

# Engenharia de Software

Natália Schots

17/março/2015 – Aula 01

# Agenda

- Apresentações
- Estrutura das Aulas
- Critérios de Avaliação
- Visão Geral da Disciplina

# Quem sou eu? (1/2)

- **Formação Acadêmica**

- Doutorado em andamento (COPPE/UFRJ)
  - Área: Engenharia de Software/Qualidade de Software – Melhoria de Processos de Software/Análise de Desempenho de Processos
- Mestrado concluído em 2010 (COPPE/UFRJ)
  - Área: Engenharia de Software/Qualidade de Software – Melhoria de Processos de Software/Análise de Causas Raiz de Problemas
- Graduação em Ciência da Computação em 2007 (UFJF – Juiz de Fora/MG)

# Quem sou eu? (2/2)

- **Formação Profissional**

- Implementadora e Avaliadora MR-MPS para software
- Implementadora e Avaliadora MR-MPS para serviços
- Professora substituta na UERJ (2013-2014)
  - Disciplina: Gestão de TI
- Início na Rural: 2014.2
  - Disciplinas: Gerência de Projetos e Fundamentos de Sistemas

# Quem são vocês? (1/2)

— Se apresentem:

- Nome
- Fez/faz estágio?
- Fez/faz Iniciação Científica?
- Quais disciplinas mais gostou até agora?

# Quem são vocês? (2/2)

– Em uma folha, escrevam:

- A. Nível de domínio com leitura no idioma inglês: fluente (1), bom (2), razoável (3), pouco (4), nenhum (5)
- B. Nível de conhecimento nos seguintes tópicos: trabalho com isto (1), possuo conhecimento teórico (2), já li a respeito (3), já ouvi falar (4), nunca ouvi falar (5)
  - i. UML – *Unified Modeling Language*
  - ii. CMMI-DEV – *Capability Maturity Model Integration for Development*
  - iii. MR-MPS – Modelo de Referência de Melhoria de Processos de Software
  - iv. Métricas de software
  - v. Gerência de configuração
  - vi. Testes de software
- C. Temas que gostaria que fossem tratados

# Estrutura das Aulas (1/3)

- As aulas serão ministradas a partir de slides que serão posteriormente disponibilizados no Moodle
- Não gostaria que fosse um monólogo
  - Portanto, interrompam sempre que tiverem alguma dúvida ou um complemento/exemplo relacionado ao tópico apresentado
  - Compartilhem suas ideias com todos!

# Estrutura das Aulas (2/3)

- Além dos slides, sempre que possível, estudaremos artigos relacionados ao tópico apresentado
  - Leitura prévia e discussão em sala de aula
- Horário das aulas
  - Terça-feira: 10h às 12h
  - Sexta-feira: 8h30 às 10h



# Estrutura das Aulas (3/3)

- Nossa comunicação oficial será realizada via Moodle
  - Disponibilização dos slides
  - Envio de trabalho
  - Alteração de planejamentos
- Eventualmente, em caso de imprevistos, a comunicação será via e-mail
  - nataliach.lessa@gmail.com
  - Fiquem atentos!

# Visão Geral da Ementa (1/2)

- Processos de desenvolvimento de software
  - Ciclos de vida
  - Grupos de processos
  - Melhoria de processos
- Gerência de configuração
- Métricas de software
  - Dificuldades
  - Técnicas

# Visão Geral da Ementa (2/2)

- Qualidade de software
  - Qualidade do processo x qualidade do produto
  - Padrões de qualidade
- Testes de software
  - Fundamentos
  - Técnicas

# Critérios de Avaliação (1/2)

- Média será composta por:
  - 1 prova individual (P)
  - 2 trabalhos em grupo, nota individual (T1 e T2)
    - T1: Trabalho em sala, paralelo às aulas teóricas
    - T2: Trabalho com elaboração e apresentação de slides
  - Nota de participação – individual (N):
    - Participação nos demais trabalhos solicitados
    - Frequência às aulas

$$Média = \frac{P + T1 + T2 + N}{4}$$

# Critérios de Avaliação (2/2)

- Prova Optativa
  - A média final será composta pela média obtida anteriormente e a nota da prova optativa (PO)

$$Média Final = \frac{Média Anterior + PO}{2}$$

- Se Média Final  $\geq 5,0$  = APROVADO
- Se Média Final  $< 5,0$  = REPROVADO

# Presença em aula (1/2)

- A contabilização das presenças será realizada por meio de uma lista de presença
  - SEMPRE assine a lista
- Para estabelecer a nota de participação
  - $60\% < x < 75\%$  de presença
    - Nota de participação =  $y - 1$
  - $< 60\%$  de presença
    - Nota de participação = 0

# Presença em aula (2/2)

- Ausências justificáveis
  - Problemas de saúde (somente com atestado)
  - Falecimento de familiar (necessário comprovar)
- Reposição de prova
  - Não haverá segunda chamada
  - Quem perder a prova deverá fazer a prova final

# Cronograma Inicial

- Não haverá aulas:
  - 03/abril: feriado Sexta-feira santa
  - 21 e 24/abril: feriado Tiradentes e recesso
  - 01/maio: feriado Dia do Trabalho
  - 05/junho: recesso
- Prova
  - 02/junho
- Apresentações trabalho T2
  - 14 e 17/julho
- Prova Optativa
  - 21/julho



# Esta disciplina

- O que vocês entendem por “Engenharia de Software”?
- Qual a expectativa de vocês?

# 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)

# Por que Engenharia de Software?

## (1/2)



Como o cliente explicou



Como o lider de projeto entendeu



Como o analista planejou



Como o programador codificou



O que os beta testers receberam



Como o consultor de negocios descreveu



Valor que o cliente pagou



Como o projeto foi documentado



O que a assistencia tecnica instalou



Como foi suportado



Quando foi entregue



O que o cliente realmente necessitava

(Fonte: [http://rapidoerasteiro.files.wordpress.com/2008/05/projeto\\_software.jpg](http://rapidoerasteiro.files.wordpress.com/2008/05/projeto_software.jpg))

# Por que Engenharia de Software?

## (2/2)

- Além de saber programar, é necessário saber:
  - O **que** programar
  - **Como** programar
  - Se o que foi programado está **certo**?
  - ...
- É necessário conhecer o processo de desenvolvimento e seus produtos, saber medi-los e melhorá-los continuamente

# Complexidade dos softwares (1/3)

## – 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?



# Complexidade dos softwares (2/3)

## – 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?

# Complexidade dos softwares (3/3)

## – Tamanho

- Mais de 4 milhões de linhas de código
- Linguagem dominante (>99%): Ada

## – Documentação

- De 100 a 10.000 páginas por subsistema
- Total de 79 subsistemas integrados

## – Duração

- 4,5 anos de desenvolvimento

## – Ampla utilização de Engenharia de Software

## – Em operação desde 1995

- Zero acidentes graves até 2006



(Murta, 2008)

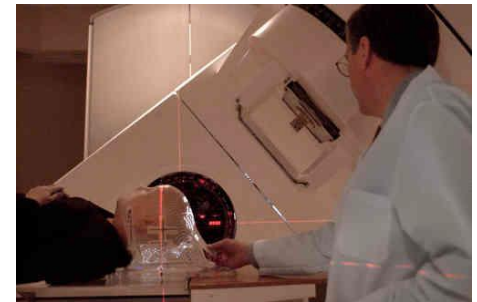
# Engenharia ou arte? (1/3)

- 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?



# Engenharia ou arte? (2/3)

- Caso real 1: Therac-25
  - Máquina de radioterapia controlada por computador
  - Problema:
    - Doses indevidas de radiação emitidas
  - Causas:
    - Interface com usuário inapropriada
    - Documentação deficiente
    - Software reutilizado sem ser adaptado para o novo hardware
    - Software de sensores de falha com defeito
  - Consequências
    - Ao menos 5 mortes entre 1985 e 1987



# Engenharia ou arte? (3/3)

- Caso real 2: Ariane 5
  - Foguete lançador de satélites
  - Problema:
    - O foguete se autodestruíu após o lançamento
  - Causa:
    - Software reutilizado sem ser adaptado para o novo hardware
    - Ausência de testes em solo deste software
  - Consequências
    - Prejuízo de mais de US\$ 370.000.000,00 em 1996



Obrigada!