

Software e Engenharia de Software

Karin Becker

Instituto de Informática - UFRGS

FAQ: Engenharia de SW vs. Ciência da Computação

- A ciência da computação dedica-se à teoria e aos fundamentos que sustentam sistemas computacionais e de software
- Engenharia de software dedica-se aos aspectos práticos de desenvolvimento e de entrega de software para o uso
- A engenharia de software requer, entre outros, conhecimentos de ciência da computação

Engenharia de Software (ES)

- F.L. Bauer, 1968
 - *“The whole trouble comes from the fact that there is so much tinkering with software. It is not made in a clean fabrication process, which it should be. What we need, is **software engineering**.”*
- **Software Engineering** (IEEE Std. 610.12 (1990))
 - (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).

O que é Software?

- SW é um produto “diferente”
 - **Virtual:**
 - falta de leis e propriedades físicas para SW
 - não degrada com o tempo (mas se torna obsoleto)
 - **Maleável:** pode ser modificado após pronto
- Dificuldades
 - NÃO há teoria subjacente (motor elétrico: equações de tensão, potência, corrente)
 - Todos os artefatos de engenharia existem em um ambiente e para propósitos externos
 - Projeto de SW frequentemente envolve reprojetar (ou mesmo inovar) sobre processos de trabalho, organizações ou aspectos do dia a dia

Programa e Software

Programa

Uso Pessoal
Doc pequena (qdo existe)
Usuário é o autor
Erro é “irrelevante”
Sem manutenção



- Programa é artefato
- Desenvolvimento é “arte”
- Atividade Pessoal
(*programming-in-the-small*)

Software

Uso Comercial
Doc rica
Usuários diferenciados
Erro é grave
Muita manutenção



- Software é **produto**
- Desenvolvimento necessita de “**engenharia**”
- Construção em equipe de SW com múltiplas versões
(*programming-in-the-large*)

FAQ: O que é software?

- Programas de computador e documentação associada, tais como requisitos, modelos de projetos e manuais de usuário.
- 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

Engenharia de Software é Programação?

- Engenharia de Software \neq Programação
 - Sistematização das atividades de analisar, especificar, projetar, programar (implementar), verificar, validar, manter e gerenciar um projeto de software.
 - Software engloba não apenas programas, mas diferentes artefatos associados:
 - especificações, projetos e planos de teste
 - documentação técnica
 - documentação voltada aos usuários
 - configurações
 - Software envolve TI, pessoas e processos

ES: Precisa?

- Standish Chaos Report
 - Análise regular sobre milhares de projetos de desenvolvimento de software nos EUA
 - Desde 1994, a cada 2 anos
 - Dividem projetos pesquisados em três categorias
 - Sucesso: prazo, orçamento, funcionalidades
 - Desafiado: acima do orçamento/cronograma, menos funcionalidades
 - Falho: cancelado
 - Análise dos fatores de sucesso/insucesso

ES: Precisa?

Table I

Standish project benchmarks over the years

Year	Successful (%)	Challenged (%)	Failed (%)
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24



Para outros estudos sobre taxas de (in)sucesso :
www.galorath.com/wp/software-project-failure-costs-billions-better-estimation-planning-can-help.php

Programação

- Ad-hoc, sem planejamento
- Sistemas pequenos
 - Pouco tempo
 - Baixo custo
- Poucas pessoas envolvidas
 - tipicamente 1 líder decide tudo
- Pouco impacto em seu entorno
 - Erro é 'irrelevante'
 - Tipicamente, usado por seu autor
- Não se preocupa com padronização ... tampouco com documentação

Engenharia de Software

- Planejamento meticuloso
- Sistemas médios-grandes
 - Alto custo
 - Muito tempo
- Programação é uma de suas etapas
- Desenvolvido por um time
 - Pode ser composto por profissionais com vários perfis
 - Decisão é colegiada
- Tem que levar em consideração a padronização existente ... e tem que ser muito documentado.
- Muito impacto em seu entorno
 - Erro é grave
 - Qualidade é crítica

ES: Por quê?

- Alguns sintomas
 - Compreensão incompleta ou imprecisa das necessidades do usuário
 - Inabilidade de lidar com requisitos que evoluem
 - Módulos incompatíveis
 - Dificuldades de estender ou manter software
 - Descoberta de defeitos graves no projeto em etapas avançadas de desenvolvimento ou mesmo em época de implantação ou uso
 - Desempenho inaceitável do software
 - Falta de coordenação na equipe

Standish Chaos Report (1995)

Project Success Factors	% of Responses
1. User Involvement	15.9%
2. Executive Management Support	13.9%
3. Clear Statement of Requirements	13.0%
4. Proper Planning	9.6%
5. Realistic Expectations	8.2%
6. Smaller Project Milestones	7.7%
7. Competent Staff	7.2%
8. Ownership	5.3%
9. Clear Vision & Objectives	2.9%
10. Hard-Working, Focused Staff	2.4%
Other	13.9%

Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%
4. Lack of Executive Support	7.5%
5. Technology Incompetence	7.0%
6. Lack of Resources	6.4%
7. Unrealistic Expectations	5.9%
8. Unclear Objectives	5.3%
9. Unrealistic Time Frames	4.3%
10. New Technology	3.7%
Other	23.0%

Standish Chaos Report (1995)

Project Impaired Factors	% of Responses
1. Incomplete Requirements	13.1%
2. Lack of User Involvement	12.4%
3. Lack of Resources	10.6%
4. Unrealistic Expectations	9.9%
5. Lack of Executive Support	9.3%
6. Changing Requirements & Specifications	8.7%
7. Lack of Planning	8.1%
8. Didn't Need It Any Longer	7.5%
9. Lack of IT Management	6.2%
10. Technology Illiteracy	4.3%
Other	9.9%

ES: Por quê?

- Causas frequentes
 - Gerência de requisitos sem processo definido
 - Comunicação entre partes envolvidas
 - Insuficiente
 - ambígua e imprecisa
 - Complexidade crescente
 - Inconsistências não detectadas em nível de análise, projeto e implementações
 - Testes insuficientes
 - Dificuldade em lidar e gerenciar riscos
 - Falta de controle sobre propagação de mudanças
 - Automação insuficiente
 - Ubiquidade
 - Diversidade de plataformas

Diversidade de Aplicações

- Stand alone
- Iterativas baseadas em transações
- Processamento de lotes
- Embarcados
- Entretenimento
- Modelagem e simulação
- Coleta de dados
- Integração de sistemas
- ...
- Sem fronteiras definidas
- Novos modelos (cloud, SaaS, PaaS, IaaS, etc)

Diversidade de aplicações

- Não há técnicas, métodos, modelos e processos se se adequem a todos tipos de aplicações
 - “Não há bala de prata !!”
- Exemplo: Embarcados vs. Interativos on-line
 - Requisitos
 - Testes
 - Restrições e eficiência
 - Interação, usabilidade
 - Manutenção
 - Abordagem de desenvolvimento

Desenvolver, com qualidade ...

- Sistemas complexos e grandes foram, e continuam sendo desenvolvidos.



From Sommerville



Desenvolver, com qualidade ...

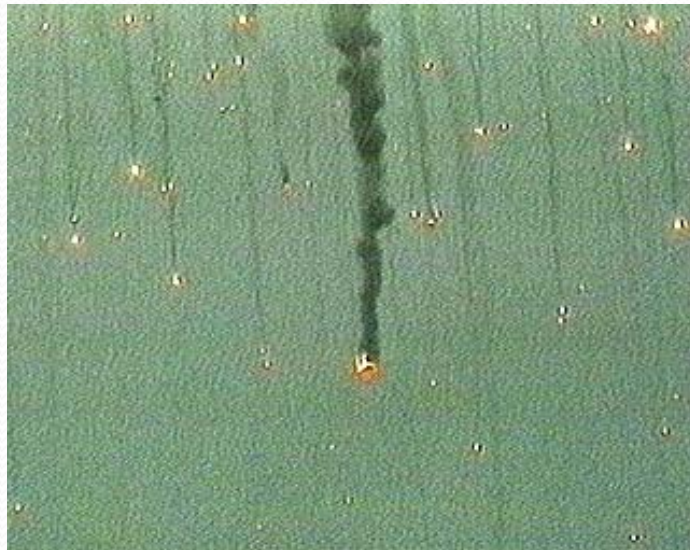
- Projeto Ariane 5



- Projeto da Agência Espacial Européia
- Garantir supremacia européia no espaço
- Custou:
 - 10 anos.
 - US\$ 8 Bilhões.
- Lançamento em 4/junho/1996

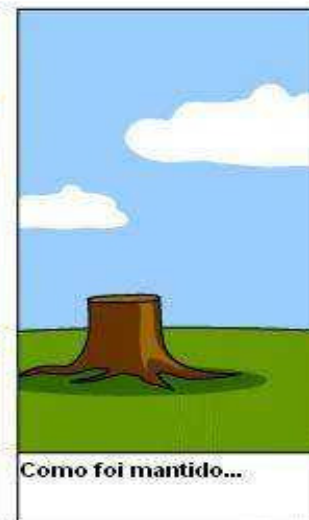
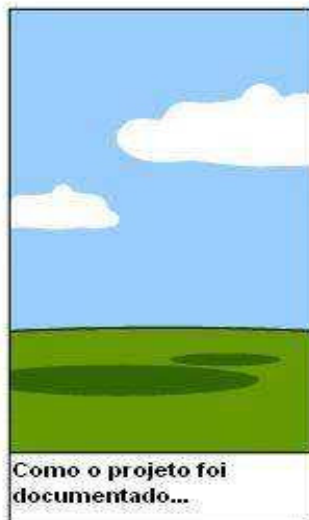
Ver em <http://www.esrin.esa.it/htdocs/esa/ariane/>

Desenvolver, com qualidade ...



- Explosão 40 segundos após a decolagem.
- Destruição do foguete e carga avaliada em US\$ 500 milhões.
- **CAUSA: FALHA DE SOFTWARE!**
 - Shut-down! Ocorreria um *run time error* (out of range, overflow , ou outro) e ambos computadores (principal e back-up) se desligaram.
 - Diagnóstico: Um programa que convertia um valor em ponto flutuante para um inteiro de 16 bits recebeu como entrada um valor que estava fora da faixa permitida.

Desenvolver, com qualidade ...



Aumentar a qualidade do software

- Qualidade relacionada à conformidade do software com os requisitos:
 - Problema: raramente os requisitos estão completos
 - Verificação : Construimos o produto corretamente?
 - Em relação à especificação e aos requisitos
- Qualidade relacionada à satisfação (das necessidades) do usuário:
 - Problema: usuários diferentes.
 - Validação: Construimos o produto correto?
 - Verificado pelo cliente/usuário

Requisitos e necessidades

- Definir com clareza os requisitos de um sistema é um **grande** desafio
- Compromisso entre requisitos do cliente (usuário) e requisitos do sistema
 - “Clientes não sabem o que querem”
 - “Clientes mudam de idéia”, durante o desenvolvimento de sistemas
 - Clientes possuem altas expectativas
 - Alguns requisitos são muito difíceis de especificar (e.g. Usabilidade)
- Especificações são muitas vezes incompletas e inconsistentes
- Mudanças de hardware e software de apoio durante o desenvolvimento.

Desenvolver, com qualidade

- Gerenciar as complexidades do processo de desenvolvimento de software
 - Tecnologia, Metodologia
 - Equipe, Motivação
 - Cronograma, orçamento
 - Estimativas, Riscos
 - Verificação contínua, Qualidade contínua
 - Rastreabilidade, Manutenção
 - Usuário
 - Métricas, mensuração, previsibilidade
 - Aprendizagem contínua, aprendizado organizacional
 - Mudança
 - Etc
- Implica **soluções de equilíbrio**

Software de qualidade

- Cada aplicação possui seus próprios atributos, e seus níveis de aceitação
- Atributos genéricos
 - Manutenabilidade: capacidade de evoluir
 - Confiança e segurança: não pode causar prejuízos físicos/econômicos, usos maliciosos
 - Eficiência: uso adequado dos recursos computacionais
 - Aceitabilidade: público alvo, compreensível, adequado aos seus objetivos

Engenharia de Software

Engenharia

de Software =

Desenvolver

Usar

Reusar

Integrar

n



Linguagens

Ferramentas

Técnicas

Métodos

Modelos

Conceitos

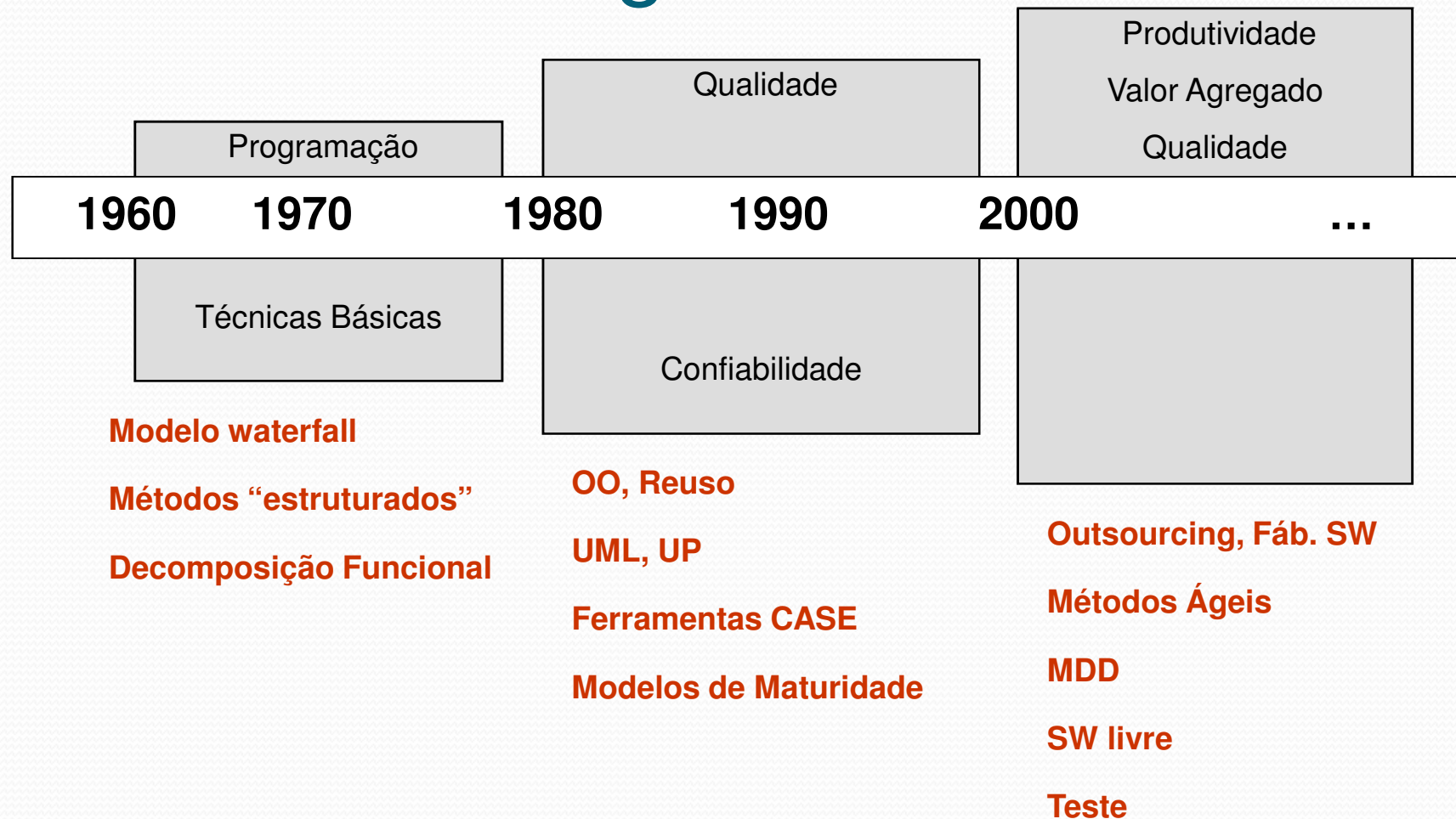
Princípios

Equipes de
Pessoas

n

Combinação de conhecimentos necessários
em **todo o ciclo de vida** do software
para a obtenção de **software de qualidade**

Ênfases da Engenharia de Software



NÃO há uma 'Bala de Prata'!!!

- *Different projects have different needs. Systems have different characteristics, and are built by teams of differing sizes, containing people having differing values and priorities. It cannot be possible to describe the one, best way of producing software.*
- Então, para conhecer esta diversidade, vamos proporcionar:
 - uma visão panorâmica nesta disciplina
 - uma visão em profundidade sobre alguns tópicos durante o curso com outras disciplinas

Pontos Importantes

- ES é uma disciplina de engenharia que se preocupa com todos os aspectos de produção de software
- Software não é apenas um conjunto de programas
- Processo de software define atividades envolvidas no desenvolvimento de software: especificação, desenvolvimento, validação&verificação, evolução
- Existem vários tipos de sistemas, cada qual com seus próprios atributos de qualidade
- Não há técnicas, notações e métodos universais

Para saber mais ...

- Sommerville, Ian. Engenharia de software. 8ª edição. Pearson Education. São Paulo:, 2007.
 - Capítulo 1 – contém um FAQ sobre conceitos básicos de engenharia de software (Fortemente Recomendado)
- Pressman, Roger. Engenharia de Software. Ed. Makron Books, 2006.
 - Capítulo 1 – enfoca a crise do software, mitos e realidades

Créditos

- Este material inclui slides cedidos ou adaptados com permissão de seus autores
 - Marcelo Pimenta
 - Material didático Pearson Education