

## *Orientação a Objetos e Java*

Sérgio Soares  
sergio@dsc.upe.br

## *Qualidade de Software*

*“Flight Simulator running on your notebook  
may be more reliable than the software that  
keeps planes from colliding in midair.”*

## *Software de Qualidade Impacto Econômico e Social*

- Segurança de pessoas
  - sistemas de controle de tráfego aéreo
- Disponibilidade de serviços essenciais
  - home banking
  - telefonia

## *Software de Qualidade Impacto Econômico e Social*

- Competitividade das empresas
- Melhores produtos a um menor custo
- Atração de novas empresas para a região
  - investimentos na região
  - arrecadação de impostos

## *Exemplos de falhas em sistemas*

- Sistema de transporte de bagagens do aeroporto de Denver
  - 33 km de trilhos
  - 4.000 carros teleguiados servindo 20 companhias aéreas
  - 100 computadores em rede

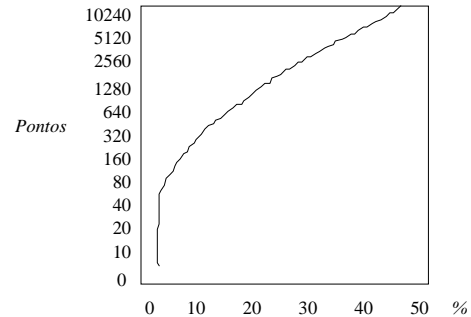
## *Exemplos de falhas em sistemas*

- 1991, pane no sistema telefônico da Califórnia e de toda a costa
  - modificação de apenas três das milhões de linhas de código introduziu um *bug*
- 1986, Therac-25
  - dois pacientes de câncer receberam doses fatais de radioatividade
  - o sistema da máquina tinha erros ocasionados por execuções concorrentes de tarefas

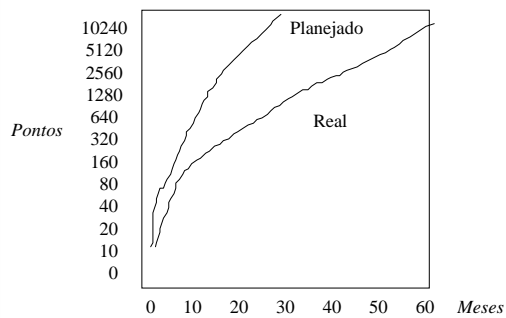
### *Crise de Software*

- 25% dos projetos são cancelados
- o tempo de desenvolvimento é bem maior do que o estimado
- 75% dos sistemas não funcionam como planejado
- a manutenção e reutilização são difíceis e custosas

### *Cancelamentos*



### *Atrasos*



### *Causas da Crise de Software*

- Essências e Acidentes
- Má qualidade das linguagens, ferramentas e metodologias
- Complexidade dos sistemas
- Dificuldade e custos de formalização
  - uma linha de código do sistema de controle de lançamento do ônibus espacial da NASA custa 1.000 dólares

### *Qualidade de Software*

- Correção
- Robustez
- Extensibilidade
- Reusabilidade
- Compatibilidade

### *Outros Fatores de Qualidade*

- Eficiência
- Portabilidade
- Testabilidade
- Integridade
- Facilidade de uso

### *Qualidade de Software (para o Varejo)*

- Correto
  - A loja não pode deixar de cobrar por produtos
- Robusto e altamente disponível
  - A loja não pode parar de vender
- Eficiente
  - O consumidor não pode esperar
  - A empresa quer investir pouco em recursos computacionais (CPU, memória, rede)

### *Qualidade de Software (para o Varejo)*

- Altamente extensível e adaptável
  - A empresa tem sempre novos requisitos (para ontem!)
  - A empresa quer o software customizado do seu jeito (interface, teclado, idioma, moeda, etc.)
- Reusável
  - Várias empresas precisam usar partes de um mesmo sistema

### *Qualidade de Software (para o Varejo)*

- Amigável e fácil de usar
  - A empresa quer investir pouco em treinamento
- Aberto, compatível, de fácil integração com outros sistemas
  - Empresa já tem outros sistemas; controle de estoque, fidelização, etc.

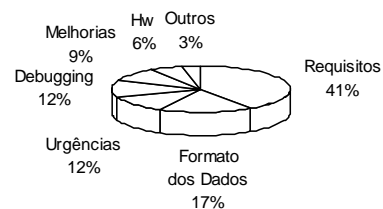
### *Qualidade de Software (para o Varejo)*

- Portável e independente de plataforma (hw e sw)
  - Cada empresa opta por uma determinada plataforma
- Baixo custo de instalação e atualização
  - A empresa tem um grande número de PDVs

### *Impacto de Qualidade em Manutenção*

- 70% do custo de software corresponde a manutenção
- Essência: modificação de requisitos
- Acidente: depuração

### *Manutenção*



### *Produtividade*

- Além de **qualidade**, é importante considerar **produtividade**
- Custo de desenvolvimento reduzido
  - A empresa consumidora quer investir pouco em software
  - A empresa produtora tem que oferecer “software barato”
- Tempo de desenvolvimento reduzido
  - Suporte rápido às necessidades do mercado

### *“Software Barato”*

*Nem tanto resultado de baixos custos de desenvolvimento, mas principalmente da distribuição dos custos entre vários clientes.*

*Reuso, extensibilidade e adaptabilidade são essenciais para viabilizar tal distribuição.*

### *Leitura*

- W. Wayt Gibbs. Software's chronic crisis. Scientific American, September 1994.
- Alan Joch. How software doesn't work. Byte, December 1995.
- Jr. Frederick P. Brooks. No silver bullet: Essence and accidents of software engineering. Computer Magazine, April 1987.