

Engenharia de Software

Diana Braga
diana@ufc.br

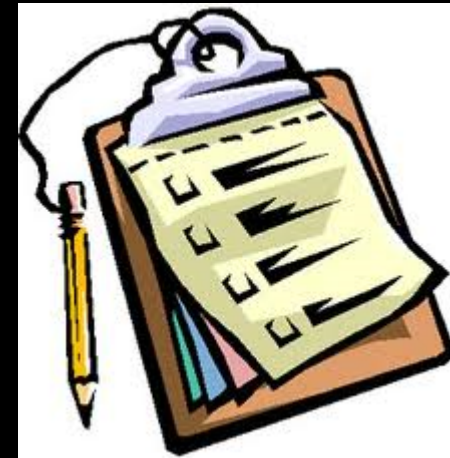
Agenda

Apresentação do Plano de Ensino

1. Motivação/ Importância
2. Ementa
3. Objetivos Gerais e Específicos
4. Metodologia de Ensino
5. Descrição do Conteúdo
6. Regras de Avaliação
7. Regras de Aprovação
8. Bibliografia
9. Outras Informações

Conhecendo as Experiências

Leitura Complementar



Motivação / Importância

- Na década de 1960, a introdução da tecnologia de circuitos integrados para a construção de computadores potencializou a indústria de software e teve uma implicação direta no que ficou conhecido na história como “a crise do software”. Desde aquela época, ficou claro que a construção de sistemas de software em grande escala **demandava uma disciplina sistemática e efetiva para o desenvolvimento que atenda aos prazos, aos custos e de maneira especial às necessidades e aos critérios de qualidade** esperado pelo contratante.
- A Engenharia de Software surgiu no final de 1960 como uma disciplina sistemática que estuda e descreve processos, métodos e ferramentas que devem ser utilizados para apoiar a atividade de construção de sistemas de software de uma forma efetiva.

Ementa

- Visão geral e introdutória dos princípios fundamentais e éticos-profissionais da Engenharia de Software. Introdução às atividades de engenharia de requisitos; projeto de software; modelos de desenvolvimento; e gerenciamento (qualidade, estimativa de custo, configuração, etc) na engenharia de software.

Objetivos Gerais e Específicos

- Objetivos Gerais:
 - Apresentar os principais métodos e técnicas da Engenharia de Software
- Objetivos Específicos:
 - Fornecer ao aluno uma visão sistemática da Engenharia de Software
 - Apresentar as principais disciplinas da Engenharia de Software
 - Capacitar o aluno para que este se torne apto a participar de projetos de desenvolvimento de software

Metodologia de Ensino

- A disciplina será ministrada em aulas teóricas e práticas, podendo-se utilizar, dentre outras metodologias, práticas em laboratório e trabalhos de equipes, onde os conteúdos poderão ser ministrados de acordo as especificidades do grupo de alunos e da disciplina



Regras de Avaliação

- $\text{Nota final} = (\text{Nota1} + \text{Nota2})/2$
- A Nota1 será realizada através de listas de exercícios e apresentações de temas que ocorrerão no decorrer da disciplina (individual e em equipe)
- A Nota2 será uma prova escrita
 - **2ª chamada da prova: solicitar via SIPPA**



Regras de Aprovação

- Será aprovado o aluno que frequentar 75% da carga horária da disciplina.
- Será aprovado ao final da disciplina por média o aluno que apresentar média aritmética das notas resultantes das avaliações progressivas, igual ou superior a 7 (sete)
- Os alunos que obtiverem ao final da disciplina média inferior a 7 (sete) e igual ou superior a 4 (quatro) irão ser submetidos a prova final, e serão aprovados os que alcançarem nota igual ou superior a 4 (quatro) nesta avaliação, média final igual ou superior a 5 (cinco)

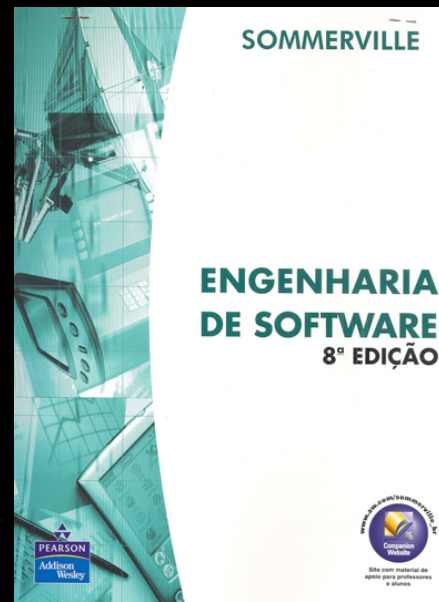
Bibliografia

- Bibliografia Básica
 - SOMMERVILLE, I. Engenharia de software. 9 ed. Addison Wesley, 2011. ISBN: 9788579361081.
 - PRESSMAN, R. Engenharia de software. 6 ed. Pearson, 2009.
 - LARMAN, Craig. Utilizando UML e padroes : uma introducao a analise e ao projeto orientados a objetos. 5. ed. Porto Alegre: Bookman, 2007. 695 p. ISBN 856003152-9
- Bibliografia Complementar
 - TELES, V. Extreme programming. Novatec. 2004. ISBN: 8575220470
 - MOLINARI, L. Gerência de configuração: técnicas e práticas no desenvolvimento do software. VISUAL BOOKS. ISBN: 8575022105
 - DELAMARO, M.; MALDONADO, J.C.; JINO, M. Introdução ao teste de software. Campus, 2007.
 - KERIEVSKY, J. Refatoração para padrões. Bookman, 2008. ISBN:9788577802449
 - PEZZÉ, M.; YOUNG, M. Teste e análise de software: processos, princípios e técnicas. Bookman, 2008. ISBN: 9788577802623
 - PILONE, D.; MILES, R. Use a cabeça! desenvolvimento de software. ALTA BOOKS, 2008. .



Bibliografia

- Roger Pressman
- Sommerville



Outras Informações

- Outros materiais poderão ser fornecidos ao longo da disciplina, além do material complementar e incluídos no sistema
- Email para contato: diana@ufc.br

Outras Informações

- Uso do SIPPA
- <https://sistemas.quixada.ufc.br/apps/sippa/>
- Para registro da frequência e notas



The image shows the SIPPA login interface. At the top is the SIPPA logo, which consists of a blue square icon with a white checkmark and the word 'SIPPA' in a bold, blue, sans-serif font. Below the logo are three input fields: 'Login:', 'Senha:', and 'Tipo de conta:'. The 'Tipo de conta:' field is a dropdown menu with the text 'Escolha uma opção' and a small downward arrow. Below these fields is a CAPTCHA image showing the number '2003' in blue, with a black line drawn over it. Below the CAPTCHA is a text prompt: 'Se os caracteres da imagem acima não estiverem legíveis, [clique aqui](#)'. This is followed by another input field for the CAPTCHA characters. At the bottom are two buttons: 'Entrar' and 'Limpar'. Below the buttons is a link: 'Esqueceu sua senha [clique aqui](#)'.

SIPPA

Login:

Senha:

Tipo de conta:
Escolha uma opção ▼

2003

Se os caracteres da imagem acima não estiverem legíveis, [clique aqui](#)
Digite a sequência de caracteres da imagem acima:

Entrar Limpar

Esqueceu sua senha [clique aqui](#)

Outras Informações

- Uso do Moodle
- <https://moodle2.quixada.ufc.br>
 - /course/view.php?id=846
- Para entregas de atividades e visualização do material e conteúdo das aulas



ENGENHARIA
DE
SOFTWARE –
02A – 2022.1

Acesso

The banner features a dark blue background with various colorful icons representing technology and education, including a laptop, a smartphone, a tablet, a book, a pencil, a gear, and a Wi-Fi symbol. The text 'ENGENHARIA DE SOFTWARE – 02A – 2022.1' is displayed in a clean, white, sans-serif font. Below the text is a blue rectangular button with the word 'Acesso' in white.

Conhecendo as Experiências...

- Apresentação dos alunos
 - Nome?
 - Experiência na área?
 - O que você deseja descobrir com a ajuda da disciplina?



Leitura Complementar

- **What does a software engineer do?** (Posted on March 3, 2010 by Owen Pellegrin)
- <http://www.owenpellegrin.com/blog/life/what-does-a-software-engineer-do/>

Dúvidas?



Introdução à Engenharia de Software

Objetivos da Aula

- Introduzir a engenharia de software e explicar sua importância
- Introduzir os conceitos de ES
- Responder as principais perguntas sobre engenharia de software

Surge o Termo Engenharia de Software

- 1968 em uma conferência sobre a crise do software
 - Desenvolvimento informal não era mais suficiente
 - Custo do hardware cai e custo do software sobe



Surge o Termo Engenharia de Software

- Comentário de participante da Conferência da OTAN
 - "Certos sistemas estão colocando demandas que estão além das nossas capacidades... Em algumas aplicações não existe uma crise... Mas **estamos tendo dificuldades com grandes aplicações.**"
- A conferência produziu um relatório, com mais de 130 páginas, que afirmava a necessidade de que software fosse construído com base em princípios práticos e teóricos, tal como ocorre em ramos tradicionais e bem estabelecidos da Engenharia. Para deixar essa proposta mais clara, decidiu-se cunhar o termo Engenharia de Software. Por isso, a Conferência da OTAN é considerada o marco histórico de criação da área de Engenharia de Software.

Engenharia de Software

- Mas qual foi o propósito de se usar o termo Engenharia?
 - O termo Engenharia foi usado justamente para associar o "conceito" Engenharia ao desenvolvimento de software, em outras palavras é ter uma abordagem sistemática, disciplinada e quantificada ao desenvolvimento, operação e manutenção de software (IEEE, 1990)
 - A engenharia é aplicação de princípios matemáticos e científicos, experiência, julgamento e bom senso para trazer coisas que beneficiam as pessoas
 - A Engenharia de Software segue este mesmo raciocínio, tendo como objetivo definir e exercitar processos, métodos, ferramentas e ambientes para construção de software que satisfaça necessidades de cliente e usuário dentro de prazos e custos previsíveis

O Que é Engenharia de Software?

É uma disciplina da engenharia dedicada a todos os aspectos da produção de software.

- Engenheiros de software devem adotar uma abordagem sistemática e organizada para o seu trabalho e usar técnicas e ferramentas apropriadas, de acordo com o problema a ser resolvido, e com as restrições e recursos disponíveis.

Fonte: Engenharia de Software - Ian Sommerville

Definição de Engenharia de Software

Área da Computação destinada a investigar os desafios e propor soluções que permitam desenvolver sistemas de software — principalmente aqueles mais complexos e de maior tamanho — de forma produtiva e com qualidade

Fonte: Engenharia Moderna – Marco Túlio

Engenharia de Software

- A Engenharia de Software é uma área dinâmica e em constante mudança. O profissional precisa saber se adaptar a essas mudanças, pois as tecnologias evoluem constantemente e o profissional precisa estar sempre atualizado.
 - As habilidades técnicas não são suficientes (O comportamento conta muito!!)
- Trabalhar em equipe não é simples, pois quando colocamos várias pessoas com personalidades distintas para trabalhar juntas, é necessário gerir os conflitos e convergir as atividades para um objetivo comum.

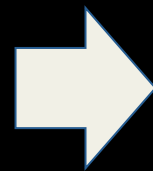
Na Engenharia de Software: Dificuldades Essenciais

Complexidade

Conformidade

Facilidade de Mudanças

Invisibilidade



Tornam Engenharia
de Software diferente
de outras engenharias

Quais os atributos de um bom software?

- O software deve entregar ao cliente as funcionalidades e desempenho solicitados e deve ser de fácil manutenção, confiável e usável
- Manutenibilidade
 - O software deve ser capaz de evoluir para atender às mudanças exigidas
- Confiança
 - O software deve ser confiável e seguro
- Eficiência
 - O software não deve desperdiçar os recursos do sistema
- Usabilidade
 - O software deve ser usável para o tipo de usuário para que ele foi projetado

Desafios

Quais são os Desafios da Engenharia de SW?



Desafios

- Desafios da Engenharia de SW
 - Lidar com sistemas legados
 - Atender à crescente diversidade
 - Prazos reduzidos

Custos do Software

- As economias de TODAS as nações desenvolvidas dependem de software
- Mais e mais sistemas são controlados por software
- Os custos variam dependendo do tipo do sistema a ser desenvolvido e dos seus atributos tais como desempenho e confiabilidade
- A distribuição dos custos depende do modelo de desenvolvimento usado

Custos do Software

- O custo do software geralmente é a maior parte do custo de um sistema de computador
 - O custo de software em um PC geralmente é maior que o custo de hardware
- Custo de manutenção X Custo do desenvolvimento
 - Para sistema de vida longa, o custo de manutenção pode ser muitas vezes o custo de desenvolvimento
- **A Engenharia de Software também está relacionada ao desenvolvimento de software com custo adequado**

Princípios

- Princípios

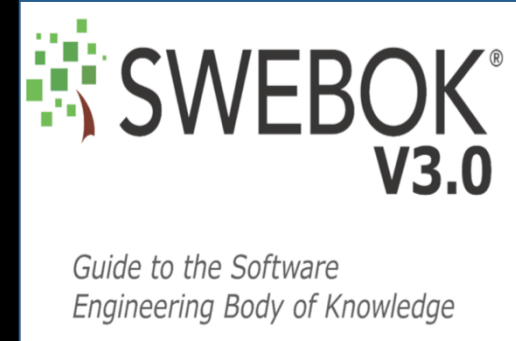
- Razão porque tudo existe: para fornecer valor aos seus usuários
- Mantenha a coisa simples: todo projeto deve ser tão simples quanto possível
- Mantenha a visão: uma visão clara é essencial para o sucesso de um projeto de software
- O que você produz os outros vão consumir: sempre especifique, projete e implemente sabendo que mais alguém terá de entender o que você está fazendo
- Esteja aberto para o futuro: nunca projete a si mesmo em um beco sem saída
- Planeje com antecedência o reuso: reduz custo e a um então valor dos componentes e do sistema ao qual são incorporados
- Pense: raciocinar clara e completamente antes da ação quase sempre produz melhores resultados

Ferramenta CASE (Computer-Aided Software Engineering)

- Programas que são usados para dar apoio às atividades de do processo de software
- Ferramentas CASE são usadas no suporte a modelos
- Ferramentas para ajudar nas atividades do processo como requisitos, projeto, programação, depuração e teste

O que se estuda em ES?

1. Engenharia de Requisitos
2. Projeto de Software
3. Construção de Software
4. Testes de Software
5. Manutenção de Software
6. Gerência de Configuração
7. Gerência de Projetos
8. Processos de Software
9. Modelos de Software
10. Qualidade de Software
11. Prática Profissional
12. Aspectos Econômicos



(1) Requisitos de Software

- Requisitos: o que sistema deve fazer para atender aos seus clientes com qualidade de serviço
- Engenharia de Requisitos: atividades onde os requisitos de um sistema são especificados, analisados, documentados e validados

Requisitos Funcionais vs Não-Funcionais

- **Funcionais:** "o que" um sistema deve fazer
 - Funcionalidades ou serviços ele deve implementar
- **Não-funcionais:** "como" um sistema deve operar
 - Sob quais "restrições" e qualidade de serviço

Exemplos de Requisitos Não-Funcionais

- Desempenho: dar o saldo da conta em 5 segundos
- Disponibilidade: estar no ar 99.99% do tempo
- Capacidade: armazenar dados de 1M de clientes
- Tolerância a falhas: continuar operando se São Paulo cair
- Segurança: criptografar dados trocados com as agências
- Privacidade: não armazenar localização dos usuários
- Interoperabilidade: se integrar com os sistema do BACEN
- Manutenibilidade: bugs devem ser corrigidos em 24 hs
- Usabilidade: versão para celulares e tablets

(2) Projeto de Software

- Definição dos principais **componentes** de um sistema
- E de suas **interfaces**
- Estuda-se:
 - Propriedades de projeto
 - Princípios de projetos
 - Padrões de projeto

(3) Construção de Software

- Implementação (codificação) do sistema
- Algumas preocupações:
 - Algoritmos e estruturas de dados
 - Ferramentas: IDEs, depuradores, etc
 - Bibliotecas e frameworks
 - Padrões de nome

(4) Testes de Software

- Verificam se um programa apresenta um resultado esperado, ao ser executado com casos de teste
- Podem ser:
 - Manuais
 - Automatizados

Observação importante



Testes de software mostram a presença de bugs, mas não a sua ausência.
-- Edsger W. Dijkstra



Allen Holub
@allenholub

...

All software is tested. The real question is whether the tester is you or your users.

12:49 PM · May 28, 2020 · TweetDeck

Defeitos, Bugs, Falhas

- O seguinte código possui um **defeito** (*defect*) ou um **bug**:

```
if (condition)
    area = pi * raio * raio * raio;    // código defeituoso;
```

- O certo é "área é igual a pi vezes raio ao quadrado"
- Quando esse código for executado ele vai causar uma **falha** (*failure*); ou seja, um resultado errado.

Verificação x Validação

- **Verificação:** estamos implementando o sistema corretamente?
 - De acordo com os requisitos e especificações
- **Validação:** estamos implementando o sistema correto?
 - O sistema que os clientes querem
 - Testes de aceitação com os usuários

(5) Manutenção de Software

- Tipos de Manutenção:
 - Corretiva
 - Preventiva
 - Adaptativa
 - Refactoring
 - Evolutiva

Exemplo de Manutenção Preventiva: Y2K Bug

- Manutenção realizada no "tamanho" de campos *data*, de DD-MM-AA para DD-MM-AAAA
- Riscos e prejuízos foram super-estimados



Refactoring

- Manutenção para incrementar manutenibilidade; ou seja, não corrige bugs, não implementa novas funcionalidades
- Exemplos:
 - Renomear variável, classe, etc
 - Extrair função
 - Mover função
 - etc

Sistemas Legados

- Sistemas antigos, usando linguagens, SOs, BDs antigos
- Manutenção custosa e arriscada
- Muitas vezes, são muito importantes (legado != irrelevante)

(6) Gerência de Configuração

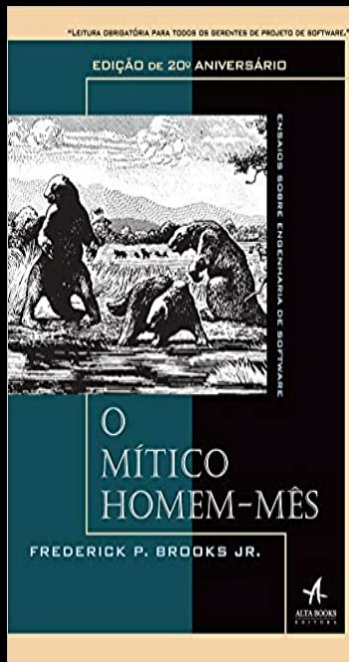
- Todo software é desenvolvido usando um sistema de controle de versões (exemplo: git)
- Atua como uma "fonte da verdade" sobre o código
- Permite recuperar versões antigas

(7) Gerência de Projetos

- Algumas atividades:
 - Negociação contratos (prazos, valores, cronograma)
 - Gerência de RH (contratação, treinamento etc)
 - Gerenciamento de riscos
 - Acompanhamento da concorrência, marketing, etc

Lei de Brooks

- Incluir novos devs em um projeto que está atrasado, vai deixá-lo mais atrasado ainda



Stakeholders

- Todas as "partes interessadas" em um projeto de software
- São aqueles que **afetam** ou **são afetados** pelo projeto
- Isto é, desenvolvedores, clientes, usuários, gerentes, empresas terceirizadas, fornecedores, governo etc

(8) Processos de Desenvolvimento de Software

- Um processo de software define quais atividades devem ser seguidas para construir um sistema de software
- Dois principais modelos
 - Waterfall ("cascata")
 - Ágil (ou incremental ou iterativo)

Exemplo: Waterfall vs Agile (apenas para ilustrar)

- Problema: construir uma ponte
- Solução Waterfall (*plan-and-document*):
 - Projeto preliminar e requisitos (largura etc)
 - Maquete e projeto de engenharia, estrutural, etc
 - Simulação em um túnel de vento
 - Construção da ponte
 - Entrega e inauguração

Exemplo: Waterfall vs Agile (cont.)

- Solução ágil (incremental, iterativo):
 - Constrói-se uma primeira versão, com uma única pista
 - Em seguida, constrói-se uma segunda pista
 - Depois, duplicam-se as duas pistas, etc
- Para projetos com grau de "incerteza", método ágil faz mais sentido

(9) Modelagem de Software

- Modelo = representação mais alto nível do que o código
- Modelo pode ser usado para:
 - explicar o "design" para os devs
 - documentar o projeto
 - facilitar compreensão e manutenção

(10) Qualidade de Software

- Qualidade Externa:
 - Correção, robustez, extensibilidade, reusabilidade, eficiência, compatibilidade, facilidade de uso, portabilidade
- Qualidade Interna:
 - Modularidade, legibilidade, testabilidade, manutenibilidade, etc

(11) Prática Profissional

- Ética
- Certificações
- Regulamentação da Profissão
- Cursos de graduação
- Currículo de Referência

Aspectos Éticos

- Engenheiros de Software começam a questionar o uso que as empresas fazem do software desenvolvido por eles

Cybersecurity

Google Engineers Refused to Build Security Tool to Win Military Contracts

A work boycott from the Group of Nine is yet another hurdle to the company's efforts to compete for sensitive government work.

<https://www.bloomberg.com/news/articles/2018-06-21/google-engineers-refused-to-build-security-tool-to-win-military-contracts>

(12) Aspectos Econômicos

- Qual o retorno de investimento?
- Como monetizar meu software? (anúncios, assinaturas, etc)
- Quanto cobrar pelo desenvolvimento de um sistema?
- Qual a melhor licença para o meu software?

Responsabilidade Profissional e Ética

- Como outras disciplinas, a ES está dentro de uma estrutura de condições legais e sociais que limita a liberdade dos engenheiros, que, por sua vez, devem admitir que o seu trabalho implica responsabilidade mais amplas do que a aplicação de habilidade técnicas
- Devem também se comportar de forma responsável ética e moralmente para serem respeitados como profissionais
- Confidencialidade
- Competência
- Direitos sobre propriedade intelectual
- Mau uso de computadores



Responsabilidade Profissional e Ética

- Por causa dos seus papéis no desenvolvimento de software, os engenheiros possuem várias oportunidades para praticar o bem ou o mal e para influenciar outros a praticar o bem ou o mal
- Para assegurar o máximo possível, que os seus esforços serão direcionados para o bem, os engenheiros deverão cometer-se em tornar a Engenharia de Software numa profissão benéfica e respeitada

Código de ética e prática profissional do Engenheiro de Software

1. PÚBLICO - Engenheiros de software devem agir de forma coerente com o interesse público.
2. CLIENTE E EMPREGADOR - Engenheiros de software devem agir de uma forma que é, no melhor interesse de seus clientes e empregadores, coerente com o interesse público.
3. PRODUTO - Os engenheiros de software devem assegurar que seus produtos e modificações relacionadas atendem os mais elevados padrões profissionais possíveis.
4. JULGAMENTO - Os engenheiros de software devem manter integridade e independência em seu julgamento profissional.
5. GESTÃO - Gerentes e líderes de engenharia de software devem aceitar e promover uma abordagem ética para a gestão de desenvolvimento e manutenção de software.
6. PROFISSÃO - Engenheiros de software devem fomentar a integridade e reputação da profissão compatível com o interesse público.
7. COLEGAS - Engenheiros de software devem ser justos e apoiar os seus colegas.
8. AUTO - Engenheiros de software devem participar na aprendizagem ao longo da vida sobre a prática da sua profissão e devem promover uma abordagem ética à prática da profissão.

- Fonte: <http://www.acm.org/about/se-code>

Princípio 1: Público

Os engenheiros de software deverão agir de acordo com o interesse público

- 1.01. Aceitar todas a responsabilidade pelo seu trabalho.
- **1.02. Moderar o seu interesse, do empregador, do cliente e dos utilizadores com o bem público.**
- 1.03. Aprovar software, que apenas seja seguro, corresponda às especificações, passe nos testes apropriados, não diminua a qualidade de vida, não diminua a privacidade ou prejudique o ambiente. O último efeito do trabalho deverá ser para o bem público.
- 1.04. Alertar as devidas pessoas ou autoridades, de algum potencial perigo para o utilizador, público ou ambiente, que esteja razoavelmente associado ao software ou documentos relacionados.
- 1.05. Cooperar em esforços para alertar, assuntos de grave preocupação pública, causado por software na sua instalação, manutenção, suporte ou documentação.
- 1.06. Ser justo e evitar, enganos nas suas declarações, particularmente nas públicas relacionadas com software, documentos, métodos e ferramentas.
- 1.07. Considerar problemas de incapacidade física, alocação de recursos, desvantagens económicas e outros fatores que podem diminuir o acesso aos benefícios do software.
- 1.08. Ser encorajado em voluntariar-se a técnicas profissionais para boas causas e contribuir para a educação do público.

Princípio 2: Cliente e Empregador

Os engenheiros de Software agirão de acordo com os melhores interesses de seus clientes e empregadores, e consistente com o interesse público

- **2.01. Providencia serviço em áreas da sua competência, sendo honestos e indicando as suas limitações da sua experiência e educação.**
- 2.02. Não permitir o uso de software que seja obtido ilegalmente e não eticamente.
- 2.03. Usar a propriedade do cliente ou empregador apenas de maneira autorizada, e com o conhecimento e consentimento deste.
- 2.04. Assegurar que qualquer documento em que eles confiem seja aprovado, quando requerido por alguém autorizado a aprová-lo.
- **2.05. Manter privado toda a informação confidencial que adquiriram no seu trabalho, desde que a confidencialidade seja consistente com o interesse público e a lei.**
- 2.06. Identificar documentos, recolher provas e relatar ao cliente ou empregador, para saber a sua opinião se um projeto irá falhar, se será dispendioso, violar a lei, ou se será problemático.
- 2.07. Identificar, documentar e relatar ao seu cliente ou empregador, problemas de preocupação social de que eles estejam cientes em software ou documentos.
- **2.08. Não aceitar trabalho externo em detrimento do seu trabalho para o seu principal empregador.**
- **2.09. Não promover nenhum interesse contrário ao seu empregador ou cliente, a não ser que, um assunto ético esteja comprometido, nesse caso deverá informar o empregador ou outra autoridade acerca desse assunto.**

Princípio 3: Produto

Os engenheiros de Software deverão assegurar que os seus produtos e as alterações a eles relacionadas cumprem o mais alto padrão profissional possível

- **3.01. Empenhar-se por mais qualidade, custos aceitáveis, e prazos razoáveis, assegurando que significantes tradeoff sejam claros e aceites pelo empregador e o cliente, e estejam disponíveis para consideração por parte do utilizador e do público.**
- 3.02. Assegurar objetivos propícios para qualquer projeto em que trabalha.
- 3.03. Identificar e definir assuntos éticos, económicos, culturais, legais e ambientes que estejam relacionados com o projeto.
- 3.04. Assegurar que são qualificados para qualquer projeto que trabalhem, ou propor para trabalhar numa combinação de educação, treinamento e experiência.
- 3.05. Assegurar que o método apropriado é usado em qualquer projeto em que trabalha.
- 3.06. Trabalhar seguindo normas profissionais.
- 3.07. Empenhar-se em entender as especificações do software em que irão trabalhar.
- **3.08. Assegurar que as especificações de software estejam bem documentadas e que satisfaça os requisitos dos clientes.**
- **3.09. Assegurar estimativas realistas do custo, prazos possíveis, qualidade, em qualquer projeto em que trabalhem.**
- 3.10. Assegurar testes adequados, debugging e revisão do software e documentos relacionados em que trabalhem.
- 3.11. Assegurar uma documentação adequada, incluindo problemas encontrados e soluções adotadas para qualquer projeto em que trabalhem.
- **3.12. Trabalhar para desenvolver software que respeite a privacidade de quem irá ser afetado por esse software.**
- **3.13. Usar apenas informação precisa, obtida por meios éticos e legais, e usar apenas de maneira autorizada.**
- 3.14. Manter a integridade da informação, tendo atenção a sua desatualização e possível falha.
- 3.15. Tratar todas as formas de manutenção de software com o mesmo profissionalismo como se trata-se de um novo desenvolvimento.

Princípio 4: Julgamento

Os engenheiros de software manterão integridade e independência em seu julgamento profissional

- 4.01. Acomodar todos os julgamentos técnicos que necessitem suporte e mantêm os valores humanos.
- **4.02. Apenas endossar documentos que foram preparados sob sua supervisão ou que estejam na sua área de competência e que estejam de acordo.**
- 4.03. Manter objetividade profissional com respeito a qualquer software ou documentos que sejam pedidos para avaliar.
- **4.04. Não aceitar suborno, pagamentos duplos ou outras práticas financeiras impróprias.**
- 4.05. Revelar a todas as partes interessadas os conflitos de interesse que não possam ser evitados.
- **4.06. Recusar em participar, como membros ou conselheiros, em grupos privados, governamentais ou profissionais preocupados com assuntos relacionados com software em que possa existir conflitos de interesse com os clientes ou empregadores.**

Princípio 5: Gestão

Os gerentes e líderes de engenharia de software adotarão e promoverão uma abordagem ética para a gestão do desenvolvimento e da manutenção do software

- **5.01. Assegurar uma boa gestão para qualquer projeto em que trabalhar, incluindo procedimentos eficazes para promoção da qualidade e redução do risco.**
- 5.02. Assegurar que os engenheiros sejam informados das normas antes de as usar.
- 5.03. Assegurar que os engenheiros conhecem as políticas do empregador e procedimentos para proteger suas palavras-chave, ficheiros e informação confidencial para o empregador ou confidencial para outros.
- 5.04. Designar trabalho apenas após ter em conta as contribuições da educação e experiência mostrada e o desejo de prosseguir os estudos e adquirir mais experiência.
- 5.05. Assegurar estimativas realistas dos custos, prazos, pessoa, qualidade e gastos em qualquer projeto em que trabalhe e providenciar acesso a essas estimativas.
- 5.06. Atrair potenciais engenheiros de software apenas com a descrição completa das condições de trabalho.
- 5.07. Oferecer uma remuneração justa.
- 5.08. Não impedir injustamente que uma pessoa tome um cargo para o qual está qualificada.
- 5.09. Assegurar que há um acordo justo quanto à propriedade de qualquer software, processo, pesquisa, escrito ou outra propriedade intelectual para a qual um engenheiro de software contribuiu.
- **5.10. Providenciar auditorias para qualquer violação das políticas do empregador ou deste código.**
- **5.11. Não pedir a um engenheiro de software que faça algo de inconsistente com o código.**
- **5.12. Não punir ninguém por expressar problemas éticos para com um projeto.**

Princípio 6: Profissão

Os engenheiros de software fomentarão a integridade da profissão, de modo consistente com o interesse público

- 6.01. Ajudar a organizar um ambiente favorável para agir eticamente.
- 6.02. Promover conhecimento público da engenharia de software.
- 6.03. Expandir o conhecimento da engenharia de software participando em organizações profissionais, encontros e publicações.
- **6.04. Apoiar, como membros da profissão, outros engenheiros de software e seguidores deste código.**
- **6.05. Não promover o seu interesse pessoal às custas da profissão, cliente ou empregador.**
- 6.06. Obedecer a todas as leis que regem o seu trabalho, a não ser que, em circunstâncias especiais, sejam inconsistentes com o interesse público.
- 6.07. Ser preciso na apresentação das características do software em que trabalharam evitando falsas reivindicações mas também reivindicações que possam ser especulativas, vazias, confusas, enganadoras ou duvidosas.
- 6.08. Tomar a responsabilidade em detectar, corrigir e relatar erros no software e documentos associados em que trabalharam.
- 6.09. Assegurar que clientes, empregadores e supervisores sabem que o engenheiro tem compromisso para com este código de ética e respectivas ramificações deste compromisso.
- 6.10. Evitar associações com negócios e organizações que estão em conflito com este código.
- 6.11. Reconhecer que violar este código é inconsistente com ser um engenheiro de software profissional.
- 6.12. Expor a sua preocupação às pessoas envolvidas quando detectadas violações significativas do código.
- 6.13. Relatar as violações do código às autoridades quando for impossível consultar as pessoas envolvidas.

Princípio 7: Os Colegas

Os engenheiros de software serão justos e darão apoio aos seus colegas

- 7.01. Encorajar os colegas a aderir ao código.
- 7.02. Assistir os colegas no desenvolvimento profissional.
- 7.03. Reconhecer o trabalho dos outros e não se apropriar de mérito alheio.
- 7.04. Rever o trabalho dos outros com objetividade e de uma maneira bem documentada.
- 7.05. Ouvir sempre as opiniões, preocupações e queixas de um colega.
- 7.06. Ajudar os colegas a tomar consciência das normas atuais de trabalho, quer sejam políticas e procedimentos relacionadas com a protecção de palavras-chave, ficheiros e outras informações confidenciais ou outras medidas de segurança em geral.
- 7.07. Não interferir, injustamente na carreira de um colega, no entanto, tendo em conta o empregador, cliente ou o interesse público poderão compelir o engenheiro a questionar a competência de um colega.
- 7.08. Em situações fora da sua área de competência pedir a opinião de outros profissionais que tenham competência dessa área.

Princípio 8: Pessoal

Os engenheiros de software participarão de aprendizagem constante com relação à prática de sua profissão e promoverão uma abordagem ética dessa prática

- 8.01. Prolongar o seu conhecimento no desenvolvimento, na análise, especificação, design, desenvolvimento, manutenção, teste de software e documentação relacionada, em conjunto com a gestão do desenvolvimento do processo.
- 8.02. Melhorar a sua aptidão em criar software seguro, de confiança e útil a um custo e prazo razoável.
- 8.03. Melhorar a sua aptidão em produzir documentação precisa, informativa e bem escrita.
- 8.04. Melhorar a sua compreensão do software e de documentos em que irão trabalhar e do ambiente em que vai ser utilizado.
- 8.05. Melhorar o seu conhecimento de normas relevantes e da lei, que regula o software e documentos relacionados em que trabalhem.
- 8.06. Melhorar o seu conhecimento acerca deste Código, a sua interpretação e aplicação no trabalho.
- 8.07. Não dar um tratamento injusto a ninguém por causa de preconceitos irrelevantes.
- 8.08. Não influenciar outros a agir contra o Código.
- 8.09. Reconhecer que violações pessoais deste Código são inconsistentes com ser um engenheiro de software profissional.

Responsabilidade Profissional e Ética

- O desenvolvimento de software implica um conjunto de responsabilidades perante várias etapas.
- A equipe de desenvolvimento tem de garantir um produto adequado ao cliente e às suas necessidades ajudando-o, eventualmente, a descobrir o que ele realmente necessita
- A qualidade do produto deve ser avaliada várias vezes durante o desenvolvimento de maneira a diminuir os riscos e obter um produto final com qualidade
- A elaboração de um projeto de software tem algumas restrições tais como a plataforma computacional e a sua parte legal, pelo que o cliente deve ser informado de todas as opções tomadas.
- Uma preocupação importante a ter em conta pelos engenheiros de software é a integridade e originalidade da aplicação desenvolvida e condenar qualquer forma de plágio.

Pergunta...

- Todos os casos de ética que envolvem computadores são problemas de ética na área de computação?

A ética na área de computação deve abranger as ações dos profissionais de computação no seu papel como profissionais de computação

Responsabilidade Profissional e Ética

- Os engenheiros de software têm de possuir um comportamento responsável, ético e íntegro. Devem também garantir e respeitar os níveis de confidencialidade dos dados e a segurança da comunicação
- O desrespeito desta conduta pode acarretar as seguintes consequências
 - O sistema não reflete as necessidades reais dos clientes
 - Sistema é inconsistente ou incompleto
 - O cliente e os usuários não ficam satisfeitos com o sistema
 - Os usuários não usam todos os recursos do sistema
 - O sistema não é confiável, apresentando falhas ou interrupções na operação
 - Os custos de manutenção são altos

Casos - Dilemas Éticos

- Comportamentos antiéticos ou dilemas éticos podem ser desencadeados ou enfrentados por qualquer um dos stakeholders envolvidos no sistema de troca, inclusive o consumidor (MALHOTRA; MILLER, 1998)
 - Neste sentido, práticas deletérias executadas pelos consumidores podem ter efeitos tão nocivos quanto àquelas praticadas pelas empresas ofertantes.
- **Pirataria Digital**
- Este fenômeno tem causado grandes prejuízos aos diversos stakeholders envolvidos na relação de troca
 - à indústria tende a dedicar menos recursos à pesquisa e desenvolvimento de novos produtos
 - aos consumidores que comprem os softwares originais são obrigados a pagar mais caro para cobrir o prejuízo causado por aqueles que pirateiam
 - ao governo que perde em termos de arrecadação de impostos, entre outros
 - à sociedade tendo em vista a ligação do roubo de propriedade intelectual com o crime organizado, incluindo tráfico de drogas, armas, pessoas e “lavagem” de dinheiro

Casos - Dilemas Éticos

- Spam
 - A utilização do e-mail para fins não consentidos pelo destinatário deste tipo de correio electrónico

Casos - Dilemas Éticos

- **Sistemas Críticos**
- Sistemas que exigem maior responsabilidade por parte dos engenheiros de software
 - Isto porque qualquer falha tem, geralmente, consequências gravíssimas
- Um sistema crítico é avaliado segundo os seguintes critérios
 - Confiança que o utilizador pode ter no sistema
 - Probabilidade de estar disponível para fornecer serviços quando solicitado
 - Probabilidade de um sistema estar operacional (sem avarias) por um dado período de tempo
 - Capacidade do sistema se proteger de ameaças externas
- O engenheiro de software deve ter em mente estes critérios principais e assumir total responsabilidade na aprovação de um software. Estes sistemas necessitam de técnicas de desenvolvimento e de avaliação que garantam que o produto é seguro. Depois desta fase vem outra igualmente importante que consiste na verificação e validação do produto

Casos - Dilemas Éticos

Sistemas Críticos

- Você suspeita algo de errado em um sistema importante que está sendo desenvolvido por um grande parceiro da sua empresa
- Quando revelar à gerência um problema com um projeto de software?

Exemplo de uma falha em um sistema crítico

07/09/2017 às 22h30

Equifax teve ataque de hackers e dados de 143 milhões foram expostos

Por Dow Jones Newswires

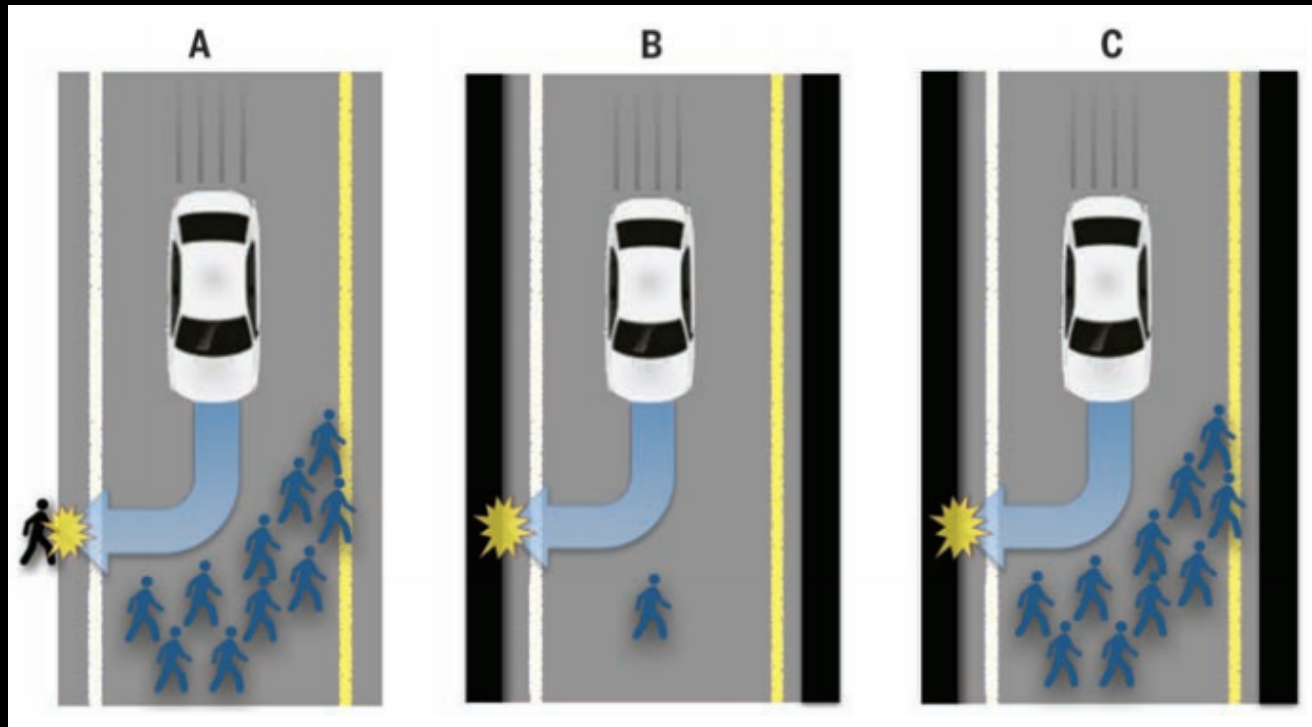


NOVA YORK - A Equifax, empresa de gestão de crédito dos EUA, disse nesta quinta-feira que hackers tiveram acesso a alguns de seus sistemas, potencialmente comprometendo a informação pessoal de cerca de 143 milhões de consumidores nos EUA em um dos maiores e mais ameaçadores ataques contra dados nos últimos anos.

O ataque à Equifax, feito entre maio e julho deste ano, pode ser especialmente prejudicial, dado que as empresas de gestão de crédito ajudam a determinar quais consumidores obtêm acesso ao financiamento e quanto disso é disponibilizado. O ataque também difere de outros, na medida em que os hackers tiveram acesso de uma só vez a várias informações dos consumidores que poderiam tornar mais fácil para os invasores tentar cometer fraudes.

Casos - Dilemas Éticos

- Carro inteligente



Suspenso teste de 'carro inteligente'



00:00 • 20.03.2018



Desde 2017, o Uber colocou nas ruas veículos autônomos, guiados por GPS e sensores instalados no teto (FOTO: UBER)

revolução no setor de transportes.

San Francisco. Pela primeira vez na história, um veículo sem um motorista responsável pela direção, guiado só por um sistema de tecnologia, matou um pedestre em via pública nos EUA, divulgou, ontem, o jornal "New York Times". Uma mulher de 49 anos atravessou a rua fora da faixa de pedestre e foi atropelada. O caso ocorreu, ontem, em Tempe, no Arizona (sudeste dos EUA), envolvendo um carro operado pela Uber, que realiza testes do gênero em algumas cidades, dentro de um ambicioso projeto daquilo que seria uma

No final do ano passado, um taxista flagrou com câmera o momento em que um "carro inteligente", sem motorista, atravessar o sinal vermelho quando uma pessoa iria atravessar a rua.

Além do Uber, outras 20 empresas, como a Google e a Tesla, também colocaram veículos sem motoristas nas ruas da Califórnia desde o ano passado.

DIÁRIO

Casos - Dilemas Éticos

■ Éticamente e agir legalmente nem sempre é o mesmo



Casos - Dilemas Éticos

Cada vez mais encontramos softwares supostamente legítimos e de empresas estabelecidas que fazem coisas altamente questionáveis, como por exemplo:

- Informar o que você fez no seu computador a um servidor central;
- Ler o seu e-mail, armazenado nos servidores deles;

Resumindo...

- Os engenheiros de software devem ser encorajados a melhorar o ambiente ao aderir aos princípios de desenvolvimento sustentável com o objetivo de aumentar a qualidade de vida do público. Além disso devem trabalhar sob um padrão de comportamento profissional que exige adesão aos princípios altos de conduta ética.
- Muitos dos princípios do código de ética para a Engenharia de Software confundem-se com a ética humana e os princípios morais que cada um deve ter da vida em sociedade. (CLARO!!)

Pergunta...

- Você acha que aderir ao código de ética traria algum benefício?



Curiosidade

- <https://exame.abril.com.br/carreira/7-dilemas-eticos-e-como-os-profissionais-reagem-a-eles/>
- Pesquisa investigou como 3.211 pessoas de 45 empresas privadas do Brasil reagiriam em sete casos de dilemas éticos.
- O levantamento mostra que 11% dos entrevistados não agem de acordo com o código de ética das empresas. Outros 69% oscilam de acordo com as circunstâncias, podendo atuar de forma ética e antiética.
- **7 Dilemas apresentados:**
 - 1 Denunciar um ato antiético cometido por um colega**
 - Quando a falta de ética vive na baia ao lado, 56% dos profissionais disseram que somente denunciariam os colegas se fossem incentivados pela empresa.
 - Na divisão por gênero e hierarquia, a pesquisa mostra que mais da metade das mulheres (61%) e dos funcionários (60%) de níveis operacionais hesitariam em delatar o colega antiético.

Curiosidade

2 Conviver com atos antiéticos

- Pouco mais da metade dos participantes da pesquisa disse não ter restrições à convivência com a falta de ética na empresa

3 Adoção de “atalho” antiético para atingir metas

- A pressão por metas pode levar 48% dos participantes do levantamento a escolherem o caminho mais curto e que fuja ao código de ética para cumprir os objetivos estabelecidos. Entre os homens, metade revelou que escolheria o atalho antiético, assim como 53% dos maiores de 34 anos.

4 Furto

- Homens (24%) e não graduados (25%) são mais propensos a furtar valores ou bens materiais consideráveis das organizações. Mas no geral, apenas 18% dos entrevistados admitiram que fariam isso.

Curiosidade

5 Aceitar suborno

- Dependendo da circunstância, quase metade dos homens adultos e não graduados – 43% – aceitaria suborno para dar vantagem a um fornecedor. Levando-se em consideração as respostas de todos os entrevistados, o índice cai para 38%.

6 Receber presentes

- De acordo com a pesquisa, 40% admitiram que beneficiariam um fornecedor em troca de brindes e presentes

7 Usar informações confidenciais em benefício próprio

- Gestores adultos e graduados são a parcela de entrevistados mais propensa a incorrer neste tipo de conduta antiética. Deste grupo, 32% tendem a usar informações secretas em benefício próprio ou de terceiros. Mas, no geral, o índice registrado foi de 28%.

Pontos-Chave

- A engenharia de software é uma disciplina relacionada com todos os aspectos de produção de software.
- Os produtos de software consistem em programas desenvolvidos e documentação associada. Os atributos essenciais do produto são: manutenibilidade, confiança, eficiência e aceitabilidade.
- O processo de software compreende todas as atividades envolvidas no desenvolvimento de produtos de software. As atividades básicas são especificação, desenvolvimento, validação e evolução de software.
- Métodos são meios organizados de produção de software. Eles incluem sugestões para o processo a ser seguido, as notações a serem usadas, modelos de sistemas a serem desenvolvidos, regras que regem estes modelos e diretrizes para o projeto.

Pontos-Chave

- Ferramentas CASE são sistemas de software projetados para apoiar as atividades rotineiras no processo de software, tais como edição de diagramas de projeto, verificação da consistência de diagramas e rastreabilidade de testes de programa realizados.
- Engenheiros de software têm responsabilidades com a profissão de engenharia e a sociedade. Eles não devem se preocupar apenas com assuntos técnicos.
- Sociedades profissionais publicam códigos de conduta que definem os padrões de comportamento esperados de seus membros.

VOCÊS TÊM UM SISTEMA DE ESTOQUE PRA FAZER ATÉ O DIA 1 DE MAIO! GRRRRRRRR!



Um mês antes...

PERAÍ, QUAL O PRAZO DESSE SISTEMA? UM DIA?



4 de Abril

GRRRR! UM MÊS! LEVOU MUITO TEMPO PRA FAZER ESSA DOCUMENTAÇÃO! AGORA É A FASE DE IMPLEMENTAÇÃO!



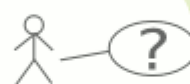
18 de Abril

NESSE CASO DE USO, O USUÁRIO ACESSA O SISTEMA...E DEPOIS...?



22 de Abril

46 PÁGINAS APENAS? GRRRR! TEMOS UM SISTEMA COMPLEXO PRA DOCUMENTAR, NÃO UMA CALCULADORA EM JAVASCRIPT!



29 de Abril

O USUÁRIO ACESSA O SISTEMA.....E DEPOIS? E DEPOIS? O QUE ESSE MALDITO USUÁRIO FAZ?! EU NÃO SEI!!!



DEIXO ESSE DIAGRAMA COM A MESMA COR DO ANTERIOR?



1 de Maio

CONVENCÍ O CLIENTE A ADIAR O PRAZO PRA SEMANA QUE VEM! É O SUFICIENTE PRA FINALIZAR O PROJETO? GRRRR...



Dúvidas?



Parallel...

- What does a Software Engineer do?
 - <http://www.owenpellegrin.com/blog/life/what-does-a-software-engineer-do/>

Referências

Slides do capítulo 1 do livro Engenharia de Software
(Ian Sommerville)

Slides do capítulo 1 do livro Engenharia Moderna
(Marco Tulio)