitoramento e controle
  - Versão parcial do produto
- 2ª apresentação e entrega
  - Ferramenta de controle de versões e de controle de modificações
  - Estratégia de ramificação
  - Conteúdo do repositório
  - Monitoramento e controle
  - Versão parcial do produto
  - Dificuldades e decisões tomadas

# Listas de Exercício

- Devem ser feitas individualmente
- Entregar no Google Classroom até a última aula antes da Prova 1 (listas 1 e 2) e da Prova 2 (listas 3 e 4)
- Valerão até 0,5 pontos na média para alunos com média entre 5,5 e 6,0, eventualmente arredondando a média para 6,0
- Não serão aceitas entregas fora do prazo

# Bibliografia do curso



# Página do curso



**Leonardo Gresta Paulino Murta**  
Associate Professor, IC/UFF  
D.Sc., COPPE/UF RJ, 2006  
M.Sc., COPPE/UF RJ, 2002  
B.Sc., IM/UF RJ, 1999



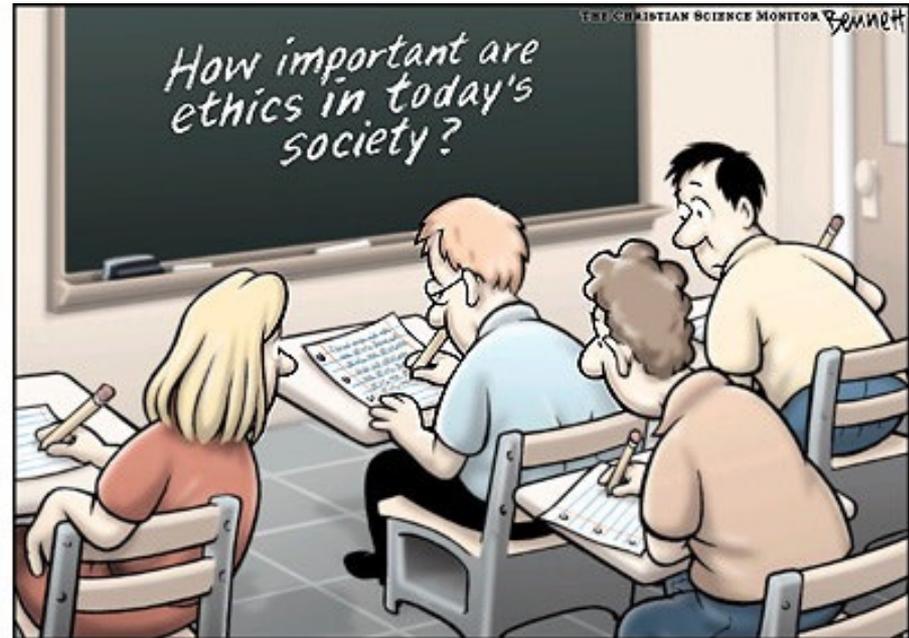
<b>Home</b>	<b>Gerência de Projetos e Manutenção de Software</b>
<b>Courses</b>	<b>Logística</b>
2022.1	Disciplina: TCC00339 - Gerência de Projetos e Manutenção de Software
Engenharia de Software II	Data: quintas, de 18:00 às 22:00
Gerência de Projetos e Manutenção de Software	Sala: 204
2021.1	<b>Ementa</b>
2020.2	<ul style="list-style-type: none"> <li>Planejamento de projetos</li> <li>Monitoramento e controle</li> <li>Gerenciamento ágil</li> <li>Gerência de riscos</li> <li>Gerência de configuração</li> <li>Engenharia reversa, refatoração e reengenharia</li> </ul>
2020.1	<b>Avaliação</b>
2019.1	<i>Média = (2 x Prova 1 + 2 x Prova 2 + Trabalho) / 5</i>
2018.2	<b>APROVADO</b>
2018.1	(Presença >= 75%) E (Média >= 6)
2017.2	<b>VERIFICAÇÃO SUPLEMENTAR</b>
2017.1	(Presença >= 75%) E (4 <= Média < 6)
2016.2	
2016.1	
2015.2	
2015.1	

<http://www.ic.uff.br/~leomurta>  
(no final da página tem o cronograma, com **datas** e **slides**)

**Importante:** toda a interação fora de sala de aula será pelo Google Classroom!

# *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>

# Exercício

- Analise com o seu grupo como irão desenvolver o trabalho
  - Qual será a duração de uma iteração?
  - O que vocês pretendem entregar em cada iteração?
  - Como e quando vocês vão se reunir para atingir esse objetivo?
  - Qual será o papel de cada membro do grupo?
  - Quais são os riscos envolvidos?
  - Quais decisões arquiteturais precisam ser tomadas (linguagem, SO, etc.)?

# Apresentação da Disciplina de Gerência de Projetos e Manutenção de Software

Leonardo Gresta Paulino Murta

leomurta@ic.uff.br