

# Introdução a Engenharia de Software

Engenharia de Software N

Profa. Karin Becker

Instituto de Informática - UFRGS



# Mercado de Software

- As economias de TODAS as nações desenvolvidas são dependentes de software
- Cada vez mais sistemas são controlados por software
- Os dispêndios com software representam uma fração significativa do PIB em todos os países desenvolvidos

# Mercado Mundial de TI

O Mercado Mundial de TI em 2008 foi de US\$ 1,470 bilhões, sendo os principais mercados:

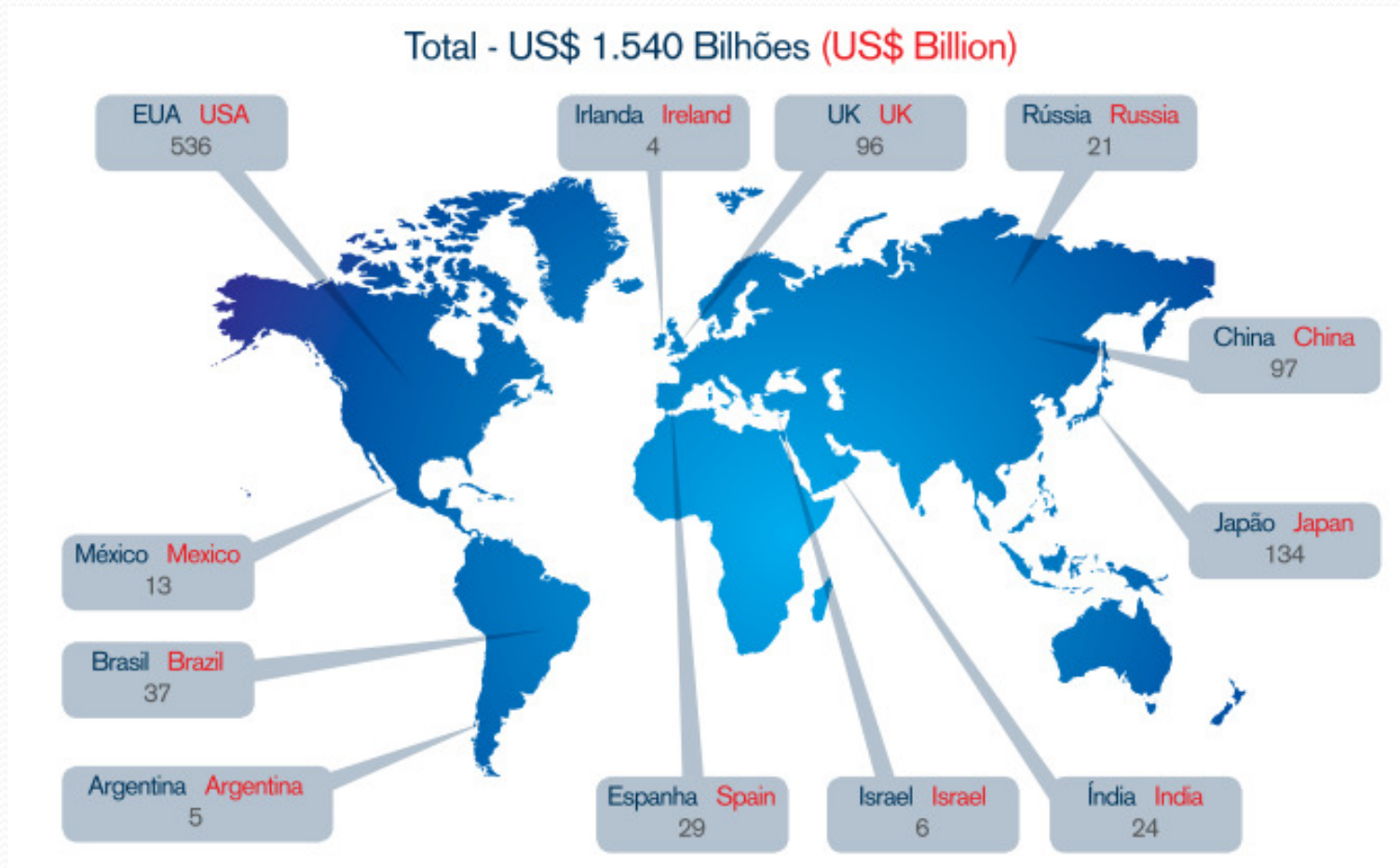
## **O Mercado Mundial de TI - 2008**

País	Mercado (U\$ bilhões)
<div>Hardware 43%      Software 20%      Serviços - Services 37%</div>	
Japão	121,0
UK	92,8
China	69,6
Espanha	32,4
→ Brasil	29,3
Rússia	25,4
Índia	22,0
México	12,2
Israel	5,8
Irlanda	4,3
Argentina	4,2
Colombia	3,4

Fonte: <http://www.abes.org.br/arquivos/MercadoBR-2009-ResumoExec.pdf>



# Mercado Mundial de TI (2010)



Fonte: [www.abes.org.br/UserFiles/Image/PDFs/Mercado\\_BR2011.pdf](http://www.abes.org.br/UserFiles/Image/PDFs/Mercado_BR2011.pdf)

## **O Mercado Brasileiro de Software e Serviços - 2008**

### **Software**

- Mercado Total de Us\$ 5,07 bilhões
- Representa 1,68 % do mercado mundial
- Atendido em 32,5 % por programas desenvolvidos no país
- Exportação de Us\$ 82 milhões em licenças
- Conta com 6.517 empresas dedicadas à exploração econômica

### **Serviços**

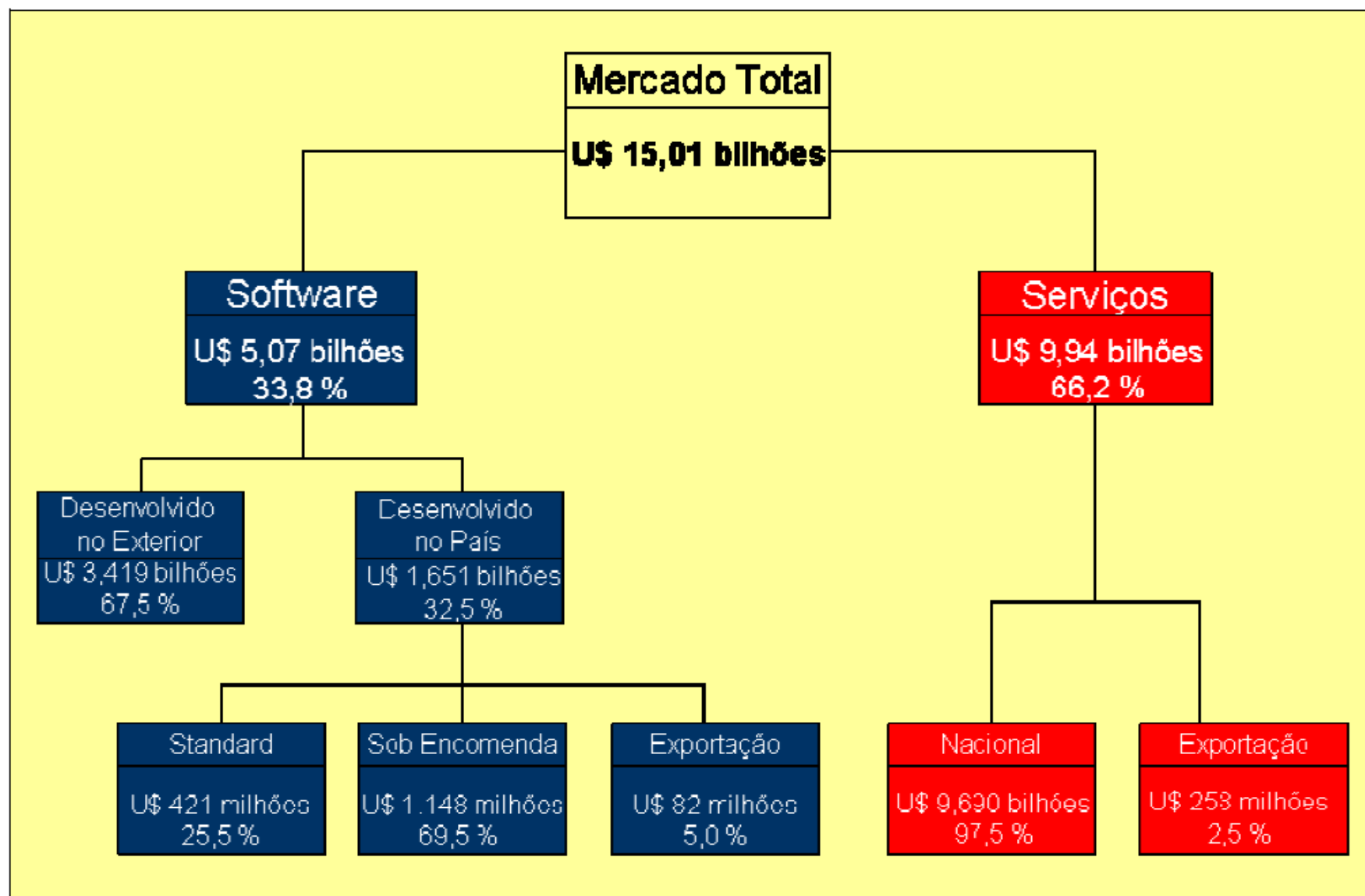
- Mercado Total de Us\$ 9,94 bilhões
- Representa 1,72 % do mercado mundial
- Exportação de Us\$ 258 milhões
- Conta com 1.978 empresas dedicadas à exploração econômica

### **Indicadores Gerais de TI**

- Mercado Total de TI no Brasil Us\$ 29,3 bilhões
- Representa 1,99 % do mercado mundial e 48,0 % do mercado AL
- 11,8 milhões de PC's vendidos em 2008
- 44 milhões de PC's compõe a base instalada
- 58 milhões de usuários da Internet

Fonte: <http://www.abes.org.br/arquivos/MercadoBR-2009-ResumoExec.pdf>


## Principais Indicadores do Mercado Brasileiro - 2008



Fonte: <http://www.abes.org.br/arquivos/MercadoBR-2009-ResumoExec.pdf>

## O Mercado Mundial de Software e Serviços- 2008

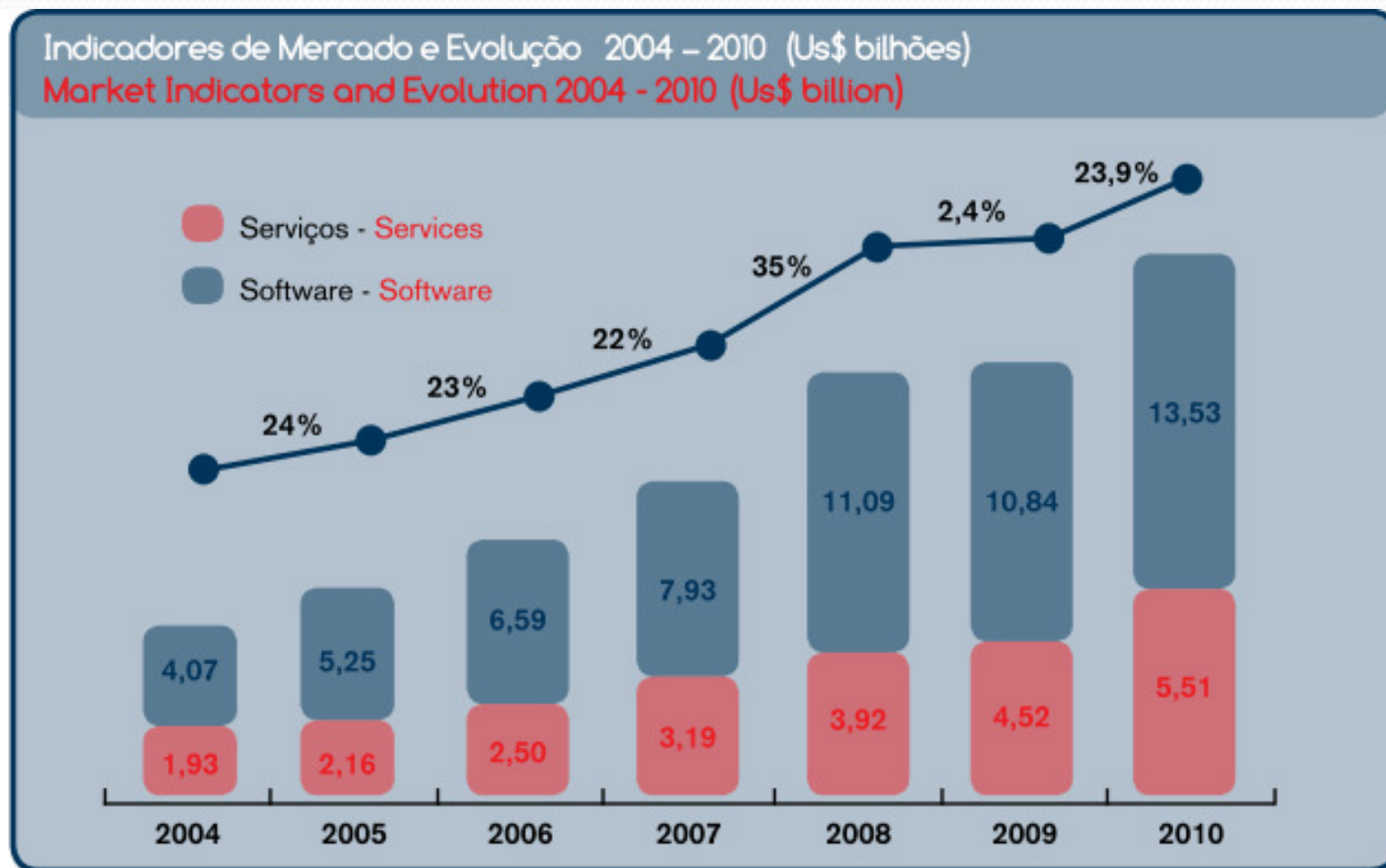
O mercado mundial de software e serviços atingiu em 2008 o valor de U\$ 872,8 bilhões, e o Brasil ficou em 12º lugar no ranking mundial com um mercado interno de U\$ 14,67 bilhões.

País	Volume (Us\$ bilhões)	Participação (%)	08/07
USA	339,6	38,9%	↓
Japão	71,7	8,21%	↓
UK	67,1	7,69%	↓
Alemanha	62,6	7,17%	↑
França	49,8	5,71%	↑
Canadá	24,8	2,84%	↓
Itália	24,1	2,76%	↑
Espanha	19,8	2,27%	↑
Holanda	18,2	2,08%	↑
Austrália	15,6	1,79%	↑
China	15,2	1,74%	↑
 Brasil	14,67	1,68%	↑
Suécia	11,6	1,33%	↑
Suíça	11,25	1,29%	↑
Coréia	8,10	0,93%	↓
ROW	118,7	13,6%	↑
<b>Total</b>	<b>872,8</b>	<b>100%</b>	<b>+15 %</b>

Fonte: <http://www.abes.org.br/arquivos/MercadoBR-2009-ResumoExec.pdf>



# Tendência de crescimento





# A crise de software



[www.jetcart.com.br](http://www.jetcart.com.br)

# A crise de software

- “What have been the *complaints*? Typically, they were:
  - a) Existing *software* production is *done by amateurs* (regardless whether at universities, software houses or manufacturers),
  - b) Existing *software development* is *done by tinkering* or by the human wave (“million monkey”) approach at the manufacturer's,
  - c) Existing *software is unreliable and needs permanent “maintenance”*, the word maintenance being misused to denote fallacies which are expected from the very beginning by the producer,
  - d) Existing *software is messy, lacks transparency*, prevents improvement or building on (or at least requires too high a price to be paid for this).

Last, but not least, *the common complaint* is:

*Existing software comes too late and at higher costs than expected, and does not fulfil the promises made for it.”*

F.L. Bauer, Information Processing (IFIP) 1971 Conference Report (Amsterdam: North-Holland Publishing Co, 1972), I, 530-538; extracted from page 530.



# Engenharia de Software

- 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).

# A crise de software

- *“Software comes too late and at higher costs than expected, and does not fulfil the promises made for it” (Bauer, 1971)*
- *“Software development fails to deliver, and fails to deliver value. This failure has a huge economic and human impact. We need to find a way to develop software” (Kent Beck, 2004)*

- Desvio nos cronogramas
- Projetos cancelados
- Dificuldade de manutenção dos sistemas
  - Custo de mudanças é muito alto, ou gera muitos efeitos colaterais
- Alta taxa de defeitos
  - Falta de credibilidade, não usado
- Falta de compreensão do negócio
  - Sistema não atende o problema de negócio que originou o sistema
- Mudanças nos negócios
  - Prioridade dos problemas de negócio mudam
  - Velocidade crescente
- Ilusão sobre as funcionalidades existentes
- Rotatividade das equipes



# A Crise de Software em Números

- 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, desvio de cronograma, menos funcionalidades
    - Falho: cancelado
  - Análise dos fatores de sucesso/insucesso
  - Metodologia questionada
    - <http://www.umsl.edu/~sauterv/analysis/Standish/glass.pdf>
    - <http://www.few.vu.nl/~x/chaos/chaos.pdf>

# A crise de Software em números

**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

# A Crise de Software em Números

## **Standish Report (2004)**

- Successful Projects: 29%
- Challenged Projects: 53%
- Failed Projects: 18% (5 billion \$ Annualy)

## **Oxford University Regarding IT Project Success (Saur & Cuthbertson, 2003)**

Successful: 16%

Challenged: 74%

Abandoned: 10%

## **Tata Consultancy 2007**

62% of organizations experienced IT projects that failed to meet their schedules

49% suffered from budget overruns

47% had higher-than-expected maintenance costs

41% failed to deliver the expected business value and ROI

33% failed to perform against expectations



# A Crise de Software em Números

**Dynamic Markets Limited 2007 Study of 800 IT managers across eight countries shows that:**

Two Reasons Why IT Projects Fail Reports:

62 % of organizations experienced IT projects that failed to meet their schedules

49% budget overruns

47% higher-than-expected maintenance costs

41% failed to deliver expected business value and ROI

25%+ of all software and services projects are canceled before completion

up to 80 percent of budgets are consumed fixing self-inflicted problems

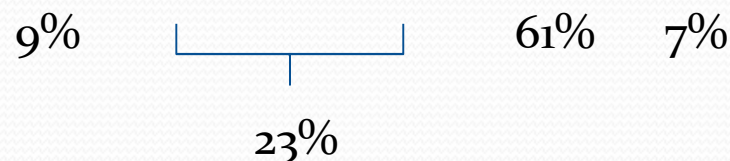


# A Crise de Software em Números

Communications of the ACM Nov 2007: Sauer, Gemino, Reich

We used a non-hierarchical clustering method to combine similar projects based on the three performance variances: schedule, budget and scope. Variances were measured as a percentage of targets as originally planned. A schedule variance of +34% therefore indicates the project was 34% later than originally planned. The results suggested separating the 412 IT projects into 5 basic types described below.

Performance Variance	Type 1: Abandoned Projects n=38	Type 2: Budget Challenged n=21	Type 3: Schedule Challenged n=74	Type 4: Good Performers n=249	Type 5: Star Performers n=30
Performance Variances (Actual as % of Originally Planned) – 100%					
Schedule	N/A	+34%	+82%	+2%	+2%
Budget	N/A	+127%	+16%	+7%	-24%
Scope	N/A	-12%	-16%	-7%	+15%



# Da Crise à Agonia Crônica

- Software :
- Produto não corresponde ao esperado

© 2000 Randy Glasbergen.  
www.glasbergen.com



"THE COMPUTER SAYS I NEED TO UPGRADE MY BRAIN  
TO BE COMPATIBLE WITH ITS NEW SOFTWARE."



# Da Crise à Agonia Crônica

- Processo de Software
  - Processo com baixos graus de satisfação
  - Baixa produtividade dos projetistas de software
  - TI tem o maior nível de stress profissional





# Software: Mitos e Realidade

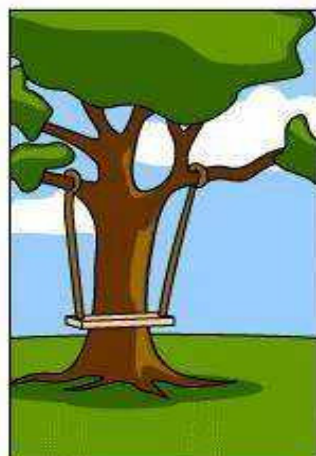
- Ilusão: Desenvolver Software é "SOFT"
  - Economicamente "soft"
  - Intellectualmente "soft"
  - Operacionalmente "soft"
- Realidade: Desenvolver Software é "HARD"
  - Atender requisitos e satisfazer usuário
  - Respeitar orçamento
  - Respeitar cronograma



# Software: Qual é o problema?



Como o cliente explicou...



Como o líder de projeto entendeu...



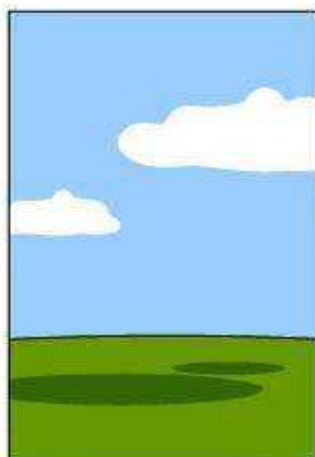
Como o analista projetou...



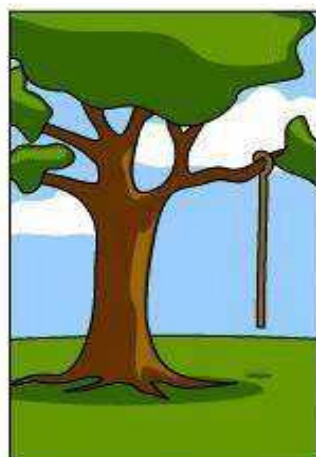
Como o programador construiu...



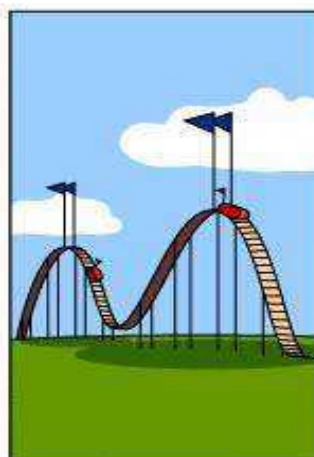
Como o Consultor de Negócios descreveu...



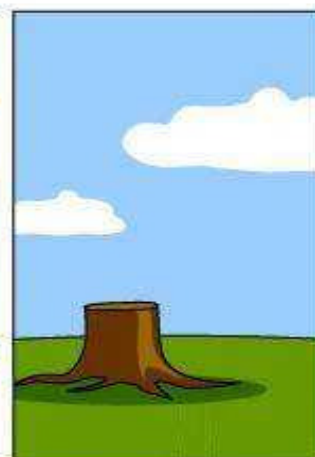
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...



# Eng. SW busca responder ...

- Por que se demora tanto para concluir um projeto ( prazos não cumpridos)?
- Por que custa tanto ?
- Por que não descobrimos os erros antes de entregar o software ao cliente?
  - Por que não entregamos o produto esperado?
  - Por que não entregamos o produto sem falhas?
- Porque temos dificuldade de estimar, medir e acompanhar o progresso enquanto o software está sendo desenvolvido?



# Causas aparentemente óbvias

- Falhas de comunicação entre cliente e equipe de desenvolvimento
  - Pouca comunicação e falha
- Dificuldade de realizar mudanças no software
  - em qualquer etapa, mas principalmente após entrega
  - Dificuldade em planejar para acomodar a mudança
- Falta de procedimentos de verificação da qualidade do que está sendo desenvolvido
- Falta de maturidade (inadequação) dos profissionais e das técnicas utilizadas.

# Causas menos óbvias

- O software não é desenvolvido ou projetado com técnicas de engenharia (é manufaturado de forma artesanal, no sentido clássico)
- Gerentes (ou chefes de projeto) sem (ou com pouco) *background* em desenvolvimento de SW
- Profissionais recebem pouco treinamento formal e frequentemente não conhecem as técnicas e possíveis soluções para seus problemas
  - ou não tem tempo para aplicá-las ☹
- Empresas investem pouco em Eng. Sw.





# Algumas motivações para a Eng. SW

- Atender necessidades dos clientes
- Aumentar a qualidade do software produzido, reduzindo erros e falhas
  - Produto
  - Processo
- Conseguir gerenciar o desenvolvimento de software (incluindo orçamento e cronograma)
- Diminuir a dificuldade de manutenção



# Conhecendo nosso time

- A professora
  - 17 anos de experiência acadêmica
    - ES, BD, Modelagem, Mineração de Dados, Data Warehouse, SOA e Web Services
    - projetos em parceria com empresas
  - 4 anos na indústria de software
    - Líder técnico de projeto, gerente de projeto, gerente de P&D
- Os alunos
  - Quais suas experiências?
  - Quais suas expectativas?