

TÉCNICAS DE PROGRAMAÇÃO PARA CIÊNCIA DE DADOS

Fatec 2025

Prof. Fábio de Paula Santos

fabio.santos@fatec.sp.gov.br

Prof. Fábio de Paula Santos

www.linkedin.com/in/fabiopsantos

- Doutor em Computação pela Universidade Mackenzie
- Mestre em Educação pela UNISO
- graduado em Processamento de Dados pela Fatec-SO
- com 6 especializações:
 - Mineração de dados complexos - Unicamp
 - Vocational Teacher Education pela HAMK University of Applied Science (Finlândia)
 - Gestão de Ambiente EAD pela Univ. Federal Fluminense
 - Administração de Empresas pela FGV
 - Engenharia de Software pela Univ. São Judas Tadeu
 - Docência no Ensino Superior pela Academia de Ensino Superior
- 23 anos de experiência profissional na área de gestão e desenvolvimento de sistemas de informação (ERP)
- 26 anos na docência (sendo 20 anos dedicados ao ensino superior)
- Cientista de dados do Centro Paula Souza

Competências

- Analisar e examinar dados, identificando padrões, tendências e insights relevantes
- Programar utilizando linguagens como Python, R ou SQL, para manipulação e processamento de dados
- Demonstrar capacidade de resolver problemas complexos e propor soluções criativas e inovadoras

Objetivos

- Utilizar os princípios da programação orientada a objetos, como encapsulamento, herança, polimorfismo e abstração
- Implementar classes e objetos para modelar sistemas complexos
- Utilizar herança e polimorfismo para criar estruturas de dados e algoritmos mais flexíveis e reutilizáveis
- Identificar e aplicar padrões de projeto (Design Patterns) para resolver problemas comuns de desenvolvimento de software
- Utilizar padrões de criação, estruturais e comportamentais para melhorar a modularidade, flexibilidade e manutenibilidade do código
- Utilizar bibliotecas e frameworks gráficos para criar interfaces interativas e intuitivas
- Implementar elementos de interface, como botões, menus, caixas de texto e gráficos
- Gerenciar eventos e interações do usuário com a GUI
- Implementar boas práticas de programação para garantir a robustez e escalabilidade do código
- Utilizar tratamento de exceções e validação de entrada para lidar com erros e garantir a estabilidade da aplicação
- Implementar testes de unidade para verificar a funcionalidade correta do código
- Utilizar recursos avançados da linguagem de programação, como programação assíncrona, threads e concorrência
- Manipular arquivos, bancos de dados e outros recursos externos
- Trabalhar em equipe, colaborando com outros desenvolvedores para desenvolver projetos de software
- Utilizar ferramentas de controle de versão e metodologias de desenvolvimento colaborativo

Ementa

- Classes, Objeto, Encapsulamento, Herança, Polimorfismo
- Declaração de Classes e Objetos
- Classe Abstrata. Métodos. Sobrecarga de Métodos. Conceitos de Herança múltipla
- Modificadores de acesso. Construtores. Manipulação de Exceções
- Conceitos e aplicações de arquitetura em Camadas
- Uso de Interface Gráfica
- Padrões de projeto Orientados a Objetos
- Desenvolvimento utilizando banco de dados. Persistência de dados utilizando
- frameworks de interface gráfica
- Padrões arquiteturais: MVC e MVP
- Desenvolvimento Dirigido a Testes (TDD)

Bibliografia

- FURGERI, Sérgio. **Programação Orientada a Objetos**: Conceitos e técnicas. São Paulo: Erica.
- FREEMAN, Eric.; FREEMAN Eisabeth. **Use a Cabeça!** Padrões de Projetos. 2 ed. Rio de Janeiro:Alta Books, 2007.
- BECK, Kent. **TDD** - Desenvolvimento Guiado Por Testes. Porto Alegre: Bookman, 2010.

O'REILLY®

Engenharia de Software para Cientistas de Dados

De Notebooks a Sistemas Escaláveis



novatec

Catherine Nelson

novatec.com.br

Avaliação

- Exercícios e atividades semanais (peso 4)
- Projeto de aplicação (peso 6)
- Entregas somente pelo MS-TEAMS

Sobre Programar



| **Prós:** O computador faz exatamente
| o que você diz para ele fazer.



| **Contra:** O computador faz exatamente
| o que você diz para ele fazer."

O que é Software?

Me after my first
program in Python
`print "Hello world"`



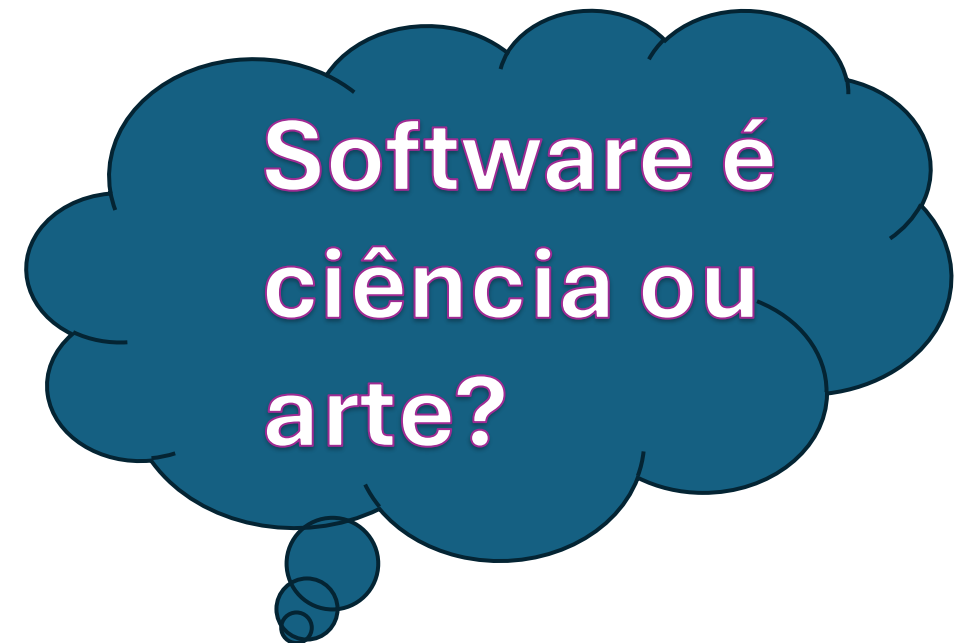
O que é software?

Software é ciência ou arte?

- Programas de computador.
- Entidade abstrata.
- Ferramentas pelas quais:
 - exploramos os recursos do hardware
 - executamos determinadas tarefas
 - resolvemos problemas
 - interagimos com a máquina
 - tornamos o computador operacional

O que é software?

- Programas de computador.
- Entidade abstrata.
- Ferramentas pelas quais:
 - exploramos os recursos do hardware
 - executamos determinadas tarefas
 - resolvemos problemas
 - interagimos com a máquina
 - tornamos o computador operacional



O que é software?

- Conceito mais amplo que inclui também:
 - Instruções que executam uma função desejada.
 - Estrutura de dados para manipular informação.
 - Documentos para desenvolver, operar e manter os programas.

O que é software?

- Produto que os profissionais de software constroem e, depois, mantêm ao longo do tempo.
- Abrange programas que executam em computadores de qualquer tamanho e arquitetura, conteúdo que é apresentado ao programa a ser executado e documentos tanto em forma impressa quanto virtual que combinam todas as formas de mídia

O que é software?

- Produtos de software podem ser desenvolvidos para um cliente particular ou para um mercado geral.
- Produtos de software podem ser:
 - Genéricos – desenvolvidos para serem vendidos para uma grande variedade de clientes, por exemplo, softwares para PC, tais como Excel e Word.
 - Personalizados – desenvolvidos para um único cliente de acordo com as suas especificações.
- Um software novo pode ser criado através do desenvolvimento de novos programas, da configuração de sistemas de software genéricos ou da reutilização de um software existente.

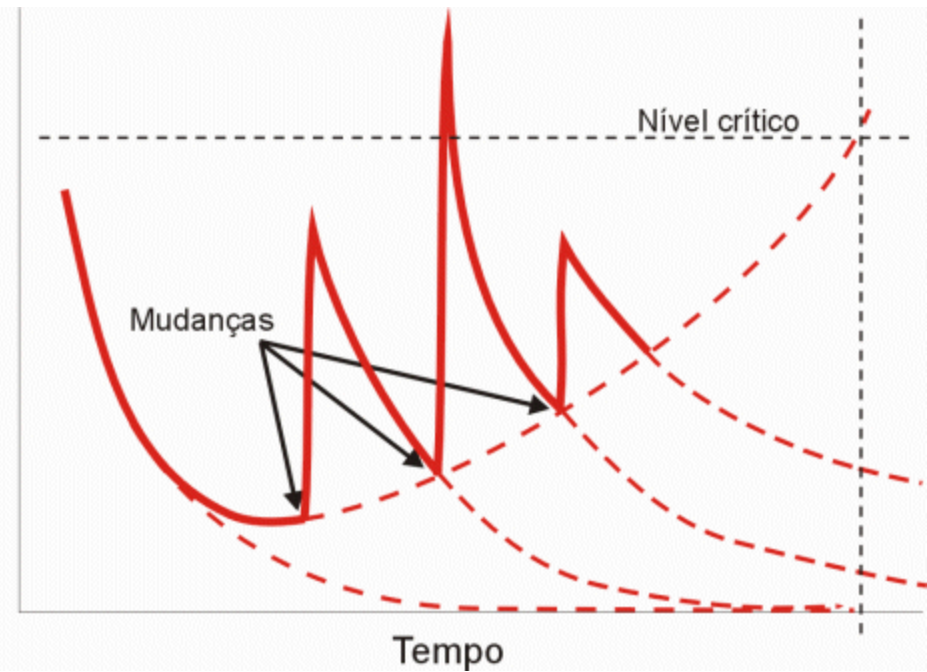
Características do software

- Software não é um elemento físico, é um elemento lógico (não tem propriedades físicas, como visualizar, medir ...)
- requer uma abstração maior
- o produto final é diferente do inicial
- o Software não pode ser manufaturado
- Custos estão concentrados no desenvolvimento e não na manufatura.
- o processo de gerenciamento é diferente;
- o relacionamento entre as pessoas é diferente;

Características do Software



Curva de falhas para o hardware



Curva de falhas para o software

o software não se desgasta com o uso; mas deteriora-se

Dificuldades no desenvolvimento

- Saber o que o software deve fazer : quais os requisitos (abstração)
- Ferramentas; linguagem; Sist. Operacionais.
- Tempo e custos elevados de desenvolvimento.
- Prever falhas (antes de entregar).
- Tratar manutenção e versões.
- Produtividade não cresce com a demanda de serviços.

Engenharia de software

- A Engenharia de Software surgiu nos anos 70 numa tentativa de contornar a *crise do software* e dar um tratamento de *engenharia* ao desenvolvimento de sistemas de software complexos.
- Sistema de software complexo :
 - conjunto de componentes abstratos de software (estruturas de dados e algoritmos) encapsulados na forma de procedimentos, funções, módulos, objetos ou agentes e interconectados entre si, compondo a arquitetura do software, que deverão ser executados em sistemas computacionais.

Engenharia de software

- Estudo ou aplicação de uma abordagem sistemática, disciplinada e baseada em métodos quantitativos para o desenvolvimento, manutenção e operação de sistemas de software
- Engloba técnicas e práticas de gerência de projetos, linguagens de programação, base de dados, ferramentas, plataformas, bibliotecas, padrões e processos

Engenharia de software

- Os fundamentos científicos para a engenharia de software envolvem o uso de modelos abstratos e precisos que permitem ao engenheiro especificar, projetar, implementar e manter sistemas de software, avaliando e garantido suas qualidades.
- A engenharia de software deve oferecer mecanismos para se planejar e gerenciar o processo de desenvolvimento.

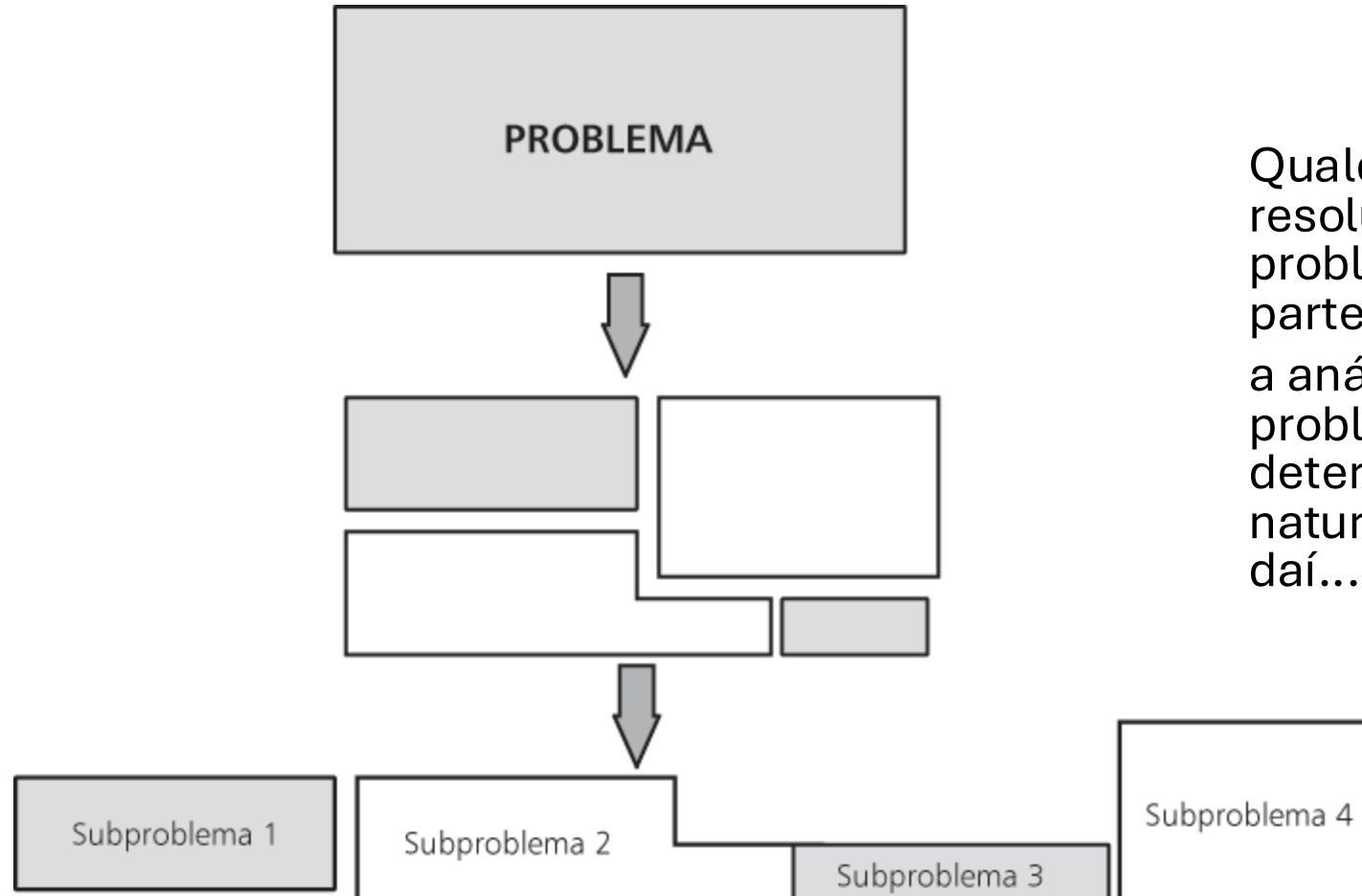
Engenharia de software

- Método ou técnica: procedimento para a produção de um resultado
- Ferramenta: instrumento ou sistema automatizado para realizar alguma coisa.
- Procedimento: receita de combinação de ferramentas e técnicas
- Paradigma: estilo de fazer algo

Engenharia de Software

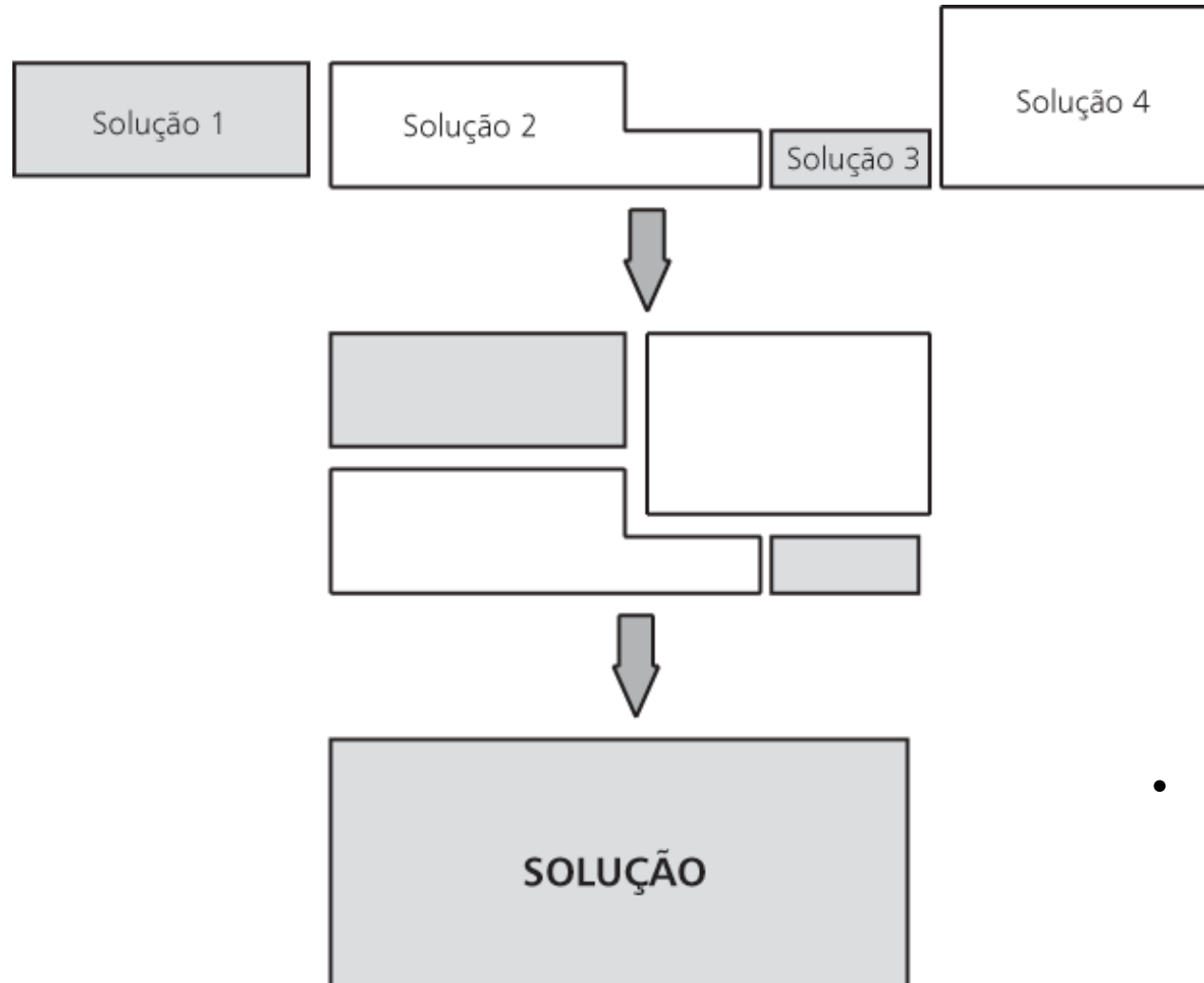
- As economias de TODAS as nações desenvolvidas são dependentes de software.
- Cada vez mais sistemas são controlados por software.
- A engenharia de software se dedica às teorias, métodos e ferramentas para desenvolvimento de software profissional
- Os dispêndios com software representam uma fração significativa do PIB em todos os países desenvolvidos.

Análise x Síntese



Qualquer técnica de resolução de problemas deve ter 2 partes:
a análise do problema para determinar a sua natureza e, a partir daí...

Análise x Síntese



- ... A síntese da solução com base em nossa análise.

Construindo uma casa

- Identificar e analisar os requisitos (tipo de telhado)
- Produzir e documentar todo o projeto (planta baixa, desenhos arquitetônicos)
- Detalhar as especificações (para que o encanador saiba onde integrar)
- Identificar e projetar os componentes
- Construir cada componente
- Testar cada componente
- Integrar os componentes
- Fazer as modificações finais
- Manutenção contínua

Construindo uma casa x Construindo um software

- Identificar e analisar os requisitos
 - Produzir e documentar todo o projeto
 - Detalhar as especificações
 - Identificar e projetar os componentes
 - Construir cada componente
 - Testar cada componente
 - Integrar os componentes
 - Fazer as modificações finais
 - Manutenção contínua
- Análise e definição dos requisitos
 - Projeto do sistema
 - Projeto do programa
 - Escrever os programas
 - Testes das unidades
 - Teste de integração
 - Teste do sistema
 - Entrega do sistema
 - Manutenção

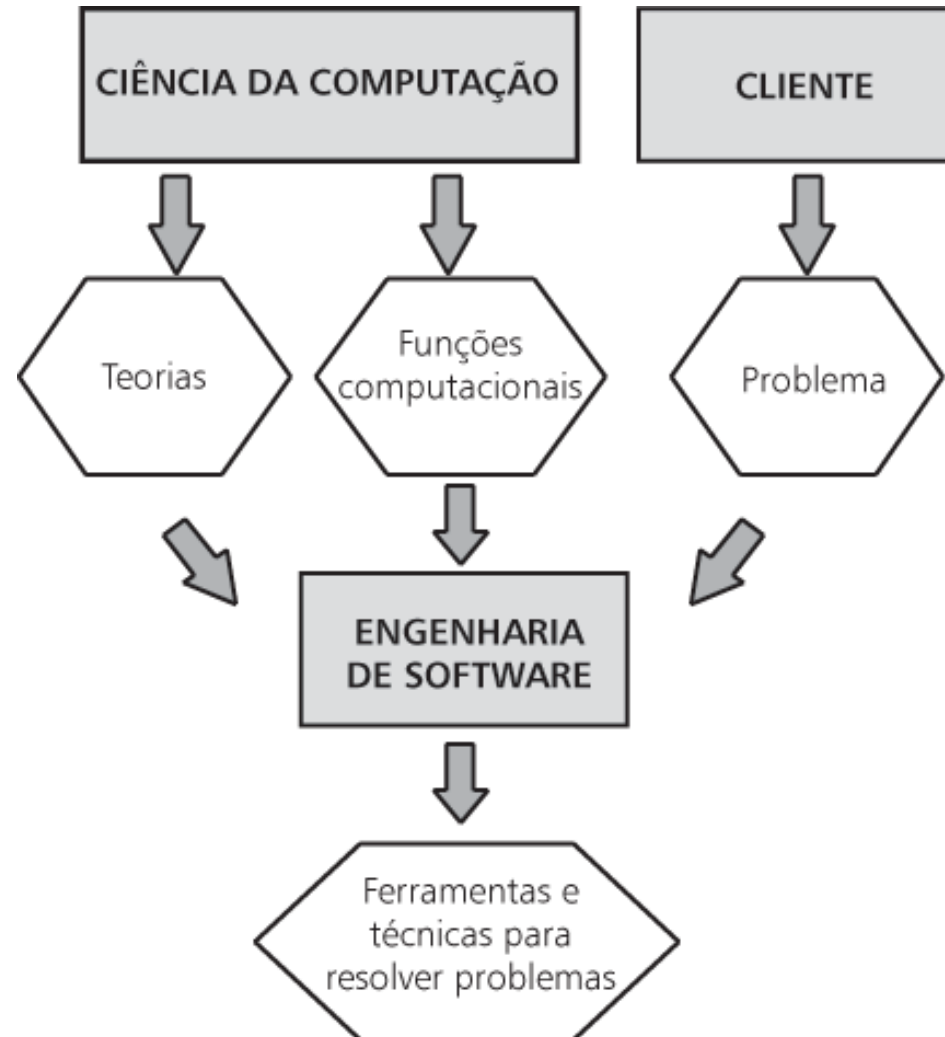
Custos de software

- Os custos de software dominam os custos de sistemas computacionais. Em um PC, os custos de software são freqüentemente maiores que o custo do hardware.
- Manter um software custa mais que desenvolvê-lo. Para sistemas com uma longa vida, os custos de manutenção podem ser muito maiores que os custos de desenvolvimento.
- A engenharia de software dedica-se ao desenvolvimento de software com custos adequados.

Custos de software

- Aproximadamente 60% dos custos são custos de desenvolvimento e 40% são custos de testes. Para software sob encomenda, os custos de evolução normalmente excedem de desenvolvimento.
- Os custos variam dependendo do tipo de sistema que está sendo desenvolvido e dos requisitos de atributos de sistema, tais como desempenho e confiabilidade.
- A distribuição de custos depende do modelo de desenvolvimento que é usado.

O papel do Engenheiro de Software



Qualidade — Terminologias

- Erro: engano humano
- Defeito: resultado do erro evidenciado em algum desenvolvimento ou manutenção do produto. Um simples erro gera vários defeitos.
- Falha: divergência entre o comportamento requerido para o sistema e o comportamento real. Indica que o sistema não está atuando como requerido, embora ele possa estar atuando conforme o especificado.



O que é um bom software?

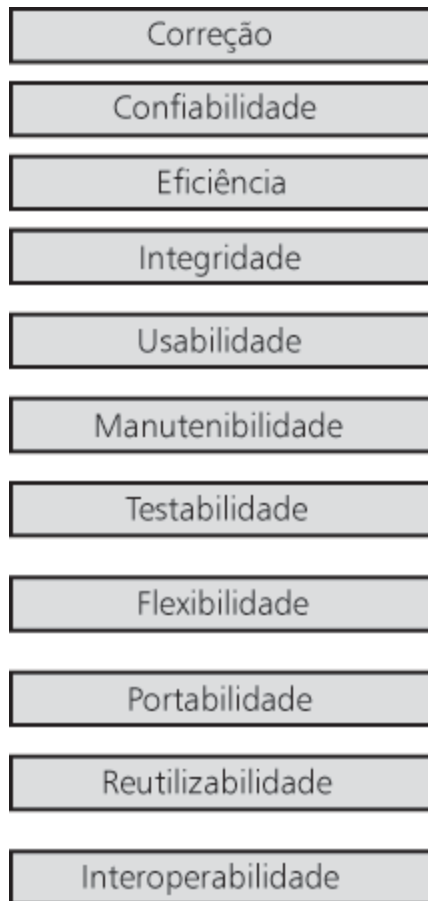
- Visão transcendental: algo que podemos reconhecer, mas não definir
- Visão do usuário: conveniência para propósito pretendido
- Visão do fabricante: conformidade com especificação
- Visão do produto: relação com as características inerentes ao produto
- Visão do mercado: dependência de quanto os consumidores estão dispostos a pagar

O que é um bom software?

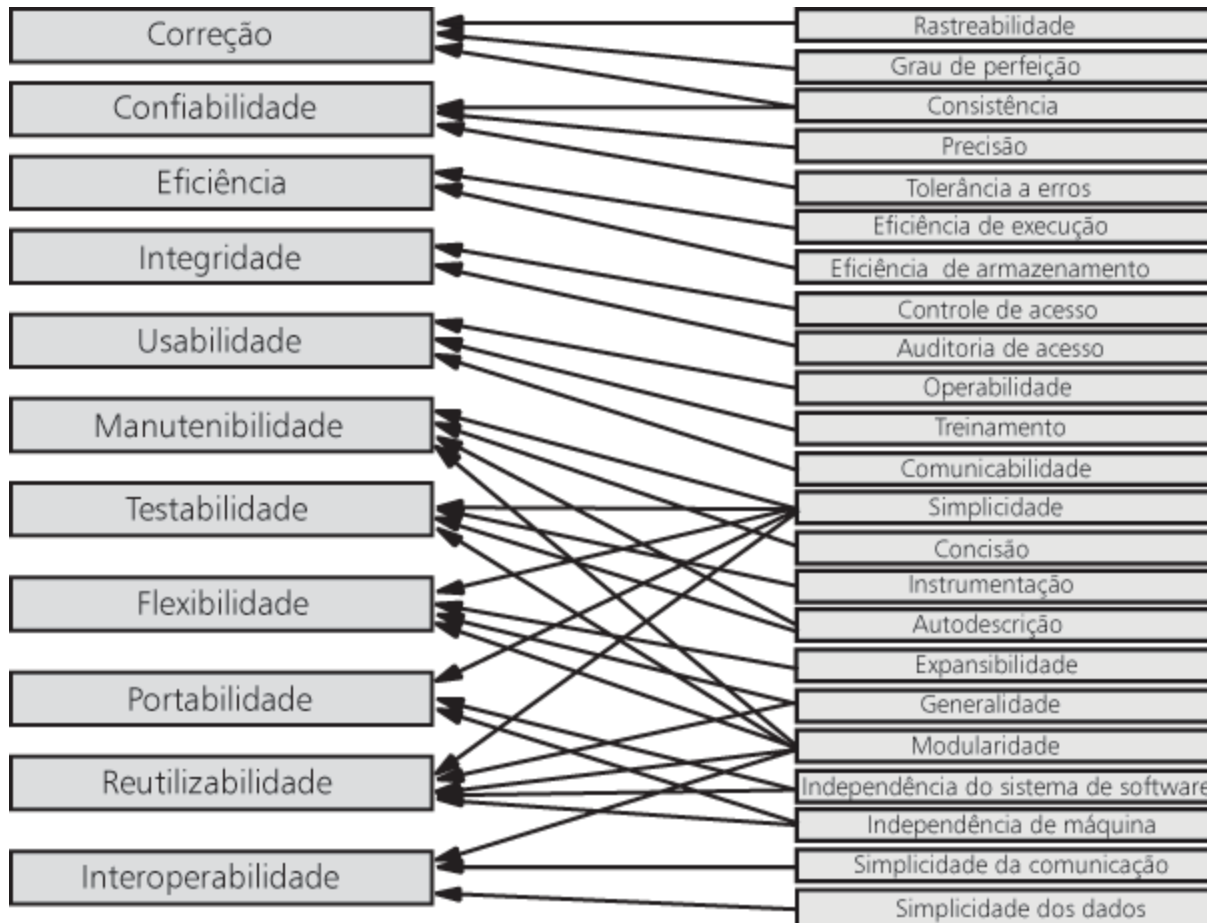
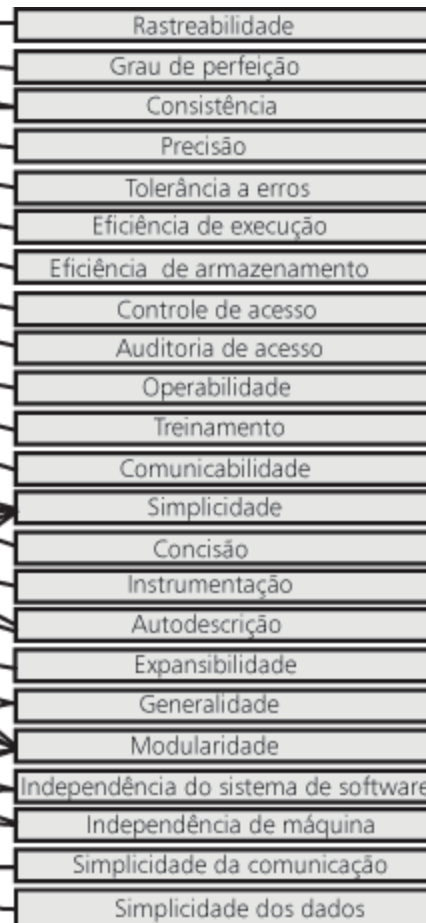
- O software deve fornecer a funcionalidade e o desempenho requeridos para o usuário e deve ser manutenível, confiável e aceitável.
- **Facilidade de manutenção**
 - O software deve evoluir para atender às necessidades de mudança;
- **Confiança**
 - O software deve ser confiável;
- **Eficiência**
 - O software não deve desperdiçar os recursos do sistema;
- **Usabilidade**
 - O software deve ser aceito pelos usuários para o qual foi projetado. Isso significa que ele deve ser compreensível, usável e compatível com outros sistemas.

Modelo de Qualidade de McCall

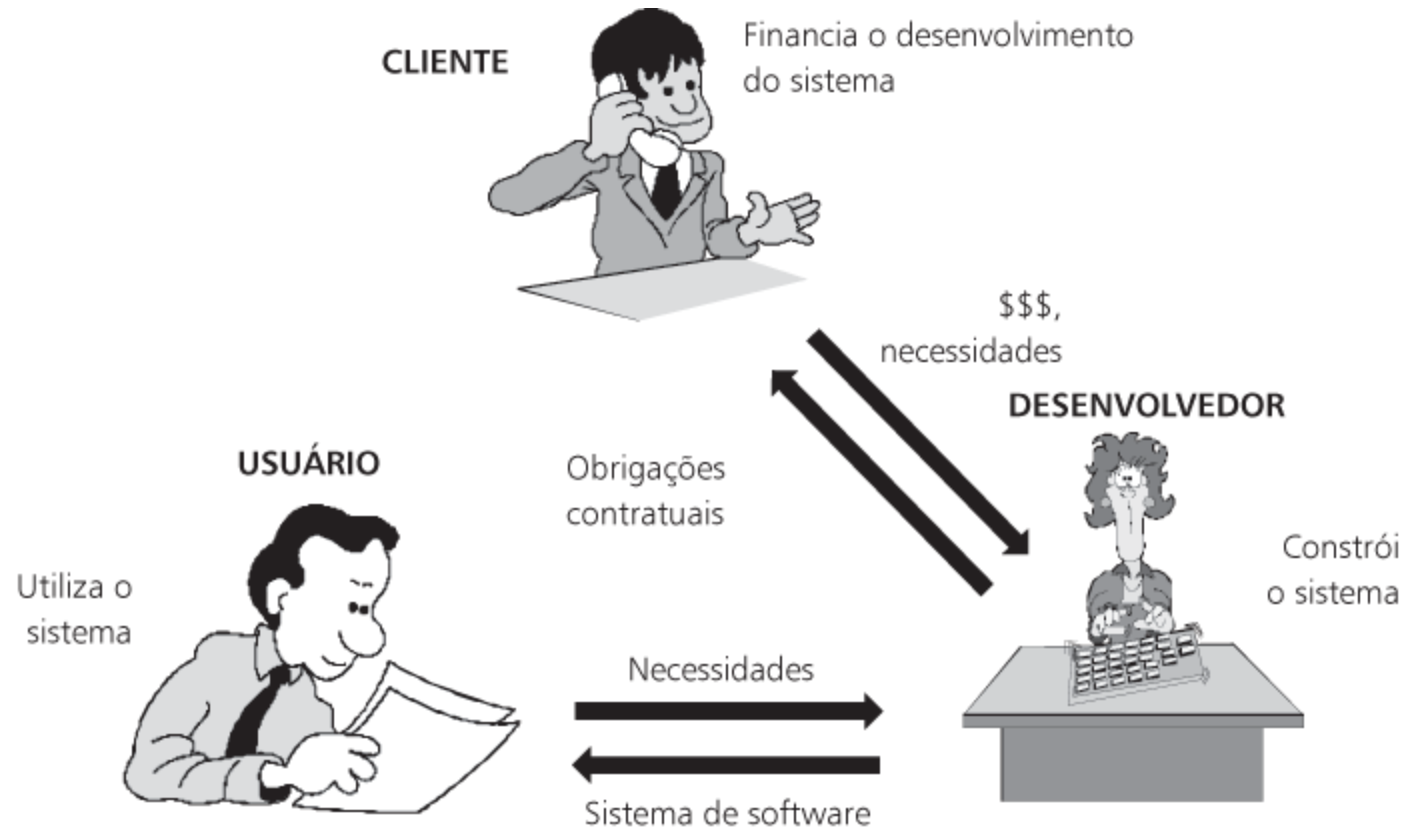
Visão Externa



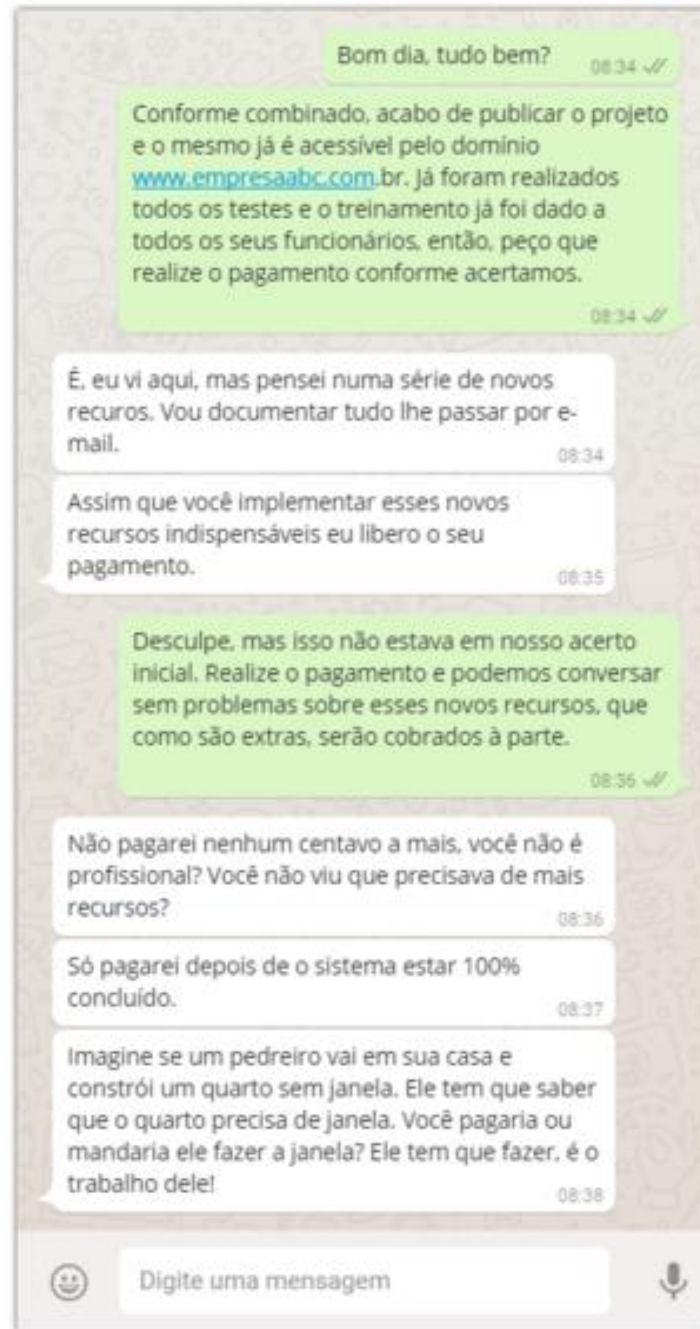
Visão interna



Quem faz a Eng. de Software?



Para refletir...



**E assim, passam-se
6 meses e você não
vê a cor do
dinheiro...**



Ética e engenharia de Software

Código de ética e prática profissional da engenharia de software

Força-tarefa conjunta da ACM/IEEE-CS para ética e práticas profissionais da engenharia de software

Prefácio

A versão reduzida do código resume as aspirações em um alto nível de abstração; as cláusulas incluídas na versão completa fornecem exemplos e detalhes de como essas aspirações mudam o nosso modo de agir como profissionais de engenharia de software. Sem as aspirações, os detalhes podem se tornar legalistas e tediosos; sem os detalhes, as aspirações podem ficar pomposas, porém vazias; juntos, as aspirações e os detalhes formam um código coeso.

Os engenheiros de software devem se comprometer a fazer da análise, especificação, projeto, desenvolvimento, teste e manutenção do software uma profissão útil e respeitada. De acordo com o seu compromisso com a saúde, segurança e bem-estar do público, os engenheiros de software devem obedecer aos oito princípios a seguir:

1. Público — Os engenheiros de software devem agir coerentemente com o interesse público.
2. Cliente e empregador — Os engenheiros de software devem agir de uma maneira que atenda aos interesses de seu cliente e empregador, coerente com o interesse público.
3. Produto — Os engenheiros de software devem assegurar que seus produtos e modificações relacionadas cumpram o máximo possível os mais altos padrões profissionais.
4. Opinião — Os engenheiros de software devem manter a integridade e independência em sua opinião profissional.
5. Gestão — Os gestores e líderes em engenharia de software devem aceitar e promover uma abordagem ética do gerenciamento do desenvolvimento e manutenção do software.
6. Profissão — Os engenheiros de software devem promover a integridade e a reputação da profissão em conformidade com o interesse público.
7. Colegas — Os engenheiros de software devem ser justos e apoiar os colegas.
8. Caráter — Os engenheiros de software devem aderir a uma aprendizagem contínua durante toda a vida no que diz respeito à prática de sua profissão e promover uma abordagem ética para a prática da profissão.

Versão resumida. Disponível em: <<https://ethics.acm.org/code-of-ethics>>. Acesso em: 27 mar. 2018.

Dúvidas mais frequentes

Pergunta	Resposta
O que é software?	Programas de computador e documentação associada. Os produtos de software podem ser desenvolvidos para um determinado cliente ou para um mercado genérico.
Quais são os atributos do bom software?	O bom software deve proporcionar a funcionalidade e o desempenho necessários e deve ser manutenível, usável e com dependabilidade (<i>dependability</i>).
O que é engenharia de software?	A engenharia de software é uma disciplina de engenharia que se preocupa com os aspectos da produção de software, desde sua concepção inicial até sua operação e manutenção.
Quais são as atividades fundamentais da engenharia de software?	Especificação, desenvolvimento, validação e evolução do software.
Qual é a diferença entre engenharia de software e ciência da computação?	A ciência da computação se concentra na teoria e nos fundamentos. A engenharia de software se preocupa com as questões práticas de desenvolver e entregar software útil.
Qual é a diferença entre engenharia de software e engenharia de sistemas?	A engenharia de sistemas se preocupa com todos os aspectos do desenvolvimento de sistemas computacionais, incluindo hardware, software e engenharia de processos. A engenharia de software faz parte desse processo mais geral.
Quais são os principais desafios enfrentados pela engenharia de software?	Lidar com a crescente diversidade, com as demandas por menores prazos de entrega e desenvolver software confiável.
Quais são os custos da engenharia de software?	Aproximadamente 60% dos custos de software são relativos ao desenvolvimento e 40%, aos testes. Quanto ao software personalizado, os custos de evolução frequentemente ultrapassam os de desenvolvimento.
Quais são os melhores métodos e técnicas de engenharia de software?	Ainda que todos os projetos de software devam ser gerenciados e desenvolvidos profissionalmente, técnicas diferentes são adequadas para tipos diferentes de sistemas. Por exemplo, jogos devem ser sempre desenvolvidos usando uma série de protótipos, enquanto sistemas de controle críticos em segurança requerem o desenvolvimento de uma especificação completa e analisável. Não há métodos ou técnicas que sejam bons para todos os casos.
Quais diferenças a internet trouxe para a engenharia de software?	A internet não só levou ao desenvolvimento de sistemas massivos, largamente distribuídos, baseados em serviços, como também deu base para a criação de uma indústria de aplicativos (ou " <i>apps</i> ") para dispositivos móveis que mudou a economia de software.