

# Engenharia de Software I

Profa. Denise Neves

[profa.denise@hotmail.com](mailto:profa.denise@hotmail.com)

Introdução  
2018



# Engenharia de Software



- Este curso apresenta métodos e técnicas para o desenvolvimento de software que abordam especificação, modelagem, arquiteturas, verificação e testes de software, bem como para o planejamento e gerenciamento do processo de desenvolvimento. A disciplina de ES se preocupa em sistematizar o desenvolvimento através de modelos, técnicas e ferramentas para o produto e para o processo.



# Engenharia de Software

- Diferença entre computação e engenharia de software

A ciência da computação está relacionada com teorias e fundamentos, engenharia de software está relacionada com a prática e o desenvolvimento de SW.

# Engenharia de Software

## Bibliografia



- Engenharia de Software  
**Ian Sommerville**  
**Pearson Education**
- Engenharia de Software  
**Roger Pressman**  
**6ª edição**  
***McGraw-Hill***

***Artigos sobre Engenharia de Software***

# Engenharia de Software

## Conceitos Básicos



- **Software:**

São programas de computadores, em suas diversas formas, e a documentação associada.

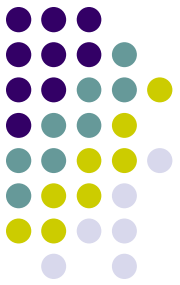
Um programa é um conjunto de soluções algorítmicas, codificadas numa linguagem de programação, executado numa máquina real.

- Software é um produto conceitual e lógico.
- Produtos genéricos : stand-alone. Produzidos e vendidos no mercado.

Produtos sob encomenda : sistemas encomendados.

# Engenharia de Software

## Conceitos básicos



### Características do Software

- Invisibilidade – Software é invisível e invisualizável
- Complexidade – Software é mais complexo do que qualquer outro produto construídos por seres humanos
- Mutabilidade – Existe sempre uma pressão para se fazer mudanças em um software
- Conformidade – O software deve ser desenvolvido conforme o ambiente. Não é o ambiente que deve se adaptar ao software. Se o software esta conforme os requisitos (o ambiente) todo o suporte operacional deve se adaptar ao software.

[Brooks, F. No Silver Bullet]

# Bugs



9/9


0800 Andam started  
 1000 " stopped - andam ✓

1300 (033) MP-MC ~~1.98264000~~  
 (033) PRO 2 2.130476415 ~~(033)~~ 4.615925059(-2)  
 conv 2.130676415

Relays 6-2 in 033 failed special speed test  
 in relay .. 11.00 test.

Relays changed

1100 Started Cosine Tape (Sine check)  
 1525 Started Multy Adder Test.

1545  Relay #70 Panel F  
 (moth) in relay.

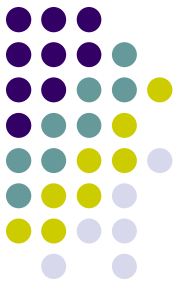
First actual case of bug being found.

1630 Andam started.  
 1700 closed down.

Relay 2145  
 Relay 3370

# Engenharia de Software

## Conceitos Básicos



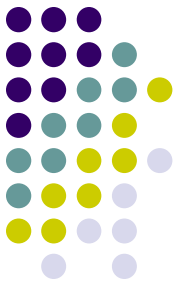
### Falhas de Software

- Ariane 5, 1996
- O foguete explodiu 40 segundos após a sua primeira decolagem.
- Prejuízo de U\$ 500 milhões
- Foi aproveitado um pacote de software de navegação do Ariane 4 que não tinha erros.
- No módulo Sistema de Referencia Inercial uma conversão de valores de 64-bits para 16-bits causou um *operando inválido* que o interrompeu
- A falha no SRI levou o computador de bordo a modificar a trajetória do foguete levando à ruptura de juntas e isto causou a ativação da auto-destruição.
- No Ariane 5, o módulo SRI só deveria ter sido utilizado antes do lançamento.



# Engenharia de Software

## Conceitos Básicos

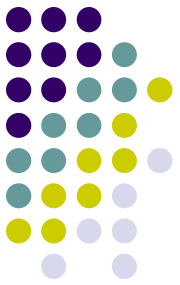


### Falhas de Software

- London Ambulance System - despacho de ambulâncias em Londres, 1992.
- Morte de pessoas que não foram socorridas em tempo.
- Problema de Gerência de Software:
  - Responsáveis contrataram uma empresa desconhecida cujo valor cobrado era menor que os cobrados pelas empresas de renome.
  - Colocaram o sistema no ar sem os devidos testes.
  - Não foi feita uma migração correta do sistema antigo para o novo.

# Engenharia de Software

## Conceitos Básicos



### Falhas de Software

- Máquina de radioterapia com software controlador (1985 - 87).
- O controle de segurança feito pelo hardware em máquinas anteriores foi removido e passou a ser feito pelo software.
- O software falhou na tarefa de, ao mesmo tempo, manter invariantes essenciais: o feixe de elétrons e o dispositivo que controla a concentração do feixe em níveis seguros.
- Diversas mortes em decorrência de queimaduras.  
O programador não tinha experiência em programação concorrente.

# Engenharia de Software

## Conceitos Básicos



- Bug do Milênio
  - Datas no formato : dd/mm/aa  
Na virada do ano 2000 -> 00 que vai ser entendido como 1900
- O *bug* pode aparecer em hardware, software (sistemas e aplicativos), bases de dados, arquivos, *scripts*, enfim, onde houver processamento de datas, ali pode estar o problema.
- No caso dos computadores e programas, a solução é relativamente simples (se não considerarmos os custos) - atualização (*upgrade*) de software, hardware e utilização dos remendos (*patches*) que estão sendo lançados pelos fabricantes.

# Engenharia de Software

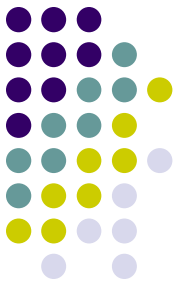


## Documento : "Testing for The Year 2000 Deadline"

- Se uma aplicação não é mais necessária, livre-se dela;
- Se você está usando atualmente uma aplicação que planeje desativar antes do ano 2000, então comece a fazê-lo gradualmente;
- Se uma aplicação for considerada em conformidade com o ano 2000, realize alguns testes preliminares e passe para o próximo software;
- Contacte seu fornecedor de software para verificar se pode ser feito o *upgrade* das aplicações;
- Determine se o software pode ser trocado por um outro pacote adequado e faça a cotação do mesmo;
- Se você tiver alguma aplicação que precise ser convertida, comece imediatamente;
- Estabeleça um plano para ajudá-lo a navegar por tudo que precisa ser feito, e imponha uma linha de tempo rígida para cada uma das atividades.

# Engenharia de Software

## Documento "Testing for The Year 2000 Deadline"

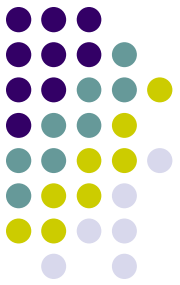


Atividades :

- Determinar as correções que devem ser aplicadas a cada elemento com problemas;
- Determinar os custos para a solução;
- Gerar um relatório de custos;
- Priorizar e aplicar as correções com base na análise de riscos;
- Documentar as correções aplicadas para cada um dos itens;
- Caso as correções afetem outros setores, comunicá-los sobre os planos de correção;

# Engenharia de Software

## Documento "Testing for The Year 2000 Deadline"

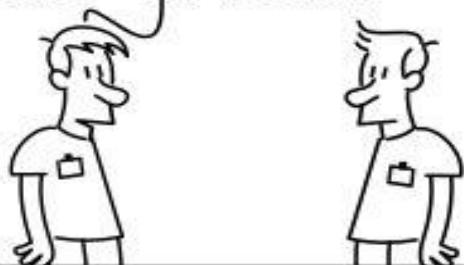


### Atividades :

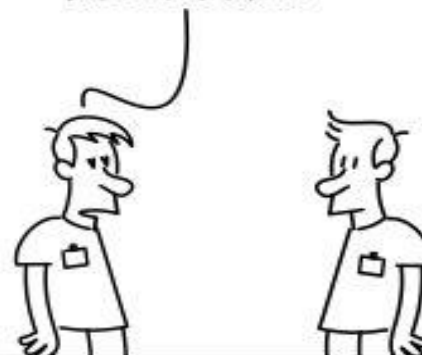
- Estabelecer planos de contingência para os elementos chaves, para o caso de haver problemas na passagem para o ano 2000;
- Estabelecer processos manuais para processos automatizados;
- Manter cópias impressas de informações importantes;
- Garantir o processo de backup;
- Deixar um estoque razoável de materiais, prevendo falhas nos fornecedores (Ex.: *toner* de impressoras, papéis para impressão, disquetes, fita DAT, material de escritório, água potável, baterias, pilhas, materiais de limpeza, etc.).



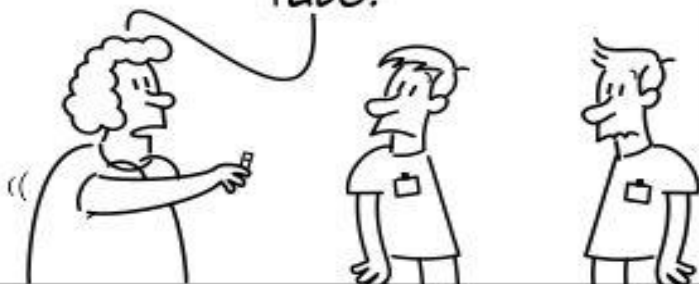
O BUG DO MILÊNIO GEROU  
UMA ENORME ONDA DE  
PÂNICO COLETIVO. ALGUNS  
PENSARAM QUE A  
HUMANIDADE IRIA VOLTAR À  
IDADE DA PEDRA.



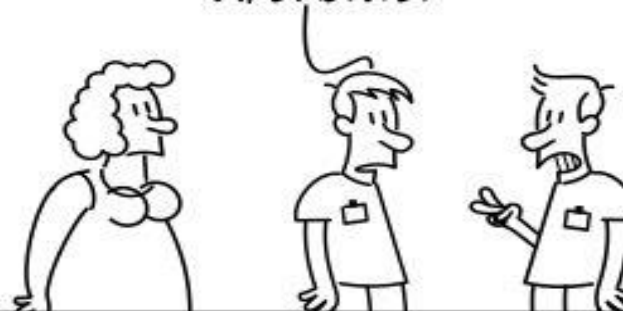
MAS NO FIM DAS  
CONTAS NADA DE MAIS  
ACONTECEU.



FIQUEI SABENDO QUE O  
ESTOQUE DESSE TAL DE  
IPV4 VAI ACABAR! SALVA  
UNS AQUI NO MEU PEN  
DRIVE ANTES QUE ESGOTE  
TUDO.

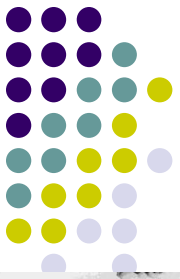


AGORA É CRUZAR  
OS DEDOS PARA  
QUE EM 2038 O  
NEGÓCIO SEJA  
DIFERENTE.

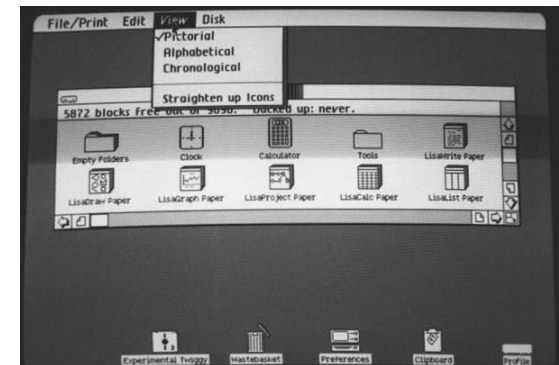
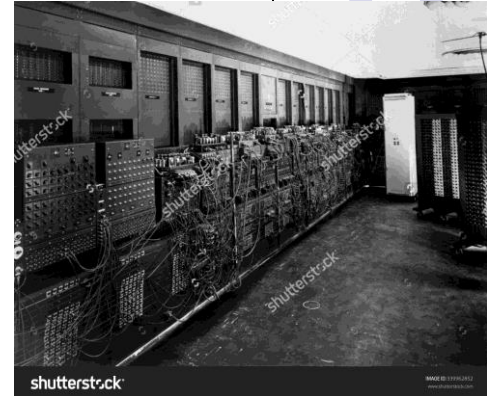


# Engenharia de Software

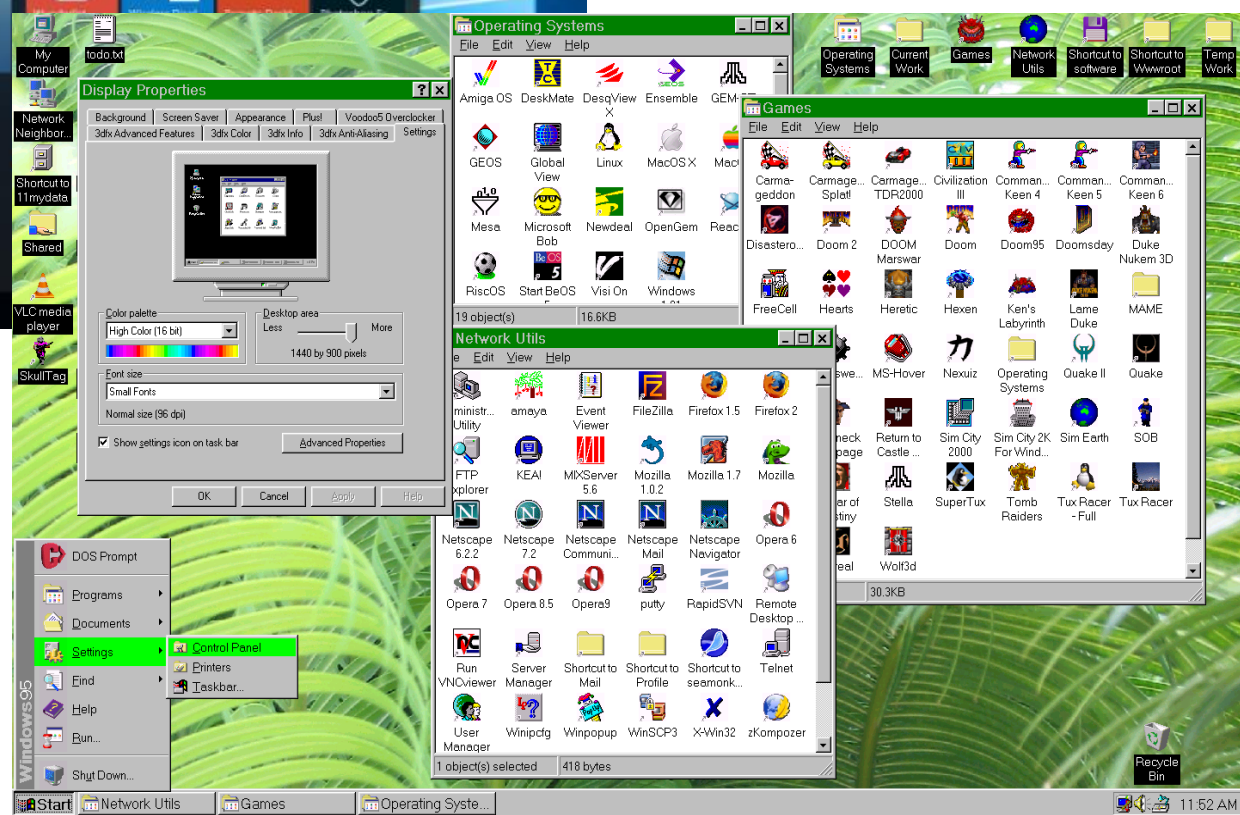
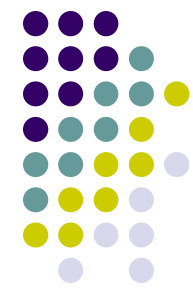
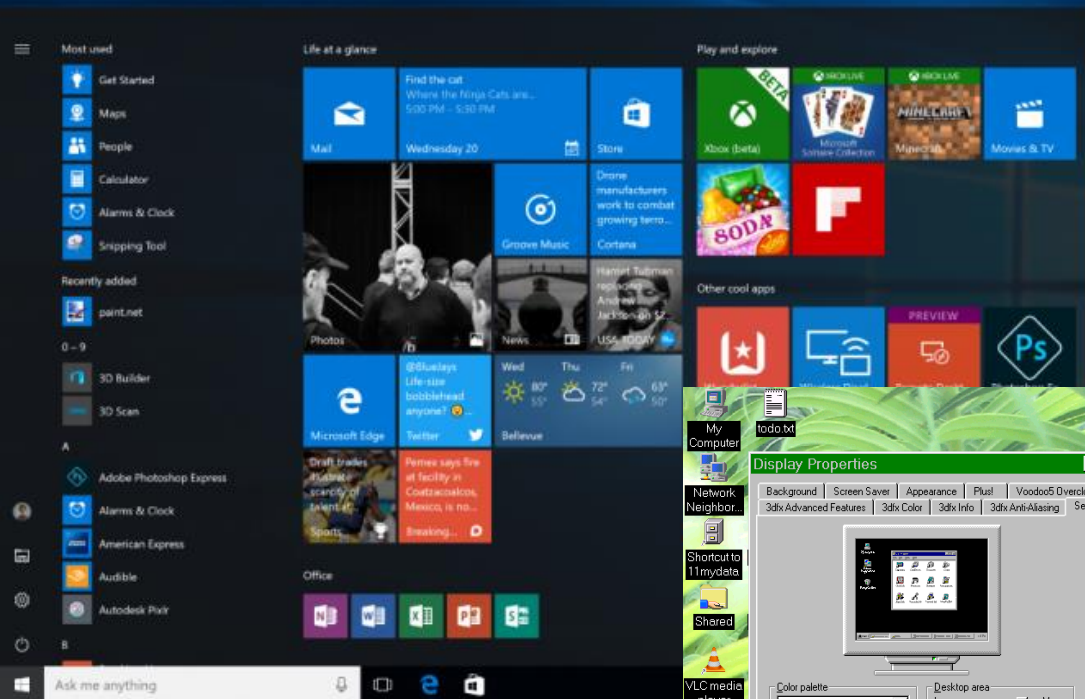
## Conceitos Básicos:Evolução



- Os primeiros anos (1950 a início dos 60)
  - Aplicações científicas e de engenharia
- A segunda era (1960 a meados de 70)
  - Aplicações comerciais em grande-porte (sistemas de informação BD)
- A terceira era (meados de 70 e década de 80)
  - Aplicativos pessoais em microcomputadores





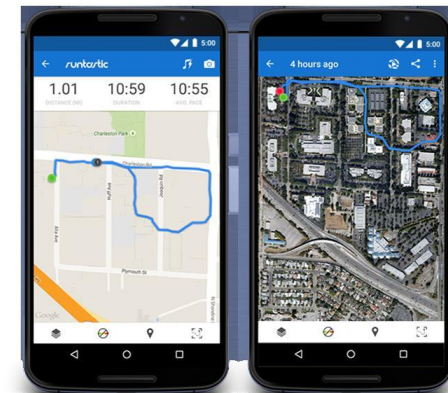
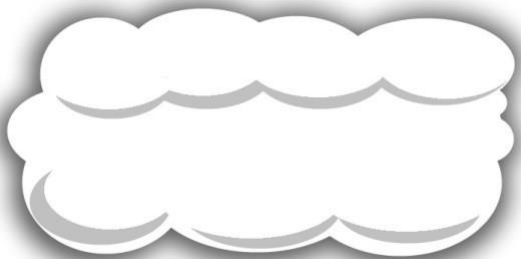


A quarta era (meados de 80 a meados de 90)  
– Aplicativos com Interfaces Gráficas

- A quarta era (meados de 80 a meados de 90)
  - Aplicativos com Interfaces Gráficas
  - Redes e Arquitetura Cliente-Servidor
- A quinta era (de meados de 90 a ???)
  - Software Distribuídos, Internet e Intranets



- Sexta era??
  - Computação Pervasiva, Móvel e Ubíqua



# Engenharia de Software

## Conceitos Básicos



### Desafios para a Indústria de Software

- Sistemas legados
  - Sistemas antigos, mas de extrema importância para uma organização e que funcionam bem.
  - Linguagens antigas – falta de pessoal
  - Específico de plataformas obsoletas
- Heterogeneidade
  - Os sistemas são distribuídos e incluem diversidade de hardware e plataforma operacional
- Entrega
  - Existe sempre uma pressão forte para diminuir o tempo de entrega.

# Engenharia de Software



- É uma disciplina da engenharia dedicada a todos os aspectos da produção de software.
- Engenheiros de software devem adotar uma abordagem sistemática e organizada para o seu trabalho e usar técnicas e ferramentas apropriadas, de acordo com o problema a ser resolvido, e com as restrições e recursos disponíveis.
- Responsabilidade profissional e ética.

Fonte: Ian Sommerville

# Referência Bibliográficas



- PRESSMAN
- SUMMERVILLE