

Introdução à Engenharia de Software

- Apresentação da disciplina
- Conceitos básicos

Objetivos

- Apresentar a área de Engenharia de Software
- Discutir a importância do software na sociedade contemporânea
- Mostrar a necessidade de se construir software de qualidade com produtividade

Referências Bibliográficas

- Básica:

PRESSMAN, Roger S; MAXIM, Bruce R. **Engenharia de software**: uma abordagem profissional. 9. Porto Alegre AMGH 2021 1 recurso online.

- Complementar:

SOMMERVILLE, Ian. **Engenharia de software**. 10 ed. São Paulo: Pearson, 2019.

O custo do hardware

- Durante as três primeiras décadas da era da computação, o principal desafio era desenvolver um hardware que reduzisse o custo de processamento e de armazenagem de dados

Isto ocorreu?

Com qual velocidade?

Um breve histórico da Computação (1/5)

- Primeira era (195? e 196?)
 - Orientação para o processamento batch
 - Distribuição limitada de softwares
 - Softwares extremamente padronizados
 - Produção artesanal sob demanda
 - Os profissionais de informática mantinham em suas cabeças o projeto das aplicações

Um breve histórico da Computação (2/5)

- Segunda era (196? e 197?)
 - Sistemas multiusuários
 - Surgimento de *software houses* e *bureaus* de processamento
 - Surgimento dos pacotes de softwares, que passam a ser tratados como produto
 - Processamentos real-time
 - Início da utilização da tecnologia de Banco de dados
 - Surgimentos dos cursos de Ciência da Computação

História da Engenharia de Software

História da engenharia de software

O conceito de engenharia de software foi proposto pela primeira vez em 1968, em uma conferência realizada para discutir o que então se chamava crise do software (NAUR; RANDALL, 1969). Ficou claro que as abordagens individuais ao desenvolvimento de programas não escalavam para sistemas de software grandes e complexos. Os sistemas não eram confiáveis, custavam mais do que o previsto e eram entregues com atraso.

Durante os anos 1970 e 1980, foi desenvolvida uma série de técnicas e métodos de engenharia de software, como a programação estruturada, a ocultação da informação (*information hiding*) e o desenvolvimento orientado a objetos. Foram desenvolvidas ferramentas e notações que compõem a base da engenharia de software atual.



(SOMMERVILLE, 2019. p. 4)

Um breve histórico da Computação (3/5)

- Terceira era (197? e 198?)
 - Sistemas distribuídos, principalmente em função da evolução das tecnologias de redes de computadores
 - Inteligência embutida nos microprocessadores
 - Diminuição do custo do hardware
 - Os sistemas de computação passam a interferir no cotidiano das pessoas

Um breve histórico da Computação (4/5)

- Quarta era (198? e 200?)
 - Aumento do poder das aplicações desktop
 - Tecnologia de orientação à objetos
 - Sistemas especialistas
 - Redes neurais
 - Computação paralela
 - Arquitetura *multitier*

Um breve histórico da Computação (5/5)

- Quinta era (200? até hoje)
 - Consolidação dos sistemas Web
 - Sistemas distribuídos de grande escala
 - Aplicações para dispositivos móveis
 - *Cloud Computing*
 - *Big Data*
 - *Machine Learning, IA e Deep Learning*

Em síntese

- A velocidade de evolução do hardware foi e está sendo muito superior a velocidade de evolução do software
- Os softwares não têm conseguido acompanhar as mudanças impostas pelos ambientes organizacionais
- As organizações estão cada vez mais dependentes dos recursos e serviços advindos dos softwares
- É necessário construir softwares cada vez mais confiáveis e de qualidade
- O prazo para desenvolvimento de um software é cada vez mais curto

Algumas questões para reflexão

- Por que os custos de desenvolvimento são tão altos?
- Por que não conseguimos encontrar todos os erros antes de entregarmos o software aos clientes?
- Por que a conclusão de um software leva tanto tempo?
- Por que gastamos tanto tempo e esforço realizando a manutenção de programas existentes?
- Por que ainda temos dificuldades de medir o progresso do desenvolvimento e a manutenção de um software?

(PRESSMAN, 2021. p. 4)

Alguns problemas (1/4)

- Relacionados ao **Processo**
 - Cronogramas muito otimistas
 - Gerência de risco inexistente
 - Falha de contratação de recursos
 - Planejamento insuficiente
 - Abandono do planejamento por problemas de tempo

Alguns problemas (2/4)

- Relacionados ao **Processo**
 - Gasto de tempo durante a concepção
 - Corte míope de atividades que não sejam codificação (ex: análise ou arquitetura)
 - Design inadequado
 - SQA ou SCM inexistente

Alguns problemas (3/4)

- Relacionados ao **Produto**
 - Requisitos mal formulados, ou mal entendidos
 - +/- 25% dos requisitos mudam em projetos
 - Desenvolvedores com tecnologias “folheadas em ouro”
 - Negociação “Puxa-empurra”
 - Cronograma é esticado, e mais tarefas são adicionadas

Alguns problemas (3/4)

- Relacionados ao **Produto**
 - Requisitos mal formulados, ou mal entendidos
 - +/- 25% dos requisitos mudam em projetos
 - Desenvolvedores com tecnologias “folheadas em ouro”
 - Negociação “Puxa-empurra”
 - Cronograma é esticado, e mais tarefas são adicionadas

Alguns problemas (4/4)

- Relacionados ao **Tecnologia**
 - Superestimar ganhos no uso de uma nova tecnologia
 - Trocar de ferramentas no meio do projeto
 - Falha de controle automático do código fonte (sem controle de versão)

Software e suas Aplicações

Um conceito para Software

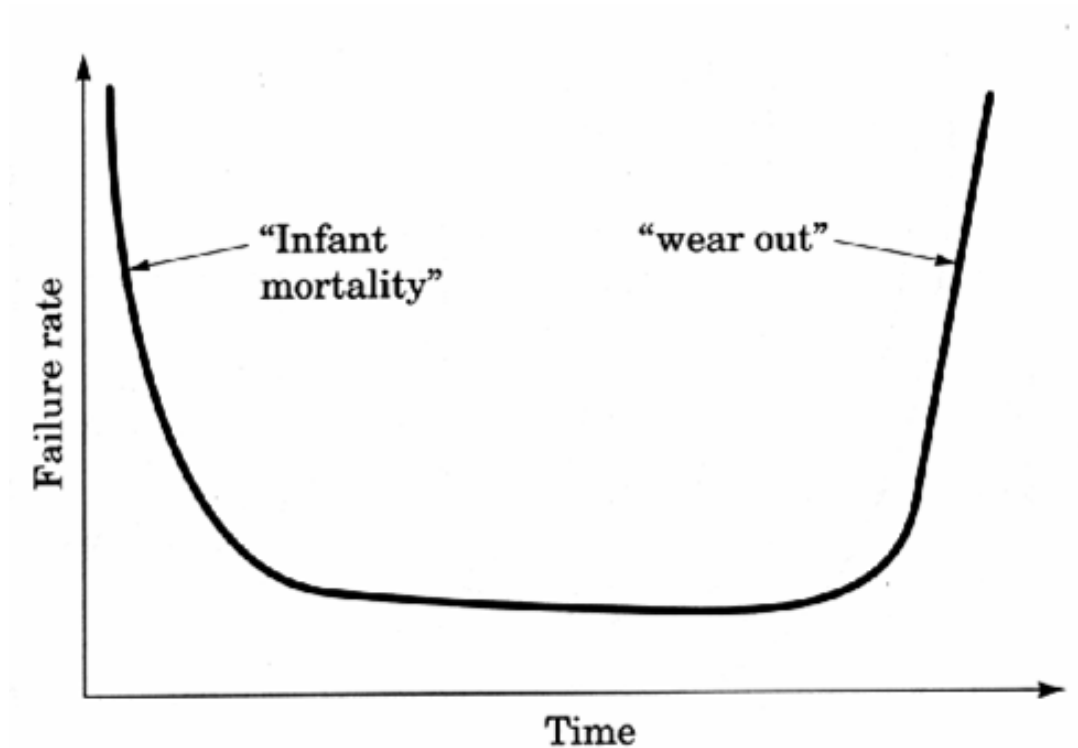
“Software consiste em: (1) instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados; (2) estruturas de dados que possibilitam aos programas manipular informações adequadamente; e (3) informação descritiva, tanto na forma impressa como na virtual, descrevendo a operação e o uso dos programas.” (PRESSMAN, 2021. p. 4)

Características do Software

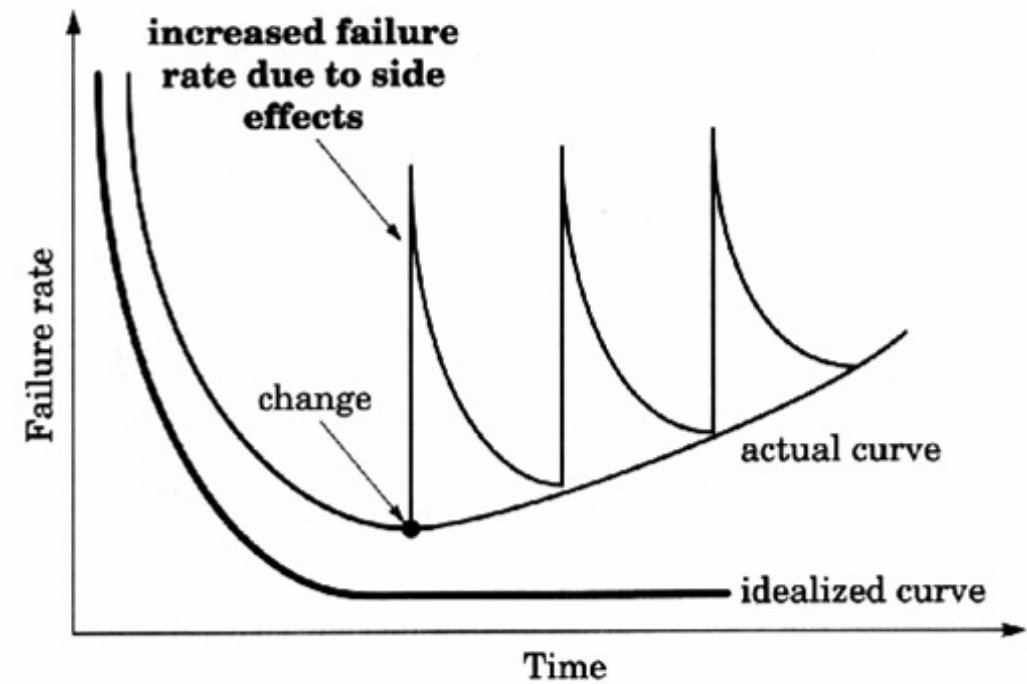
- O software é desenvolvido ou projetado por engenharia, e não manufaturado no sentido clássico
- O software não “se desgasta”
- Embora a indústria caminhe para a construção com base em componentes, a maioria dos softwares continua a ser construída de forma personalizada (sob encomenda)

Curva de falha

(Pressman, 2016, p. 6)



Hardware



Software

Natureza do Software

- Software é um produto
 - produz, gerencia, adquire, modifica, exhibe ou transmite informação

Natureza do Software

- Software é um veículo para entrega de um produto
 - controla outros software (Sistemas Operacionais)
 - viabiliza a comunicação de dados (Redes)
 - facilita a construção de outros softwares

Aplicações do Software (1/9)

- Software de sistemas
 - programas que apoiam o funcionamento de outros programas
 - forte interação com o hardware

Aplicações do Software (2/9)

- Software de tempo real
 - monitora, analisa e controle eventos do mundo real tempo real é diferente de interativo ou *time-sharing*

Aplicações do Software (3/9)

- Software de aplicação
 - amplamente difundido
 - estruturam os dados de forma a facilitar a gestão das organizações e a vida das pessoas

Aplicações do Software (4/9)

- Software científico e de engenharia
 - vão desde a astronomia até a vulcanologia
 - trabalham e processam números
 - CAD

Aplicações do Software (5/9)

- Software embutido (*embedded software*)
 - reside na memória só de leitura (*read only*)
 - controla produtos e sistemas no mercado industrial

Aplicações do Software (6/9)

- Aplicações Web / Aplicativos Móveis
 - Hoje, fazem parte do dia-a-dia de muitas pessoas ao redor do mundo!

Aplicações do Software (7/9)

- Software de Inteligência Artificial
 - faz uso de algoritmos não numéricos para resolver problemas complexos que não sejam favoráveis à computação
 - sistemas especialistas baseados no conhecimento
 - redes neurais artificiais

Aplicações do Software (8/9)

- Software Livre
 - possuem código fonte auto descritivo que facilita a sua modificação e evolução

Aplicações do Software (9/9)

- Computação Ubíqua
 - softwares que permitem pequenos dispositivos e computadores pessoais se comunicarem em qualquer ambiente criando um contexto *anywhere* e *always-on*

Sistema de Software Legado

Sistemas de software legado... foram desenvolvidos décadas atrás e têm sido continuamente modificados para se adequar às mudanças dos requisitos de negócio e a plataformas computacionais. A proliferação de tais sistemas está causando dores de cabeça para grandes organizações que os consideram dispendiosos de manter e arriscados de evoluir.

(PRESSMAN, 2021. p. 8)

Software Legado

- Os softwares precisam estar adaptados aos novos ambientes e às novas tecnologias
- Os softwares crescem para atender os novos requisitos
- Os softwares precisam estender a sua interoperabilidade
- Os softwares precisam ser rearquitetados para os novos ambientes de rede

Engenharia de Software

Alguns fatos reais

- Entender o problema antes de desenvolver uma solução inovadora de software
 - Contudo, o software não pode complicar a vida do usuário
- Projetar é uma atividade fundamental
- Um software deve ter uma qualidade elevada
- O software deve ser fácil de ser mantido

Uma definição

[Engenharia de software é] o estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter software de maneira econômica, que seja confiável e funcione de forma eficiente em máquinas reais

A definição do IEEE

Engenharia de software: (1) A aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software; isto é, a aplicação de engenharia ao software. (2) O estudo de abordagens como definido em (1).

Camadas da Engenharia de Software



Figura 1.3

Camadas da engenharia de *software*.

(PRESSMAN, 2021. p. 9)

Questões para a Engenharia de Software

- Qual problema tem que ser resolvido?
- Quais características do software são utilizadas para resolver o problema?
- Como o software será construído?
- Como os erros serão identificados?
- Como o software será mantido?

Código de Ética do Engenheiro de Software

FIGURA 1.3 Código de ética da ACM/IEEE (© 1999 by the ACM Inc. and the IEEE, Inc.).

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.

(SOMMERVILLE, 2019. p. 15)

Código de Ética do Engenheiro de Software

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.

(SOMMERVILLE, 2019. p. 15)