

Engenharia de Software

Natália Schots

Agenda

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

Quem sou eu? (1/2)

- **Formação Acadêmica**

- Doutorado concluído em 2016 (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
- Professora efetiva na Rural: desde 2014.2
 - Disciplinas: Engenharia de Software, Gerência de Projetos, Empreendedorismo, Medição e Qualidade de Software

Quem são vocês?

- Respondam, por favor, o questionário
“Caracterização dos alunos” disponibilizado em:



Visão geral da disciplina

Estrutura das Aulas (1/2)

– **Aulas expositivas:**

- Exposição do conteúdo a partir de slides, que serão disponibilizados no SIGAA
 - Interrompam sempre que tiverem alguma dúvida ou um complemento/exemplo relacionado ao tópico
 - Compartilhem suas ideias com todos!
- Além dos slides, vídeos e materiais complementares serão disponibilizados no SIGAA

Estrutura das Aulas (2/2)

- **Apresentação do trabalho (em grupo)**
 - A cada etapa do trabalho, os alunos devem apresentar os principais resultados obtidos, dificuldades enfrentadas e dúvidas

Ferramentas de Comunicação

— SIGAA

- Disponibilização dos slides e dos materiais para estudo
- Envio de trabalhos
- Alteração de planejamentos

— E-mail

- Envio de dúvidas
- Comunicação em caso de imprevistos
- natalia.schots@ufrj.br

— Discord

- Compartilhamento de dúvidas e conhecimento

Alinhando Expectativas

- Prazo para respostas via e-mail ou Discord:
 - 24 horas
 - Segunda a sexta-feira
- Levem em consideração este acordo, principalmente para dúvidas e *feedback* de versões preliminares

Critérios de Avaliação (1/2)

- Média será composta por:
 - 2 provas individuais (P1 e P2)
 - Trabalho em grupo, **nota individual** (T1 e T2)
 - T1: etapas 1 e 2
 - T2: etapas 3, 4 e 5
 - Nota de participação – **individual** (N)
 - Participação em sala e em demais trabalhos
 - Frequência às aulas

$$\text{Média} = P1*0,2 + P2*0,3 + T1*0,2 + T2*0,2 + N*0,1$$

Critérios de Avaliação (2/2)

- Prova Optativa
 - A nota da prova optativa (PO) substituirá a menor nota obtida durante a disciplina
- Segunda chamada
 - Abrir processo de acordo com o regimento

Presença em aula

- A contabilização das presenças será realizada por meio da lista de chamada
 - A chamada será realizada **30min após o início da aula**. Após a realização da chamada, nenhuma presença será contabilizada
- Critério para estabelecer a nota de participação (N):
 - $60\% < x < 75\%$ de presença
 - $N = y - 3$ (onde y é a nota obtida por meio dos outros critérios – questionários e participação)
 - $< 60\%$ de presença
 - $N = 0$

Programação

- Disponível no Plano de Curso no SIGAA
- **Prova P1:** 16/dez
- **Prova P2:** 06/mar
- Apresentação do trabalho:
 - Etapa 1: 28/nov
 - Etapa 2: 12/dez
 - Etapa 3: 3/fev
 - Etapa 4: 13/fev
 - Etapa 5: 10/mar
- **Prova optativa:** 13/mar

Sujeito a
mudanças!

Visão Geral da Ementa (1/2)

- Processos de desenvolvimento de software
 - Ciclos de vida
 - Grupos de processos
 - Melhoria de processos
 - Métodos ágeis
- Engenharia de requisitos
 - Técnicas de elicitação de requisitos
 - Especificação de requisitos
 - Documentação de requisitos
 - Casos de uso

Visão Geral da Ementa (2/2)

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

Algumas referências-base (1/2)



Algumas referências-base (2/2)



*Guide to the Software
Engineering Body of Knowledge*

Editors

Pierre Bourque
Richard E. (Dick) Fairley



IEEE  computer society



Disponível em:

<https://www.computer.org/education/bodies-of-knowledge/software-engineering>

Disponível em:

<https://engsoftmoderna.info/>

Introdução à ES

O que é Engenharia de Software?

- “O estabelecimento e uso de um conjunto de **princípios de engenharia** com o objetivo de se construir software confiável, eficiente e viável economicamente em máquinas reais” (F. L. Bauer, 1969)
- “Aplicação de uma **abordagem sistemática, disciplinada e quantitativa** para o desenvolvimento, operação e manutenção de software, isto é, a aplicação da engenharia ao software” (IEEE, 1993)


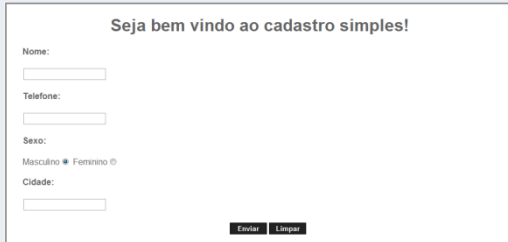
Engenharia ou arte? (1/8)

- 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? (2/8)

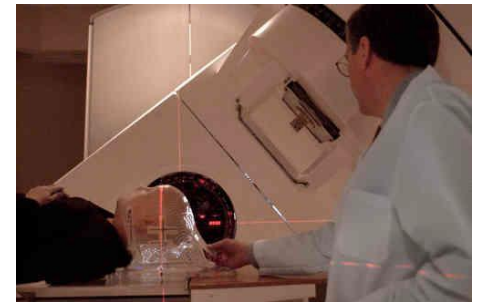
- Desenvolver software é uma arte ou uma engenharia?
 - É arte: depende da criatividade e habilidade dos envolvidos
 - Mas é engenharia também: é necessário planejar, projetar, construir, testar...
- A formalidade no emprego da engenharia depende da complexidade e criticidade do que está sendo construído

Engenharia ou arte? (3/8)

	Simples	Complexo
Engenharia Civil		
Engenharia de software		

Engenharia ou arte? (4/8)

- 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? (5/8)

- 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



Engenharia ou arte? (6/8)

- Caso real 3: Recall após incêndios (2019)



A decisão ocorre dias após a divulgação de vídeo nas redes sociais em que um Onix Plus aparece em chamas perto da cidade de Mirador (MA), quando se dirigia a Teresina (PI) no último dia 30. Antes, uma unidade do sedã da Chevrolet pegou fogo ao ser manobrado no pátio da fábrica da GM em Gravataí (RS). O proprietário diz ter escutado um barulho no carro e, quando parou e levantou o capô, viu as chamas. O veículo foi completamente destruído.

Segundo um comunicado interno da montadora datado no dia 5 de novembro, "em condições muito específicas de pressão atmosférica, temperatura ambiente, umidade relativa do ar e composição do combustível, condições combinadas pouco prováveis, o software de gerenciamento do motor pode, eventualmente, apresentar uma falha, com risco de danos ao motor e potencial incêndio. Esta condição é precedida de um alerta visual no painel de instrumentos (luz do motor).

Engenharia ou arte? (7/8)

- Caso real 4:

Falha de sistema bloqueia 6 mil cartões do Bilhete Único em São Paulo

Segundo a SPTrans, naquela data cerca de 6 mil cartões foram bloqueados por causa de “uma ação indevida, por parte de uma garagem de ônibus, no envio de dados para o sistema”. O usuário que se deparou com o problema e ligou para o 156, telefone de atendimento da prefeitura, recebeu na sexta, dia 25, a resposta de que seu cartão tinha sido bloqueado pela auditoria.

O **Metro Jornal** questionou a SPTrans se o problema não pode ser solucionado em outros locais que não o posto central, para facilitar para os usuários, prejudicados por uma falha no sistema. A empresa respondeu que “apenas neste posto é possível fazer, durante o atendimento do usuário, a análise necessária e a transferência dos créditos para o novo bilhete”.

Engenharia ou arte? (8/8)

- Caso real 5:



Mais de 300 pessoas morreram na queda de 737 Max na Indonésia e na Etiópia, em decorrência de problemas de fabricação da aeronave.

A Boeing assumiu que teve conhecimento dos problemas de software do modelo 737 Max, ao menos um ano antes de acontecerem dois acidentes fatais com suas aeronaves. A empresa fez o pronunciamento nesta segunda-feira (6), contando que encontrou inconformidades entre design e software, mas que não determinavam impactos na segurança ou operação do avião.

Por que Engenharia de Software? (1/2)



Como o cliente explicou



Como o líder 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

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

