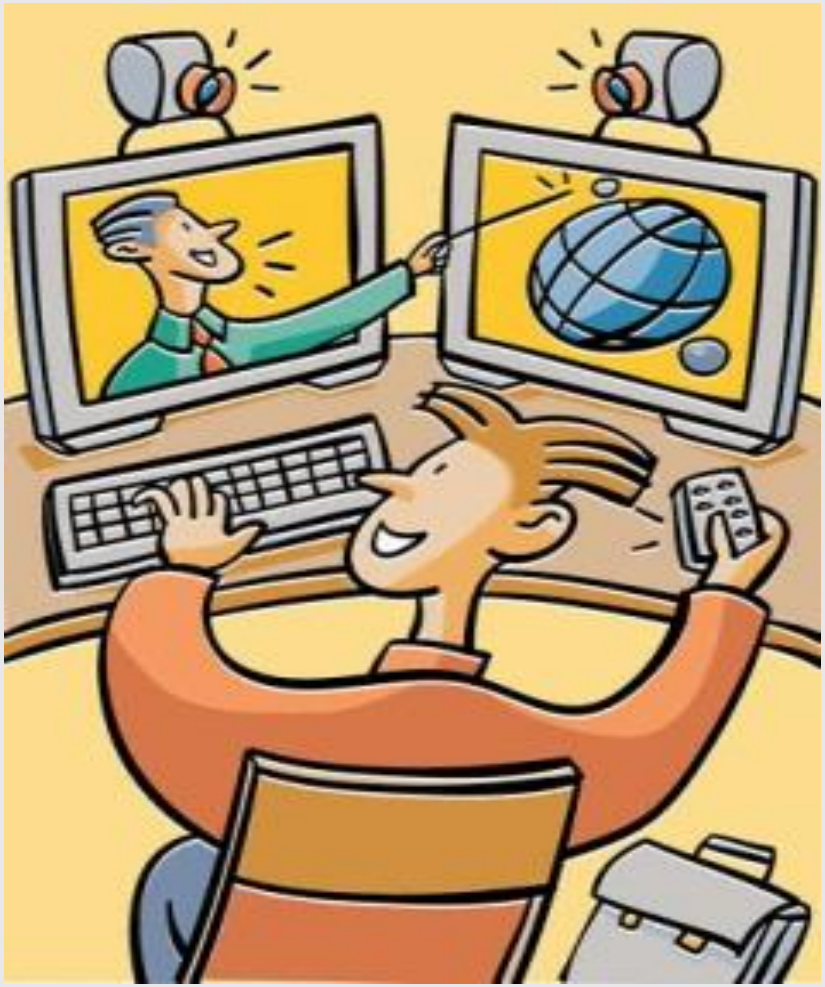
Aula 1 – Apresentação e Introdução

Prof. Paulo R. Nietto

Universidade Anhembi Morumbi
LAUREATE INTERNATIONAL UNIVERSITIES

Quem sou eu?



E quem são vocês?

- Quais seus interesses?
- O que você imagina que aprender nesta disciplina?
- Onde quer chegar?

Objetivos da Disciplina

- Conhecer os tipos e técnicas de testes existentes, sendo capaz utilizá-los da forma mais apropriada de acordo com o de sistema que esta sendo desenvolvido
- Aplicar os conceitos vistos em casos práticos
- Definir casos de testes
- Definir padrões de testes

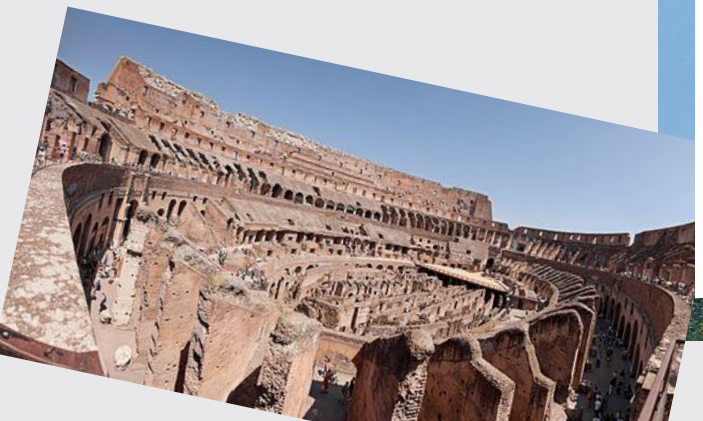
Programação das Aulas

- Ver Plano de Aulas

O que é qualidade?

- O conceito de qualidade é bem subjetivo
- Um produto pode ter qualidade para um cliente e ser desprezado por outro...
- A preocupação com a qualidade existe historicamente a muitos anos...

História



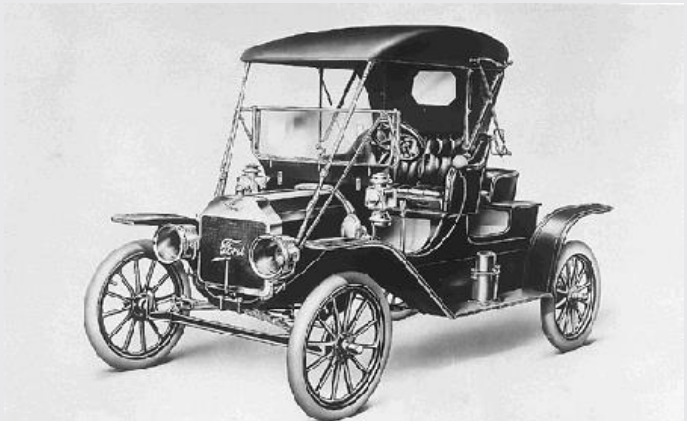
- Os egípcios estabeleceram um padrão de medida de comprimento: o cúbito.
- Equivale ao comprimento do braço do faraó.
- Utilizado em construções:
- O responsável deveria verificar se as medidas estavam corretas.
- Caso acontecesse erros de medição, o responsável poderia ser punido com a morte (Código de Hamurabi).
- O que acontecia com a troca de faraó?
- Cúbito padrão!!!

História

- Em geral, para obter maior precisão é necessário mais tecnologia.
- Revolução industrial:
  - Início da automação;
  - Surgimento de diversas empresas;
  - Aumento da concorrência.

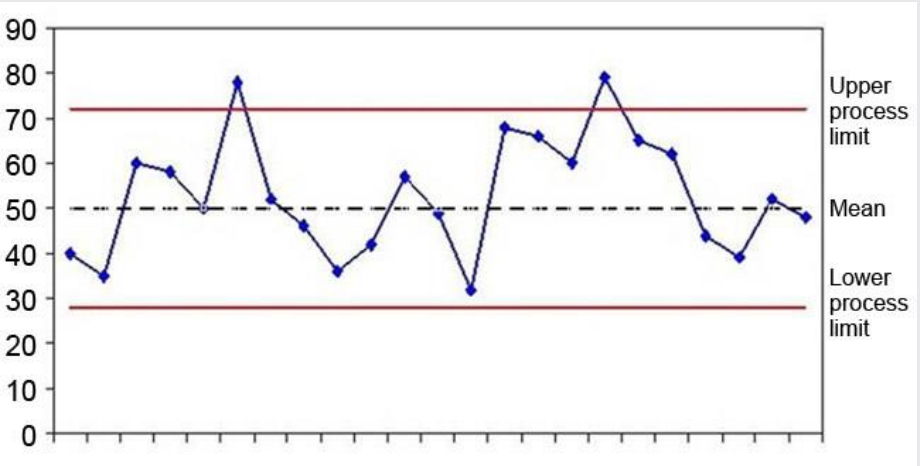
História

- Processo de melhoria contínua:
  - A eficiência tornou-se condição imprescindível;
  - Quem não atendia às necessidade de qualidade do mercado, estava fora.
    - Nos EUA, no início do século XX existiam cerca de 1.800 fábricas de automóveis diferentes.



História

- Na década de 1920 surgiu o controle estatístico de produção:
  - Garantir a qualidade individual de cada peça produzida.
    - De encontro ao trabalho artesanal.
  - Diagramas de controle (*Shewhart chart*):



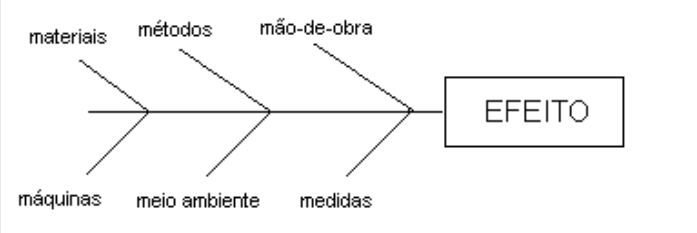
História

- Na década de 1940, surgiram vários organismos ligados à qualidade:
  - ABNT (Associação Brasileira de Normas Técnicas)
  - ASQC (*American Society for Quality Control*)
  - ISO (*International Standardization Organization*)

História

- 2ª Guerra Mundial:
  - Melhoria nas técnicas de manufatura (material bélico);
  - Novos métodos de controle de qualidade.
    - Controle de qualidade: aplicação dos processos de qualidade visando eliminar os produtos que não atingiram o nível exigido.
- Após a guerra, os japoneses começam a investir em suas industrias.

História



- Criaram novos métodos para vender seus produtos por preços menores mas com qualidade igual ou superior aos concorrentes.
  - Método de Taguchi para projeto experimental;
  - Metodologia 5S (整理, 整頓, 清掃, 清潔 e 躾);
    - Utilização, organização, limpeza, higiene e disciplina).
- Diagramas de causa e efeito de Ishikawa.
- Diagramas Espinha de Peixe.
- Desenvolvimento de métodos de controle que buscavam evitar que os defeitos ocorressem, ao invés de eliminar as peças defeituosas.

História

- Os computadores digitais já estavam em uso.
  - Restritos a meios militares e acadêmicos.
- A evolução do hardware permitiu que produtos mais complexos fossem criados.
  - Não haviam escolas e nem cursos de programação.
  - Os programadores não possuíam ferramentas como dispomos hoje.
  - Não existia “oficialmente” a profissão de programador.

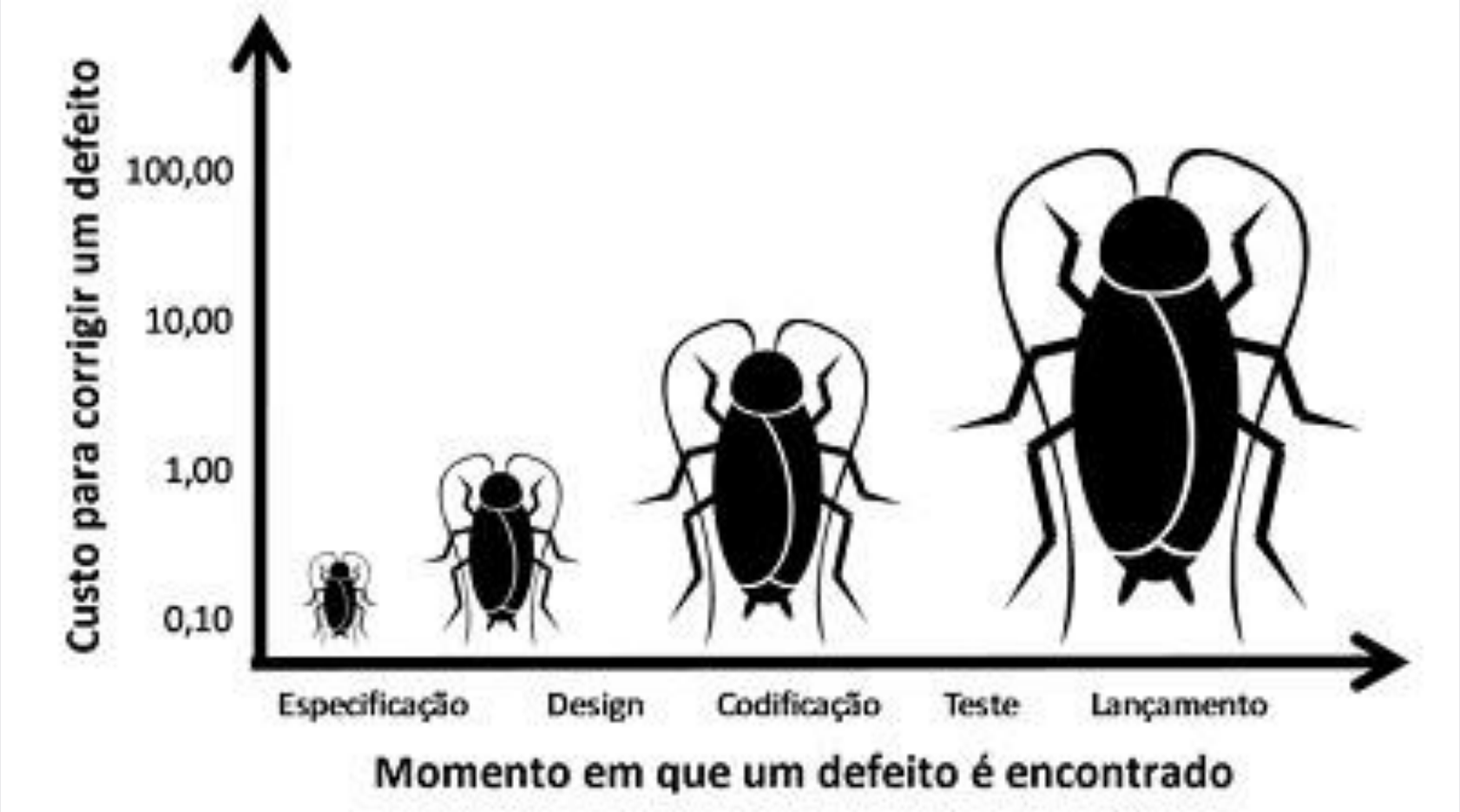
História

- Click to edit Master text styles
  - Second level
    - Third level
      - Fourth level
        - Fifth level

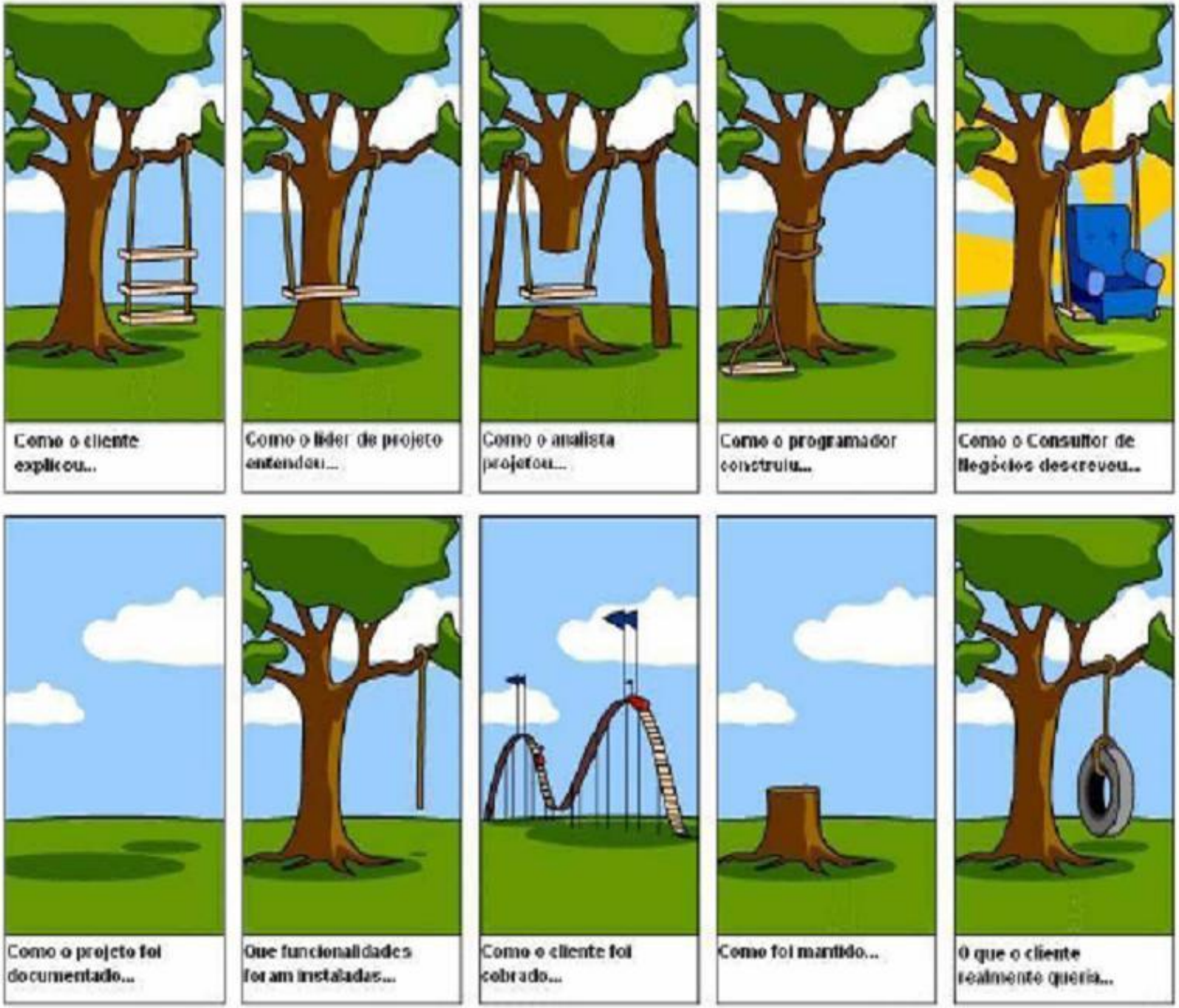
História

- Popularização do computador.
  - Necessário garantir a qualidade dos softwares desenvolvidos.
- Influência negativa sobre a qualidade de software: **complexidade**.
  - Tamanho das especificações;
  - Dificuldade de compreensão;
  - Interações entre os componentes do sistema;
  - Manutenibilidade;
  - Aumenta a possibilidade de erros.

Quanto custa corrigir um defeito?



E por que se preocupar com qualidade de software?


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...

Radiações do Therac-25 (1985)

O Therac-25 era uma aparelho de radioterapia desenvolvido por Canadenses e Franceses. Podia aplicar dois tipos de terapia por radiação: uma com recurso a partículas beta e outra utilizando raios-X. Infelizmente, o sistema operativo do equipamento foi desenvolvido por um programador sem os skills exigidos. Os técnicos que o operavam podiam acidentalmente configurar o Therac-25 para potências muito acima das recomendadas, sem a necessária proteção dos pacientes. Esta falha de programação teve com resultado 5 mortos por radiação e vários pacientes com ferimentos graves.

Segunda feira negra (1987)

No dia 19 de Outubro de 1987, um longo período de valorização dos mercados bolsistas foi assombrado por uma investigação do SEC () por suspeita de negociações com informação privilegiada. Os sistemas de trading estavam preparados para vender os títulos automaticamente se o seu valor fosse menor que determinada quantia. Quando um grupo de investidores começou a vender os seus títulos com receio dos efeitos da investigação, desencadeou uma reação em cadeia de vendas automáticas que lançou o pânico no mercado. Todas as bolsas mundiais foram afetadas e num único dia o índice Dow Jones perdeu 22,6% e o S&P 20%. Ainda hoje esta é a maior perda diária em Wall Street num único dia.

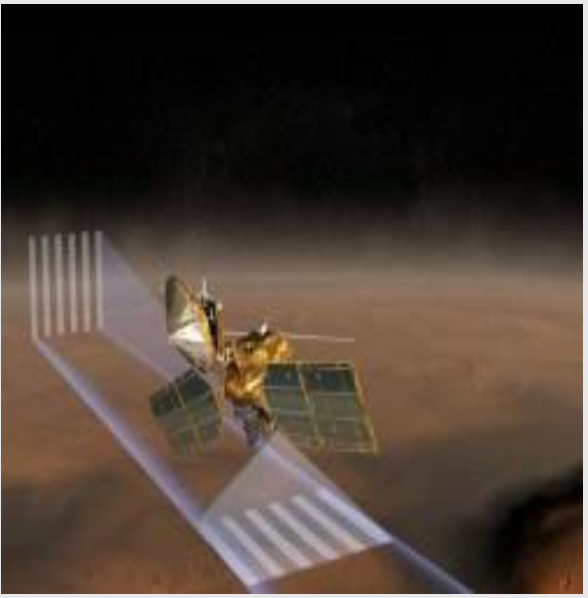
Mísseis Patriot (1991)

Durante a primeira guerra do Golfo foi instalado um sistema Americano de mísseis Patriot para defender as tropas aliadas e os civis Israelitas dos misseis SCUD Iraquianos. **Um erro de arredondamento numa versão inicial do software de gestão destes misseis fez com que o sistema calculasse incorretamente o tempo de reação,** ignorando muitos dos seus alvos. Como resultado, uma bateria de misseis na Arábia Saudita falha a intercepção de um SCUD Iraquiano. Resultado do ataque: uma base Americana totalmente destruída, 28 mortos e ferimentos em mais de 100 pessoas.

Ariane 5 vôo 501 (1996)

Em 1996 o mais recente foguete de lançamento de satélites da Europa, o Ariane 5, estava preparado para ser lançado, reutilizando muito do software que tinha sido utilizado na geração anterior de foguetes. **Contudo, uma vez que a velocidade atingida pelo novo foguete era muito superior a qualquer outra, não foi identificado em testes um erro crítico.** Durante o lançamento a o software fez a tentativa de converter um número de 64 bits dentro de uma variável de 16-bits. Resultado: o foguete se auto-destruiu, mandando para o espaço um investimento de 8 mil milhões de dólares, mais 500 milhões em satélites.

Mars Climate Orbiter (1998)



A Mars Climate Orbiter era uma sonda de 338 quilos lançada pela NASA em Dezembro de 1998 para estudar o clima e a atmosfera de Marte. Em Setembro de 1999, quando a sonda se aproximou da sua posição orbital sobre o planeta, deixou de haver comunicação. As medições de aproximação foram pensadas em newton-seconds (unidade métrica) pela NASA e implementadas no software da sonda em pound-seconds (unidade imperial) pela Lockheed. A manobra de aproximação levou a sonda a tocar na alta atmosfera de Marte, onde se desintegrou.

Bug do ano 2000

O bug do ano 2000 foi o maior de todos os problemas ocorridos até hoje. Muitos dos sistemas criados e desenvolvidos nas décadas de 80 e 90 só contemplavam dois dígitos para mostrar o ano numa data (p.ex. 98 em vez de 1998). Com a chegada do ano 2000 acreditava-se que os sistemas ficariam com 00 nas suas datas, levando as aplicações a assumir 1900 em vez de 2000 como a data atual. Gerou-se o pânico! Foram investidos milhões/bilhões na atualização de grande parte dos sistemas utilizados por bancos, escolas, hospitais, segurança social, transportes, etc.. Na passagem do milénio, nada de extraordinário aconteceu e a esmagadora maioria dos sistemas nem sequer reagiu à data. É difícil de concluir se este sucesso foi o resultado do esforço global para mitigação do problema ou se o problema se quer ocorria...

Apagão na América do Norte (2003)

Afetou aproximadamente 55 milhões de pessoas, no norte do EUA e na zona de Ontário (Canadá), este foi um dos maiores apagões da história. Começou quando uma estação de geração ficou offline com a sobrecarga de pedidos de energia da rede. Isso stressou o resto da rede e as linhas de distribuição sobrecarregadas, aqueceram e começaram a expandir-se. Este efeito cascata foi tão intenso que a capacidade da rede ficou reduzida a 20%. Este resultado não teve origem direta num bug de programação mas na forma como o balanceamento de cargas e a competição por recursos era gerido até então.

Apoio à criança da EDS (2004)

Em 2004, a EDS implementou uma complexa aplicação para a gestão da Child Support Agency (CSA) na Inglaterra. Ao mesmo tempo, a instituição foi reestruturada em termos organizacionais. O software e a organização eram incompatíveis e **foram gerados erros cujo resultado era irreversível**. Com mais de 500 erros graves reportados nos primeiros dias, os resultados foram desastrosos – foram pagas pensões a mais de 1,9 milhões de pessoas; 700 mil não receberam os valores a que tinham direito; 7 mil milhões de dólares de pagamento não foram efetuados; custo final para os contribuintes: mais de mil milhões de dólares.

Modelo Investimento AXA Rosenberg (2009)

A justiça americana condenou em 2010 três entidades pertencentes à AXA Rosenberg por fraude em valores mobiliários, **ocultando um erro grave no código do modelo quantitativo de investimento utilizado pelas instituições para gestão dos ativos dos clientes**. **O erro gerou ao longo de dois anos 217 milhões de dólares em prejuízos para os investidores.** Além de ser obrigada devolver todo o dinheiro aos clientes, a AXA Rosenberg pagou uma multa de 25 milhões e teve que contratar um consultor independente para auditar todos os procedimentos de gestão de carteiras e a sua política de Compliance.

Knight’s Capital (2012)

Após a atualização de um software, em menos de uma hora os computadores da Knight Capital executaram uma série de ordens automáticas que deveriam acontecer ao longo de vários dias. Milhões de ações mudaram de mãos. A perda resultante, quase quatro vezes o lucro da companhia em 2011, mandou a Knight para perto da falência. Um grupo de investidores teve de injetar 400 milhões de dólares na empresa para esta se manter em atividade. Os custos para reverter as situações causadas pelo bug chegaram a quase 500 milhões de dólares.

Para refletir...

Quais são as causas desses problemas relatados, e de outros não-relatados?

Reflexão

"O desenvolvimento de sistemas de software envolve uma série de atividades de produção, em que as oportunidades de injeção de falhas humanas são enormes. Erros podem começar a acontecer logo no começo do processo, onde os objetivos... podem estar errônea ou imperfeitamente especificados, além de erros que venham a ocorrer em fases de projeto e desenvolvimento posteriores...

Por causa da incapacidade que o ser humano têm de executar e comunicar com perfeição, o desenvolvimento de software é acompanhado por uma atividade de garantia de qualidade" [DEU79]

Definições

- Ação humana que produz um resultado incorreto
- Incorreção em um passo, processo ou definição de dados; manifestação no software de um engano cometido pelo desenvolvedor
- Diferença entre o valor obtido e o valor esperado, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução do software
- Incapacidade de fornecer o serviço conforme especificado

Definições


Erro
Uma pessoa comete um erro ...
Defeito
... que cria um defeito no software...
Falha
... que pode causar uma falha na operação

Defeitos no Processo de Desenvolvimento

- A maior parte é de origem humana
- São gerados na comunicação e na transformação de informações
- Permanecem presentes nos diversos produtos de software produzidos e liberados (10 defeitos a cada 1000 linhas de código)
- A maioria encontram-se em partes do código raramente executadas
- Quanto antes a presença do defeito for revelada, menor o custo de correção do defeito e maior a probabilidade de corrigi-lo corretamente
- **Principal causa:** tradução incorreta de informações

Como garantir qualidade de sofware?

- Aplicando um conjunto de **atividades técnicas** durante **todo o processo de desenvolvimento.**

- VV&T - Verificação, Validação e Teste

Garantia de qualidade de software

- **Verificação**:

  "Estamos construindo corretamente o produto?"
- **Validação**:

  "Estamos construindo o produto certo?"
- **Teste**:
  - Examina o comportamento do produto através de sua execução.
  - Garante a qualidade de software e representa a última revisão de especificação, projeto e codificação.

Garantia de qualidade de software

- Aplicação de técnicas adequadas de desenvolvimento
- Realização de revisões técnicas formais
- Testes de software
- Aplicações de padrões
- Controle de mudanças
- Medição da qualidade
- Manutenção de registros de ocorrências relativas à qualidade

Testes de Software

- Testar é submeter o objeto do teste a situações diferentes de forma a confrontar os resultados previstos com os realizados.

- "Teste é o processo de executar um programa ou sistema com a finalidade de encontrar erros"

  (Myers, 1979)

- "Testar é medir a qualidade do sistema".

  (Hetzel, 1984)

Por que testar?

Quando envolver o “tester”?

Atividade

- Escolha um site ou app (individualmente ou em dupla), e:

a) Idealize três exemplos de funcionalidade que você gostaria de testar.
b) Para cada exemplo do exercício anterior, imagine um erro possível.
c) Como analista de teste, que recomendação você daria para quem desenvolveu esse produto?

Questões

1. Qual a importância de se testar um software?

2. Se uma empresa desenvolver um software de forma cuidadosa, utilizando todas as técnicas de engenharia de software para as fases de análise, projeto e codificação, pode-se pular a fase de testes? Por quê?

3. Quando se diz que um teste é bem sucedido?

4. Suponha um programa com 1 milhão de linhas de código desenvolvido por 10 programadores em um mês. Suponha que a fase de testes não encontrou nenhum erro. Pode-se considerar que o programa está livre de erros? Ou pode-se supor que os testes não foram bem feitos? Por quê?

Obrigado

paulo.nietto@animaeducacao.com.br

Universidade
Anhembi Morumbi
LAUREATE INTERNATIONAL UNIVERSITIES