

Engenharia de Software

01 - Fundamentos e Ética

Prof. Msc. Francisco de Assis de Souza Rodrigues

UniFAP
CENTRO UNIVERSITÁRIO PARAÍSO

"O coração do entendido adquire conhecimento, e o ouvido dos sábios busca a sabedoria."

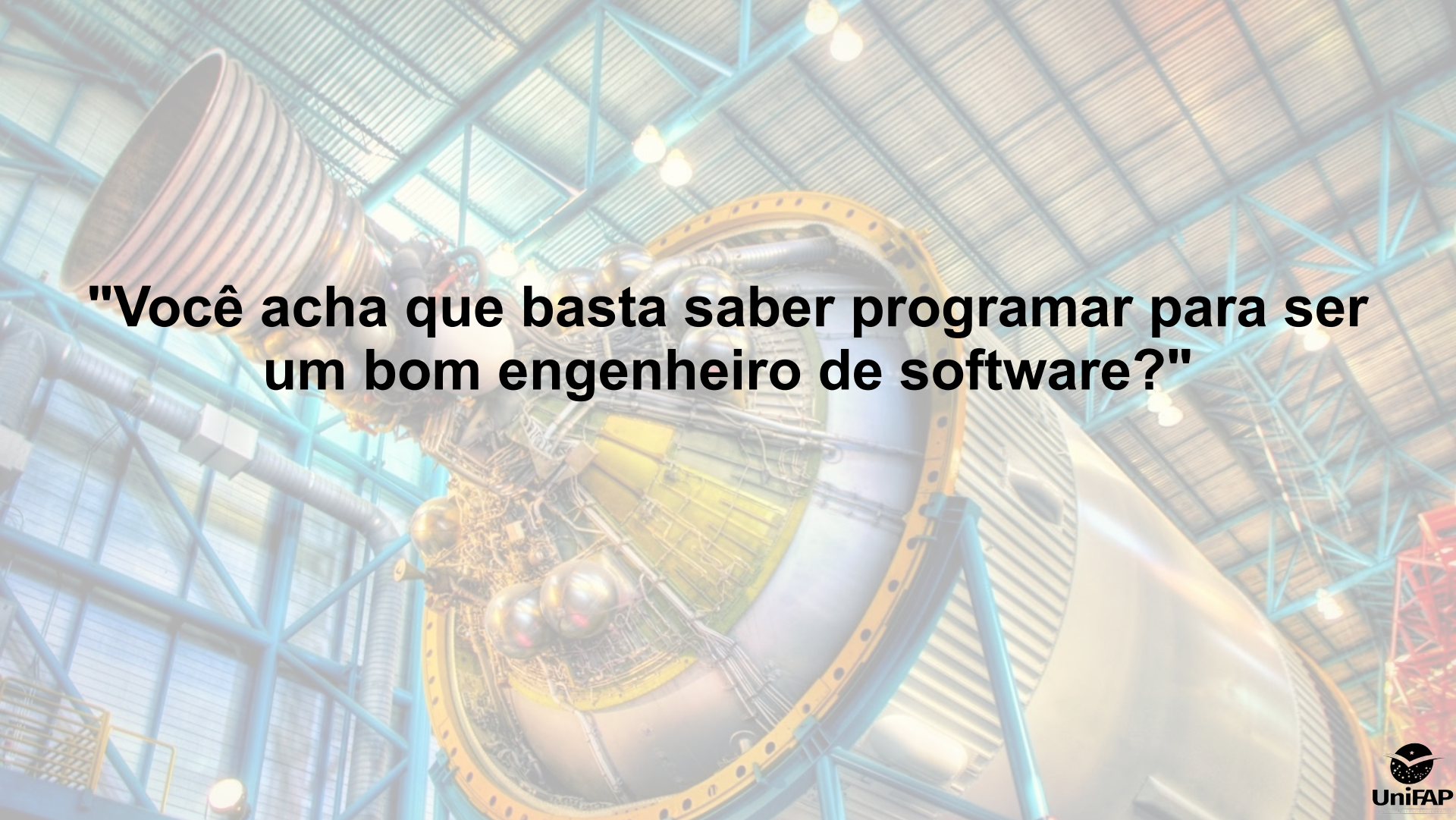
Provérbios 18:15

Ter curiosidade e dedicação aos estudos é sinal de maturidade. Quem procura aprender se torna mais preparado para a vida.

🎯 Objetivos

1. Apresentar o conceito de Engenharia de Software (ES) e sua importância.
2. Diferenciar ES de programação.
3. Discutir a responsabilidade profissional e os princípios éticos.
4. Analisar dilemas éticos e suas implicações no desenvolvimento de software.
5. Aplicar conceitos em um debate de casos reais.

UniFAP


"Você acha que basta saber programar para ser um bom engenheiro de software?"

📜 Breve Histórico da Engenharia de Software

Anos 1950–1960 – Origem do Problema

- Computadores eram grandes, caros e usados principalmente por governos e indústrias.
- Programação era feita por poucas pessoas, muitas vezes sem métodos formais.
- Com a crescente complexidade dos sistemas, começaram a surgir **erros graves, atrasos e estouros de orçamento**.
- Faltava padronização: cada projeto seguia um estilo próprio.

UniFAP

📜 Breve Histórico da Engenharia de Software

1968–1969 – A “Crise do Software”

- Termo cunhado durante conferências da OTAN (NATO Software Engineering Conferences).
- “Crise” = dificuldades em:
  - Entregar no prazo.
  - Cumprir orçamento.
  - Garantir qualidade e confiabilidade.
- Muitos sistemas **falhavam ou nunca eram concluídos**.
- A solução proposta: tratar desenvolvimento de software como **engenharia**, não apenas programação.

UniFAP

📜 Breve Histórico da Engenharia de Software

Anos 1970–1980 – Consolidação da Disciplina

- Surgem métodos formais de desenvolvimento (ex.: **modelo cascata**, metodologias estruturadas).
- Maior foco em **documentação, testes e manutenção**.
- Softwares começam a ser vistos como **ativos estratégicos** das empresas.

📜 Breve Histórico da Engenharia de Software

Anos 1990–2000 – Mudança de Paradigma

- Popularização da internet.
- Ciclos de desenvolvimento mais curtos (ex.: **iterativo**, **espiral**).
- Surgimento da **engenharia de requisitos** como área essencial.
- Preocupações com **usabilidade, segurança e escalabilidade** crescem.

UniFAP

📜 Breve Histórico da Engenharia de Software

2001 em diante – Engenharia de Software Ágil e Global

- Manifesto Ágil (2001) promove colaboração, entregas rápidas e adaptação constante.
- Desenvolvimento distribuído e equipes globais.
- Integração contínua, DevOps, microsserviços e foco em qualidade contínua.

UniFAP

📜 Breve Histórico da Engenharia de Software

💡 Por que surgiu como disciplina?

- **Necessidade de método:** Sem processos claros, os projetos falhavam com frequência.
- **Complexidade crescente:** Softwares passaram a controlar sistemas críticos (aviões, hospitais, bancos).
- **Confiabilidade e segurança:** Erros passaram a ter **impacto real na vida das pessoas**.
- **Custo e prazo:** Projetos mal geridos estouravam orçamento e atrasavam anos.
- **Profissionalização:** Assim como a engenharia civil ou mecânica, o desenvolvimento precisava de **padrões, ética e boas práticas**.

UniFAP

Introdução à Engenharia de Software

"Engenharia de software é uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software..."

Sommerville (10ª ed.)

"Disciplina da engenharia" → envolve métodos, processos e princípios sistemáticos, assim como outras áreas da engenharia (civil, elétrica, mecânica).

"Todos os aspectos" → não apenas codificação, mas também análise, projeto, testes, documentação, implantação, manutenção e evolução.

"Produção de software" → considera desde a ideia inicial até o descarte/obsolescência.

Enfoque prático e humano → engenharia de software lida tanto com tecnologia quanto com pessoas e equipes.

UniFAP

Diferenças entre Engenharia de Software, Ciência da Computação e Programação

Área	Foco Principal	Exemplo de Atividade
Ciência da Computação	Conceitos teóricos, algoritmos, fundamentos matemáticos e modelos computacionais.	Criar um novo algoritmo de ordenação mais eficiente.
Engenharia de Software	Processos, métodos, ferramentas e práticas para projetar, construir, testar e manter sistemas de software de alta qualidade.	Planejar e desenvolver um sistema de controle de tráfego urbano, garantindo confiabilidade e segurança.
Programação	Implementação de soluções computacionais em uma ou mais linguagens de programação.	Codificar a tela de login de um aplicativo em Java ou Python.

- Saber programar é **necessário**, mas não é **suficiente** para ser engenheiro de software.
- Engenharia de Software integra **técnica, organização, gestão, qualidade e ética**.

UniFAP

Características de um bom software

Funcionalidade

- O software faz **o que o usuário precisa** e de forma correta.

- Atende aos requisitos funcionais especificados.

Exemplo: Um aplicativo bancário que realmente permite transferências, pagamentos e consultas, sem erros ou recursos incompletos.

UniFAP

Características de um bom software

Confiabilidade

- O sistema apresenta comportamento previsível, mesmo em condições adversas.

- Evita falhas graves e lida com erros de forma controlada.

Exemplo: Um sistema médico que nunca apresenta diagnósticos trocados devido a falhas de software.

UniFAP

Características de um bom software

Usabilidade

- Fácil de aprender, usar e lembrar.

- Experiência agradável para o usuário, reduzindo erros operacionais.

Exemplo: Interface clara e intuitiva no aplicativo de transporte, sem menus confusos.

Características de um bom software

Eficiência

- Faz uso adequado de recursos (tempo de processamento, memória, rede).
- Evita consumo excessivo de CPU/bateria.

Exemplo: Um app que abre em 2 segundos e não trava, mesmo em celulares modestos.

Características de um bom software

Manutenibilidade

- Fácil de corrigir, adaptar e melhorar.
- Código organizado, modular e bem documentado.

Exemplo: Um sistema em que adicionar um novo relatório não exige reescrever toda a aplicação.

UniFAP

Características de um bom software

Portabilidade

- Capacidade de rodar em diferentes plataformas e ambientes com poucas ou nenhuma modificação.

Exemplo: Um jogo que funciona tanto no Windows quanto no macOS e Linux, sem refazer todo o código.

UniFAP

Desenvolvimento Profissional em Engenharia de Software

O papel de um profissional de Engenharia de Software vai muito além de dominar linguagens de programação. Envolve a capacidade de integrar conhecimentos técnicos, habilidades interpessoais e postura ética para entregar sistemas que sejam úteis, confiáveis e socialmente responsáveis.

Competências Técnicas e Comportamentais

Competências técnicas (hard skills) – são conhecimentos mensuráveis, que podem ser aprendidos por meio de estudo formal e prática:

- **Análise de requisitos:** entender e documentar necessidades do cliente e usuários.
- **Modelagem de sistemas:** uso de diagramas, UML, BPMN, especificações formais.
- **Programação e arquitetura de software:** domínio de linguagens, frameworks, padrões de projeto.

UniFAP

Competências Técnicas e Comportamentais

Competências técnicas (hard skills)

- **Testes e qualidade:** testes unitários, integração, validação, verificação e auditorias de código.
- **Manutenção e evolução:** refatoração, integração contínua, deploy automatizado.
- **Segurança:** práticas de desenvolvimento seguro, criptografia, autenticação e proteção de dados.

Competências Técnicas e Comportamentais

Competências comportamentais (soft skills) – são habilidades interpessoais e atitudes que permitem aplicar as técnicas de forma efetiva:

- **Comunicação clara:** explicar conceitos técnicos para públicos diversos.
- **Pensamento crítico:** analisar problemas, avaliar soluções e prever impactos.
- **Colaboração:** trabalhar de forma construtiva com outros desenvolvedores, analistas, clientes e usuários finais.

Competências Técnicas e Comportamentais

Competências comportamentais (soft skills)

- **Gestão de tempo:** priorizar tarefas e cumprir prazos realistas.
- **Adaptabilidade:** lidar com mudanças de requisitos e novas tecnologias.
- **Responsabilidade ética:** tomar decisões pensando no impacto para usuários e sociedade.

Muitas vezes, falhas de projeto não ocorrem por “erro de código”, mas por falhas humanas — má comunicação, pressa, negligência ou descaso com a ética.

Ciclo de Vida de Aprendizado

O aprendizado na área de ES é contínuo. Podemos imaginar um **ciclo em 3 eixos**:

1. **Conhecimento Técnico**
2. **Prática Profissional**
3. **Reflexão Ética**

Ciclo de Vida de Aprendizado

Conhecimento Técnico

- Aprender fundamentos (algoritmos, estruturas de dados, arquitetura).
- Atualizar-se constantemente sobre novas linguagens, frameworks, ferramentas e padrões.

Ciclo de Vida de Aprendizado

Prática Profissional

- Participar de projetos reais ou simulados (como PIs, estágios, hackathons).
- Experimentar diferentes papéis no ciclo de desenvolvimento (analista, programador, tester, líder).

Ciclo de Vida de Aprendizado

Reflexão Ética

- Questionar: “Minha solução atende a todos com segurança, acessibilidade e respeito à privacidade?”
- Avaliar dilemas e tomar decisões responsáveis.
- Consultar códigos de ética e boas práticas.

Esse ciclo nunca se fecha totalmente — cada nova tecnologia, projeto ou desafio reinicia o processo.

Trabalho em Equipe e Comunicação

A Engenharia de Software moderna é **colaborativa por natureza**. Poucos sistemas são desenvolvidos por uma única pessoa; normalmente, há times multidisciplinares.

Isso exige:

- Comunicação estruturada
- Escuta ativa e empatia
- Resolução de conflitos
- Coordenação e papéis definidos

UniFAP

Trabalho em Equipe e Comunicação

Comunicação estruturada:

- Reuniões curtas e objetivas (daily meetings, follow-ups).
- Uso de ferramentas de colaboração (Git, Jira, Trello, Slack, Teams).
- Documentação clara e acessível.

Trabalho em Equipe e Comunicação

Escuta ativa e empatia:

- Entender as necessidades e limitações dos outros membros do time.
- Respeitar diferentes pontos de vista e formas de trabalho.

Trabalho em Equipe e Comunicação

Resolução de conflitos:

- Discutir ideias, não pessoas.
- Focar em fatos e dados.
- Buscar consenso ou compromisso equilibrado.

Trabalho em Equipe e Comunicação

Coordenação e papéis definidos:

- Scrum Master, Product Owner, Dev, QA etc.

- Saber quando liderar e quando seguir instruções.

Segundo o **Sommerville**, a comunicação é tão vital que erros de entendimento entre equipe e cliente podem gerar mais prejuízo do que erros técnicos no código.

UniFAP

Ética Profissional

O que é Ética Profissional em Engenharia de Software?

Definição:

Conjunto de princípios e valores que orientam o comportamento e as decisões dos profissionais de software para garantir que seu trabalho seja responsável, seguro, honesto e benéfico à sociedade.

UniFAP

Ética Profissional

Importância:

- Evitar danos a pessoas, empresas e sociedade.
- Preservar a confiança dos usuários e clientes.
- Manter a credibilidade da profissão.
- Prevenir problemas legais.

Ética Profissional

Princípios Gerais

O **Código de Ética e Conduta Profissional da ACM/IEEE** é uma referência global para orientar engenheiros de software.

Alguns princípios essenciais:

…

Ética Profissional

Princípios Gerais do **Código de Ética e Conduta Profissional da ACM/IEEE**:

Contribuir para o bem-estar da sociedade e da qualidade de vida.

- O software deve trazer benefícios e evitar danos.

Ética Profissional

Princípios Gerais do **Código de Ética e Conduta Profissional da ACM/IEEE**:

Evitar danos aos outros.

- Antecipar e prevenir problemas que possam prejudicar usuários, empresas ou o público.

Ética Profissional

Princípios Gerais do **Código de Ética e Conduta Profissional da ACM/IEEE**:

Ser honesto e transparente.

- Comunicar limitações, riscos e custos de forma clara.

Ética Profissional

Princípios Gerais do **Código de Ética e Conduta Profissional da ACM/IEEE**:

Manter a confidencialidade.

- Proteger informações privadas obtidas no trabalho.

Ética Profissional

Princípios Gerais do **Código de Ética e Conduta Profissional da ACM/IEEE**:

Respeitar propriedade intelectual.

- Cumprir licenças, patentes e direitos autorais.

Ética Profissional

Princípios Gerais do **Código de Ética e Conduta Profissional da ACM/IEEE**:

Melhorar continuamente competência e qualidade.

- Atualizar conhecimentos e aplicar boas práticas.

Ética Profissional

Princípios Gerais do **Código de Ética e Conduta Profissional da ACM/IEEE**:

Aceitar responsabilidade pelo próprio trabalho.

- Não culpar terceiros sem fundamento; assumir erros e corrigi-los.

Ética Profissional

Princípios Gerais do **Código de Ética e Conduta Profissional da ACM/IEEE**:

Respeitar leis e regulamentos.

- Seguir legislação nacional e internacional pertinente.

Exemplos de Responsabilidades Éticas

a) Privacidade dos Usuários

- **Descrição:** Garantir que dados pessoais sejam coletados, armazenados e processados de forma segura, com consentimento informado.

Exemplo real: Escândalo da *Cambridge Analytica*, que usou dados de milhões de usuários do Facebook sem autorização.

- **Boas práticas:**
  - Implementar políticas claras de privacidade.
  - Criptografar dados sensíveis.
  - Cumprir leis como LGPD (Brasil) e GDPR (Europa).

UniFAP

Exemplos de Responsabilidades Éticas

b) Segurança de Sistemas

- **Descrição:** Prevenir vulnerabilidades que possam ser exploradas por atacantes, causando danos financeiros, sociais ou à vida humana.

Exemplo real: Ataque ransomware *WannaCry* (2017) que paralisou hospitais e empresas.

- **Boas práticas:**
  - Revisar código e realizar testes de segurança.
  - Manter software atualizado.
  - Implementar autenticação robusta.

UniFAP

Exemplos de Responsabilidades Éticas

c) Propriedade Intelectual

- **Descrição:** Respeitar direitos autorais, patentes e licenças de software.

Exemplo real: Uso indevido de bibliotecas de código sem verificar licença, gerando processos milionários.

- **Boas práticas:**
  - Conferir licenças antes de usar bibliotecas.
  - Citar fontes e autores de códigos e ideias.
  - Evitar plágio de código e documentação.

UniFAP

Exemplos de Responsabilidades Éticas

d) Impacto Social de Sistemas

- **Descrição:** Avaliar consequências sociais, econômicas e culturais do software produzido.

Exemplo real: Algoritmos de recomendação que amplificam fake news ou discursos de ódio.

- **Boas práticas:**
  - Monitorar efeitos não intencionais do sistema.
  - Criar mecanismos de mitigação de impactos negativos.

UniFAP

Consequências de Decisões Antiéticas

Individuais:

- Perda de reputação e empregabilidade.
- Processos judiciais e multas.
- Responsabilidade criminal em casos graves (ex.: falhas que resultam em mortes).

Consequências de Decisões Antiéticas

Para a organização:

- Multas por descumprir leis como LGPD/GDPR.
- Danos à marca e perda de clientes.
- Custos elevados com correção de danos.

Consequências de Decisões Antiéticas

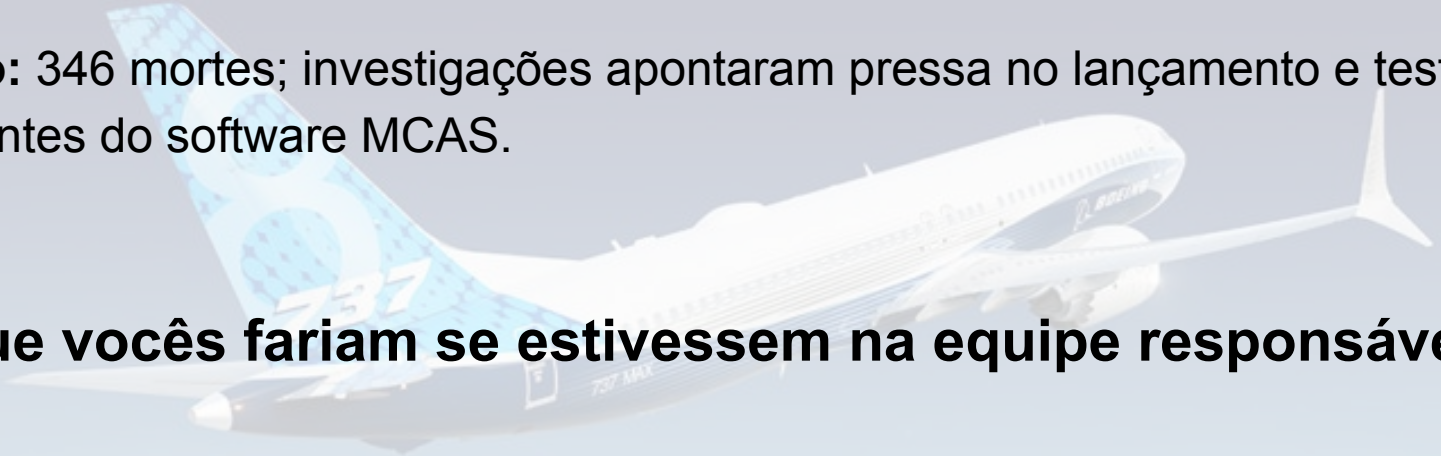
Para a sociedade:

- Perda de confiança em sistemas digitais.
- Aumento de crimes cibernéticos.
- Prejuízos financeiros e sociais amplos.

"Boeing 737 MAX: falha de software ligada a dois acidentes fatais"

Ano: 2018-2019

Impacto: 346 mortes; investigações apontaram pressa no lançamento e testes insuficientes do software MCAS.



"O que vocês fariam se estivessem na equipe responsável?"

… a decisão correta seria a mais fácil de tomar sob pressão?

"Equifax paga US$ 700 milhões por vazamento de dados de 147 milhões de pessoas"

Ano: 2017

Impacto: Informações financeiras e pessoais expostas devido a vulnerabilidade não corrigida em tempo hábil.

"O que vocês fariam se estivessem na equipe responsável?"

… a decisão correta seria a mais fácil de tomar sob pressão?

UniFAP

"Facebook e Cambridge Analytica: escândalo de uso indevido de dados de 87 milhões de usuários"

Ano: 2018

Impacto: Uso de dados pessoais sem consentimento para influenciar eleições nos EUA e no Brexit.

"O que vocês fariam se estivessem na equipe responsável?"

… a decisão correta seria a mais fácil de tomar sob pressão?

UniFAP

"Therac-25: falha de software em equipamento médico causa overdose de radiação"

Ano: 1985–1987

Impacto: Pelo menos 6 pacientes receberam doses letais; erro de concorrência no código e ausência de mecanismos de segurança.

"O que vocês fariam se estivessem na equipe responsável?"

… a decisão correta seria a mais fácil de tomar sob pressão?

UniFAP

"Explosão do foguete Ariane 5 devido a falha de conversão de variável"

Ano: 1996
Impacto: Foguete destruído segundos após lançamento; erro ao converter número de 64 bits para 16 bits.

"O que vocês fariam se estivessem na equipe responsável?"

… a decisão correta seria a mais fácil de tomar sob pressão?

UniFAP

Caso	Tipo de Falha	Princípios ACM/IEEE Violados	Possíveis Ações Preventivas
Facebook & Cambridge Analytica (2018)	Ética + Técnica	Contribuir para o bem público; Evitar danos; Ser honesto e transparente; Respeitar a privacidade	Implementar consentimento claro; Cumprir legislações (LGPD/GDPR); Transparência sobre uso de dados
Boeing 737 MAX (2018–2019)	Técnica + Ética	Evitar danos; Garantir qualidade; Assumir responsabilidade	Testes mais extensos; Treinamento de pilotos; Revisão independente do software
Equifax (2017)	Técnica + Ética	Evitar danos; Manter competência técnica; Proteger informações privadas	Correção imediata de vulnerabilidades; Monitoramento proativo de segurança
Therac-25 (1985–1987)	Técnica + Ética	Evitar danos; Garantir qualidade; Assumir responsabilidade	Redundância de segurança; Testes exaustivos; Auditoria independente de software
Ariane 5 (1996)	Técnica	Garantir qualidade; Manter competência técnica	Revisão de código crítico; Simulação completa em ambiente realista; Tes de conversão de dados

UniFAP

🛠️ **Parte Prática –** Atividade em Grupo

Passo 1: Dividir a turma em 4-5 grupos.

🛠️ **Parte Prática –** Atividade em Grupo

Passo 2: Cada grupo recebe um dilema ético:

1. Vigilância Disfarçada
2. Falha Crítica Ignorada
3. Lançamento às Pressas
4. Licença Esquecida
5. Algoritmo Preconceituoso
6. Relatórios Manipulados
7. Risco à Privacidade Infantil

https://docs.google.com/document/d/1lSayF9u_2qGnlzW0ysJmCOVb9-LUPy8DWKhHwQAgXK8/edit?usp=sharing

UniFAP

🛠️ **Parte Prática –** Atividade em Grupo

Passo 3: Analisar:

- Qual é o problema ético?
- Quais possíveis consequências?
- Quais princípios do código de ética estão envolvidos?
- Como agir profissionalmente?

🛠️ **Parte Prática –** Atividade em Grupo

Passo 4: Apresentar brevemente cada turma (1 à 5 min por grupo).

"Se você tivesse que lançar um software amanhã que pode salvar vidas, mas ainda não foi testado, o que faria?"

UniFAP

Bibliografia

SOMMERVILLE, Ian. Engenharia de software. 10. ed. São Paulo, SP: Pearson, 2018. E-book. Disponível em: https://plataforma.bvirtual.com.br. Acesso em: 15 ago 2025.

https://plataforma.bvirtual.com.br/Acervo/Publicacao/168127



Referências

CHATGPT. Expansão conceitual sobre integração entre COBIT, ITIL e PMBOK. OpenAI, 2025. Disponível em: https://chat.openai.com/. Acesso em: 11 ago. 2025.


UniFAP


OBRIGADO
Arigato
Spasibo
Obrigado
Dankon
Maake
Multumesc
Ua Tsaug Rau Koj
Mochchakkeram
Kiitos
Merci
Raibh Maith Agat
Chokrane
Grazie
Terma Kasih
Asante
Mamana
Kia Ora
THANK YOU
Jusparxar
Welalin
Kiitos
Grazie
Obrigado
Arigato
Spasibo
Merci
Chokrane
Grazie
Kiitos
Multumesc
Terma Kasih
Maake
Mochchakkeram
Kiitos
Grazie
Obrigado
Merci
Raibh Maith Agat
Kiitos
Multumesc
Kia Ora
Grazie
Kiitos
Obrigado
Grazie
Chokrane
Merci
Mochchakkeram
Grazie
Kiitos
Obrigado
Grazie
Multumesc
Merci
Mochchakkeram
Chokrane
Grazie