

Engenharia de Software: Histórico, Terminologia e Casos

Prof. André Takeshi Endo

Engenharia de Software

- Contextualizou o cenário de engenharia de software no qual vocês estavam!
- Motivou a importância da disciplina

Histórico (1950-1965)

- O hardware sofreu contínuas mudanças
- O software era uma arte "secundária" para a qual havia poucos métodos sistemáticos
- O hardware era de propósito geral
- O software era específico
- Não havia documentação

Histórico (1965-1975)

- Multiprogramação e sistemas multiusuários
- Técnicas interativas
- Sistemas de tempo real
- 1ª geração de SGBD's
- Produto de software - software houses
- Bibliotecas de Software

Histórico (1965-1975)

- Cresce o número de sistemas de software
- Manutenção quase impossível
- Crise de Software

Histórico

- 1968, *NATO Software Engineering Conference*
 - Termo “crise de software”
 - Termo “Engenharia de software”

Histórico

- Crise de software
 - Hardware barato + software caro
- Manifestações da crise
 - Projetos atrasados
 - Projetos acima do orçamento
 - Software ineficiente
 - Software de baixa qualidade
 - Software não atendia os requisitos
 - Problemas de gerenciamento e manutenção
 - Software não era entregue

Histórico

- Crise de software
- Segundo Dijkstra,
 - *“the major cause is... that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a **mild problem**, and now we have gigantic computers, programming had become an equally **gigantic problem**.”*

Histórico (1975 - hoje)

- Sistemas distribuídos
- Redes locais e globais (Internet)
- Uso generalizado de microprocessadores - produtos inteligentes
- Hardware de baixo custo
- Alto consumo de software
- Dispositivos móveis
- Computação pervasiva ou ubíqua

Histórico

- E a crise?
 - Alguns acreditam que ela ainda não passou
 - Outros dizem que ela é coisa do passado
- O que vocês acham?
 - Exemplos mais adiante

Histórico - Pessoas

- E.W. Dijkstra (algoritmos, grafos)
- David Parnas (modularidade)
- Donald Knuth (*art of computing programming*)
- Fred Brooks
- Linus Torvalds, Richard Stallman (software livre)
- Barry Boehm (qualidade de software)
- Gang of Four (padrões de projeto)
- Victor Basili (GQM, qualidade, experimentos)
- Kent Beck, Martin Fowler (métodos ágeis)

"10 Best Influences on Software Engineering"

<http://www.stevemcconnell.com/ieeesoftware/eic09.htm>

Terminologia

- O que é Software? [Pfleeger07]
- “O conceito mais geral de software compreende todo o conjunto de programas, procedimentos, dados e documentação associados a um sistema de computador, e não somente ao programa em si.”

Criar um mapa de conceitos!

Terminologia

- O que é Software? [Pressman11]
- “Software de computador é o produto que profissionais de software desenvolvem e ao qual dão suporte no longo prazo. Abrange programas executáveis em um computador de qualquer porte ou arquitetura, conteúdos (apresentados a medida que os programas são executados), informações descritivas tanto na forma impressa (hard copy), como na virtual, abrangendo praticamente qualquer mídia eletrônica.”

Terminologia

- O que é Software? [Sommerville03]
- “Software não é apenas o programa mas também toda a documentação associada e os dados de configuração necessários para fazer com que esses programas operem corretamente. Um sistema de software, usualmente, consiste em uma série de programas separados, arquivos de configuração que são utilizados para configurar esses programas, documentação do sistemas, que descreve a estrutura desse sistema, e documentação do usuário, que explica como utilizar o sistema e, no caso de produtos de software, sites Web para os usuários fazerem download das informações recentes sobre o produto.”

Terminologia

- O que é Software? [IEEE90]

software. Computer programs, procedures, and possibly associated documentation and data pertaining to the operation of a computer system. *See also:* **application software; support software; system software.** *Contrast with:* **hardware.**

Terminologia

- O que é Engenharia de Software?
 - “estabelecimento e 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*.”

software engineering. (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.
(2) The study of approaches as in (1).

Terminologia

- O que é Engenharia de Software? [Pressman]



Ferramentas

Métodos

Processo

Foco na qualidade

Terminologia

- Software e Engenharia de Software
- Software legado
- Papéis
- Fases
- Artefatos
- Métodos / técnicas / ferramentas

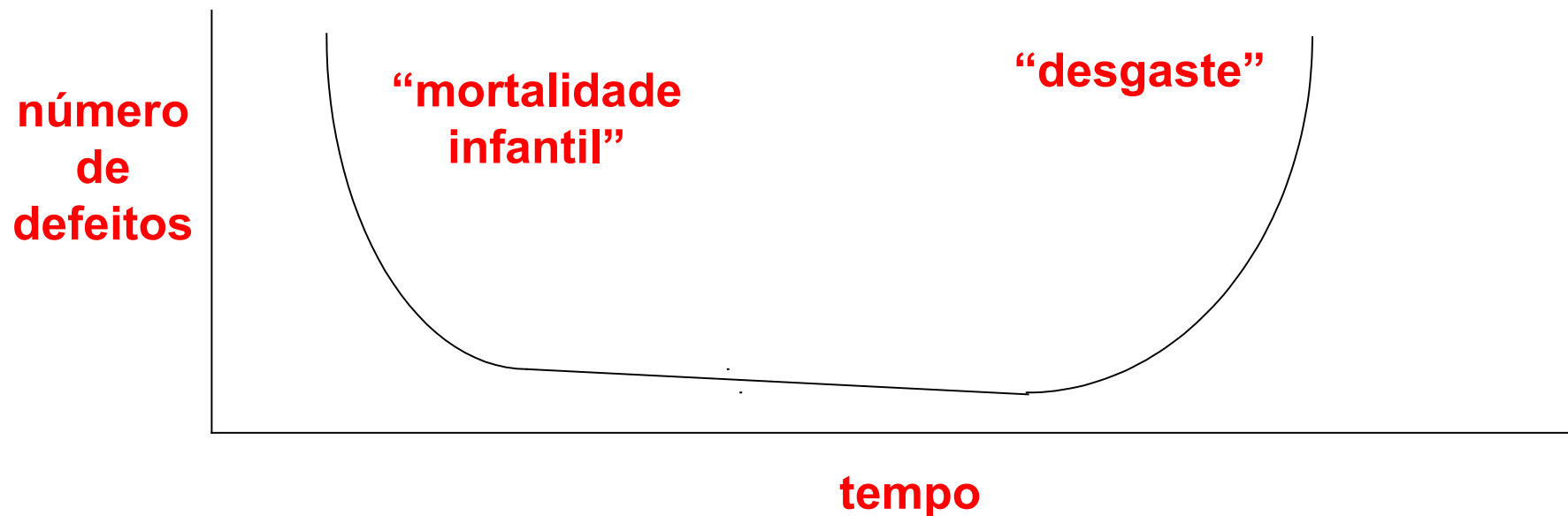
Natureza do Software

- Software é desenvolvido ou passa por um processo de engenharia, não é fabricado no sentido clássico
- “*não se desgasta mas se deteriora*”

Natureza do Software

- Como é em hardware?

Q: vocês conseguem interpretar esse gráfico?



CURVA DE DEFEITOS DO HARDWARE

Natureza do Software

- “*Software não se **desgasta** mas se **deteriora**”*”

Significado de Desgastar

v.t. Gastar, destruir aos poucos.
Consumir pela fricção ou pelo atrito.
Digerir.

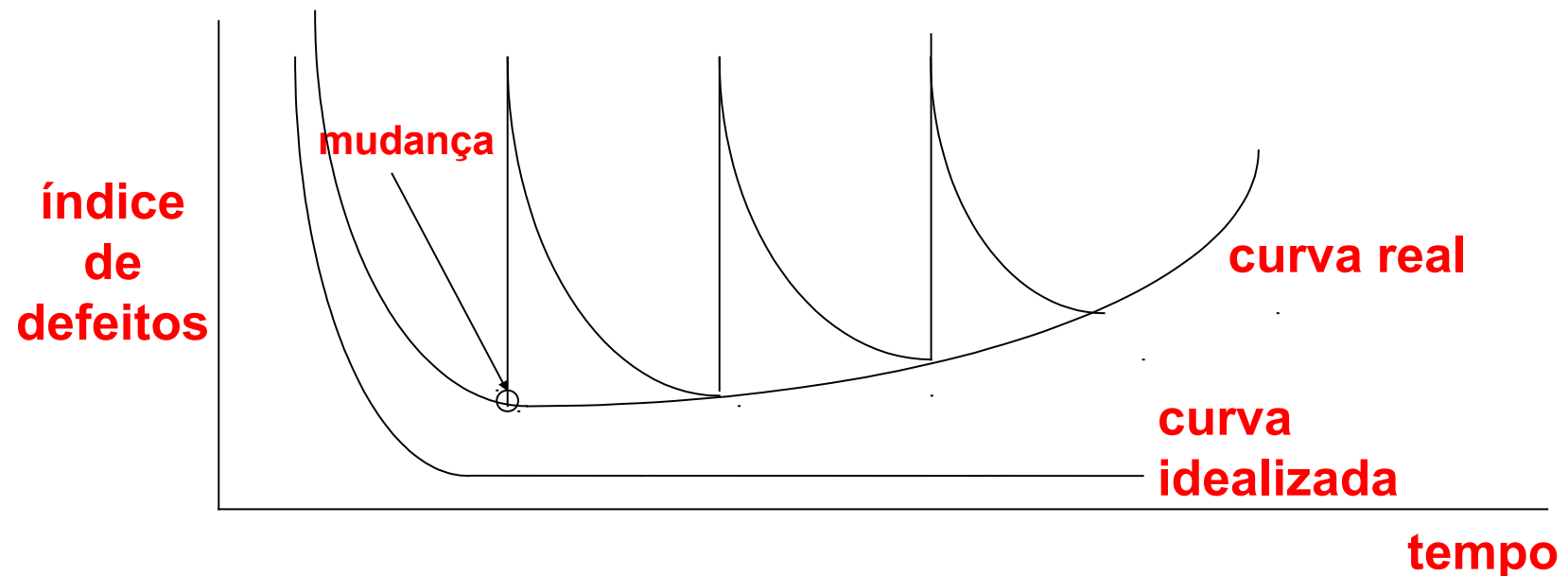
Significado de Deteriorar

v.t. Pôr em mau estado; danificar; estragar; arruinar.
V.pr. Danificar-se, estragar-se, apodrecer.

Natureza do Software

- Como é em software?

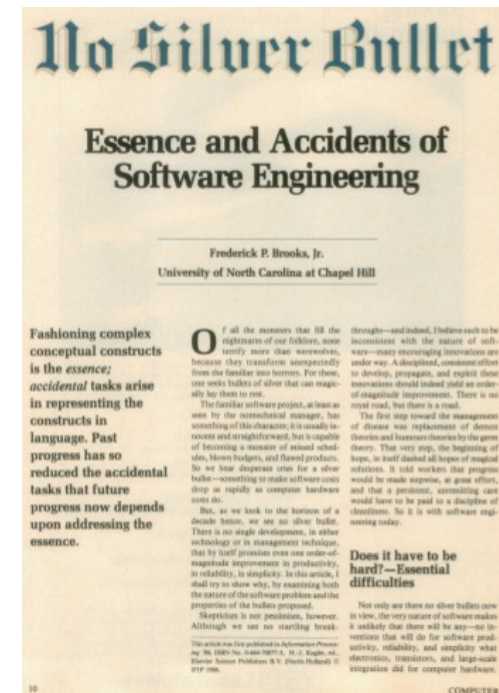
Q: vocês conseguem interpretar esse gráfico?



CURVA DE DEFEITOS DO SOFTWARE

Propriedades inerentes do Software

- “*No Silver Bullet: Essence and Accidents of Software Engineering*” [Brooks87]
- Complexidade
- Conformidade
- Modificabilidade
- Invisibilidade



Propriedades inerentes do Software

- Complexidade
 - Quanto maior, mais complexo
 - Partes (componentes) não se repetem
 - Software + computadores é uma das coisas mais complexas que o homem já contruiu (se não for a mais!)
 - Complexidade em software não aumenta linearmente

Propriedades inerentes do Software

- Conformidade
 - Física também é complexa (teoria unificada)
 - Complexidade é arbitrária
 - Software deve ser adequar (conformidade)
 - Interfaces, instituições, clientes, regras, governos

Propriedades inerentes do Software

- Modificabilidade
 - Propenso (pressionado) a mudanças
 - Outros produtos não mudam
 - Embuti funções
 - Software é infinitamente maleável
 - Software está inserido em contextos mutantes (leis, usuários, comércio, etc)

Propriedades inerentes do Software

- Invisibilidade
 - Software é invisível e não-visualizável
 - Não possui representação geométrica (prédios, carros, etc)
 - Não há ainda um padrão para representação (UML)

Casos

- Situações reais
 - Sucesso e falha!

Casos (de falha)

- Década de 80 [Pflegeer]
 - Receita federal dos Estados Unidos
 - Sperry Corporation
 - Processamento de formulários do IR
 - US\$ 103 milhões (inicial) + US\$90 milhões
 - Atrasos
 - US\$ 40,2 milhões de juros
 - US\$ 23 milhões de horas extras
 - US\$ 4 bilhões

Casos (de falha)

- Máquina Therac-25
 - Terapia de radiação e raio X
 - Morte de pacientes
 - Uso não-convencional das teclas de seta
 - Alta dose de radiação

Casos (de falha)

- Nasdaq
 - IPO do Facebook
 - US\$10 milhões
 - Mensagens não foram enviadas aos investidores
 - US\$42,05 → US\$38,00

Casos (de falha)

- Knight Capital
 - Software que negocia ações automaticamente
 - 45 minutos → US\$ 440-460 milhões
 - Impacto em várias ações
 - Comprada pela Getco LLC

Casos (de falha)

- Detector de alvos (jato Harrier), 1996
 - Dois policiais
 - Uma radar de velocidade
 - “300 milhas”
 - Inimigo



Casos (de falha)

- E o nosso estudo de caso?

Casos (de sucesso)

- Facebook
 - Rede social (76 milhões)
 - Mark Zuckerberg
 - É complexo de implementar?
- Conhecem outros exemplos?



Bibliografia

- [Pfleeger07] S. L. Pfleeger, “Engenharia de Software: Teoria e Prática”, 2007.
- [Pressman11] R. S. Pressman, “Engenharia de Software: uma abordagem profissional”, 2011.
- [Sommerville03] I. Sommerville, “Engenharia de Software”, 2003.
- [Brooks87] “No Silver Bullet: Essence and Accidents of Software Engineering”, 1987.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1663532
- [IEEE90] “IEEE Standard Glossary of Software Engineering Terminology”, 1990.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159342

Bibliografia

- [UU] Materiais didáticos elaborados pelos grupos de engenharia de software do ICMC-USP e DC-UFSCAR.