



Engenharia de Software

Alessandra Marta de Oliveira
alessandreia@gmail.com



Histórico (era pré-ES)

- 1940s: Primeiro computador eletrônico de uso geral – ENIAC
 - Custo estimado de US\$ 500.000,00
 - Início da programação de computadores
- 1950s: Primeiros compiladores e interpretadores
- 1960s: Primeiro grande software relatado na literatura – OS/360
 - Mais de 1000 desenvolvedores
 - Custo estimado de US\$ 50.000.000,00 por ano
- 1968: Crise do software – nasce a Engenharia de Software

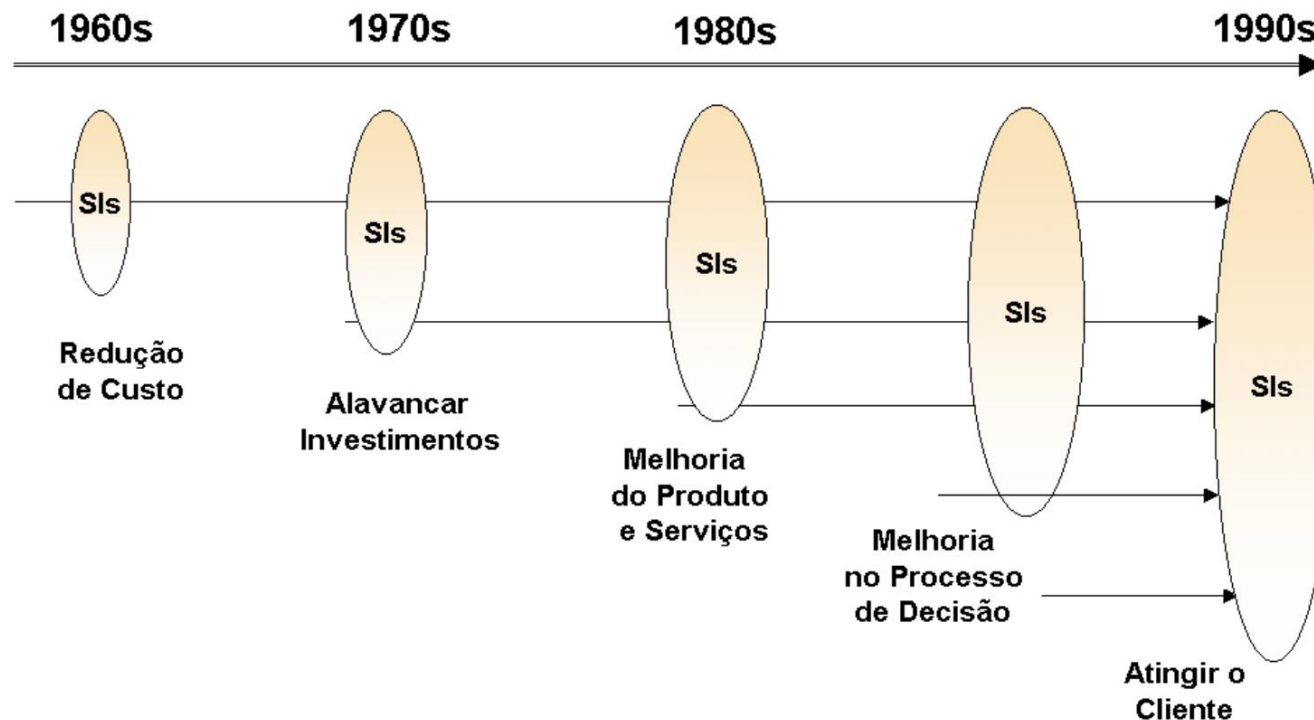


Histórico (era pós-ES)

- 1970s:
 - Lower-CASE tools (programação, depuração, colaboração)
 - Ciclo de vida cascata
 - Desenvolvimento estruturado
- 1980s:
 - Ciclo de vida espiral
 - Desenvolvimento orientado a objetos
- 1990s: Upper-CASE tools
 - Processos
 - Modelagem
- Atualmente:
 - Métodos ágeis
 - Desenvolvimento dirigido por modelos
 - Linhas de produto
 - Experimentação

A Importância do Software

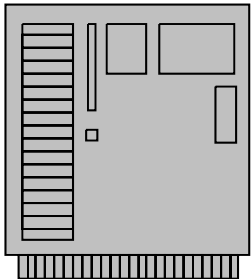
- A complexidade do software vem crescendo ao longo do tempo





A Importância do Software

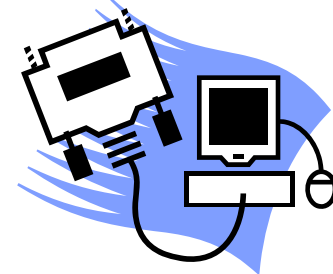
- As três primeiras décadas da era do computador:
 - Principal desafio: Desenvolver um hardware que reduzisse o custo de processamento e armazenagem de dados
 - O software na década de 60 era um único programa executado no mainframe.
 - Realizava uma única função bem definida





A Importância do Software

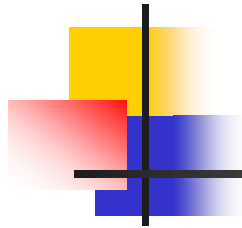
- Ao longo da década de 1980:
 - Avanços na microeletrônica resultaram em maior poder de computação a um custo cada vez mais baixo
 - O sw atingia o processamento distribuído
 - Softwares cada vez mais complexos realizavam comunicações entre módulos de programa (Client/Server)





A Importância do Software

- Década de 1990 em diante
 - Principal desafio: Melhorar a qualidade (e reduzir o custo) de soluções baseadas em computador – soluções que são implementadas com software



A Importância do Software

- O poder de um mainframe da década de 1980 agora está a disposição obre uma escrivaninha
 - As assombrosas capacidades de processamento e armazenagem do moderno hw representam um grande potencial de computação
 - O sw é o mecanismo que nos possibilita aproveitar e dar vazão a esse potencial



A Importância do Software

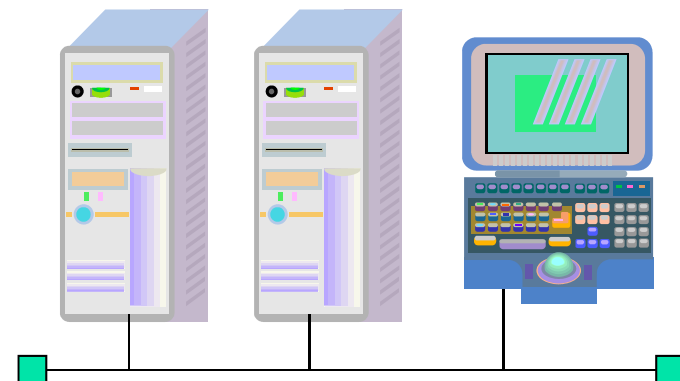
- No fim da década de 90 a Internet levou as operações de negócio para o mundo virtual de forma ampla
- Software cada vez mais complexos e vitais para as organizações (ERP, etc...)





A Importância do Software

- Hoje a arquitetura dos softwares é em três camadas (cliente, aplicação e servidor)
- Uso de componentes que se comunicam entre si e são reutilizáveis (re-uso de software)





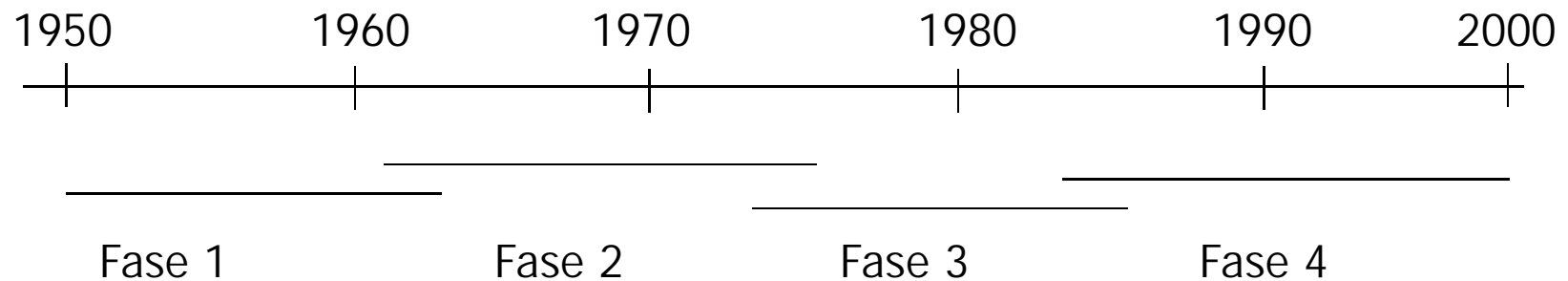
A Importância do Software

- Além do crescimento do software, seu desenvolvimento passa agora pelo uso de várias tecnologias que aumentam ainda mais sua complexidade:
 - Firewall, Criptografia, Web Services, entre várias outras...
 - Tal complexidade exige softwares com maior qualidade
- E mais...
 - Qualidade, Desenvolvimento baseado na Web, OO e componentes, Gestão do conhecimento...
 - Web Semântica, Agentes, SOA e Aspectos ...
 - CRM, DW, etc...



Papel evolutivo do Software

- Quase 5 décadas de desenvolvimento





Papel evolutivo do Software

- Fase 1

- Hardware caro
- Software em segundo plano/customizado
- Distribuição muito limitada
- Orientação **BATCH**



Papel evolutivo do Software

- Fase 2
 - Multiusuário
 - Tempo Real
 - Banco de Dados
 - Produção de Software



Papel evolutivo do Software

- Fase 3

- Sistemas Distribuídos
- Inteligência embutida
- Hardware de Baixo Custo
- Impacto de consumo



Papel evolutivo do Software

- Fase 4

- Sistemas Desktop poderosos
- Sistemas Especialistas
- Tecnologia Orientada a Objeto
- Computação paralela
- Redes Neurais Artificiais
- ...



Software:

Conceitos e aplicações



Conceitos



- O que é Software?
 - Instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados
 - Estruturas de dados que possibilitam que os programas manipulem adequadamente a informação
 - Documentos que descrevem a operação e o uso dos programas
 -



Aplicações do Software

- O Software pode ser aplicado a qualquer situação em que um conjunto previamente especificado de passos procedimentais (algoritmo) tiver sido definido
- Notáveis exceções a essa regra são o software de sistemas especialistas e o software de rede neural



Características do Software

- Software é desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico
 - os custos do software estão centrados no trabalho de engenharia, no processo
- Software não se “desgasta”
 - o software se “deteriora”
 - a manutenção do software é complexa
- A maioria dos software é feita sob medida
 - o crescente uso da OO e técnicas de reutilização tem contribuído para o reuso de componentes



Categorias do Software

- Software Básico, Software de Tempo Real, Software Comercial, Software Científico e de Engenharia, Software Embutido ou Embarcado, Software de Computador Pessoal, Software de Inteligência Artificial, Software do Tipo Hipermídia,.....



Crise do Software

- Conjunto de problemas que são encontrados no desenvolvimento de software, como:
 - software que não funcionam
 - manutenção de um volume crescente de software existente
 - atendimento a uma demanda cada vez maior, etc.



Crise do Software

- Relatório do Standish Group (Caos - 1995)
 - Em 1995 os Estados Unidos gastaram \$81 milhões em projetos de software que foram cancelados
 - 31% dos projetos foram cancelados antes de estarem concluídos
 - 53% excederam mais de 50% da estimativa de custo
 - Somente 9% dos projetos das grandes empresas foram entregues em tempo e estimativa
 - Em pequenas empresas os números são de 28% e 16% respectivamente



Problemas

- Falta de dados históricos sobre o processo de desenvolvimento
- Insatisfação de clientes
- Qualidade suspeita do software
- Manutenção difícil



Projetos de Software

- O desenvolvimento de software é ainda imprevisível
 - Somente 10% dos projetos de software são entregues com sucesso dentro das estimativas de orçamento e custo
 - O nível de software jogado fora e que tem necessidade de re-trabalho é um indicativo de processo imaturo



Mitos do Software





Mitos do Software

- Antigas atitudes e hábitos são difíceis de serem modificados, e os remanescentes dos mitos de sw ainda merecem crédito quando nos movimentamos em direção a mais uma década de existência do sw



Mitos Administrativos

- Os gerentes que tem responsabilidade pelo sw freqüentemente se encontram sob pressão para manterem orçamentos, evitarem que os prazos saiam de controle e melhorarem a qualidade
 - Como uma pessoa que está se afogando e agarra-se a uma palha, o gerente muitas vezes se agarra à crença de um mito de sw caso esse mito atenuar (mesmo que temporariamente) a pressão que pesa sobre ele



Mitos Administrativos

- Mito:

- Temos um manual repleto de padrões e procedimentos para a construção de sw. Isso não oferecerá ao meu pessoal tudo o que eles precisam saber?

- Realidade:

- O manual pode existir, mas ele será usado? Os profissionais de sw tem conhecimento de sua existência? Ele reflete a moderna prática de desenvolvimentos de sw? É completo?



Mitos Administrativos

- Mito:
 - Meu pessoal tem ferramentas de desenvolvimento de sw de última geração, afinal de contas compramos os mais novos computadores



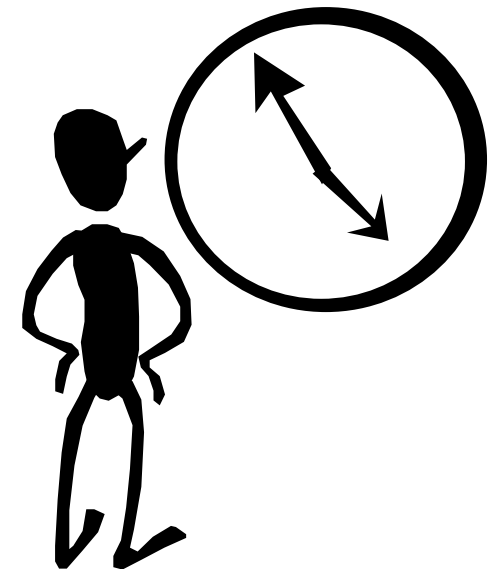
Mitos Administrativos

- Realidade:
 - É preciso muito mais do que o último modelo de computador para se fazer um desenvolvimento com alta qualidade. Ferramentas CASE por exemplo, são mais importantes do que o hw para se conseguir boa qualidade e produtividade, mas muitos desenvolvedores ainda não as utilizam



Mitos Administrativos

- Mito:
 - Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e tirar o atraso





Mitos Administrativos

- Realidade:
 - O desenvolvimento de sw não é mecânico como a manufatura. Acrescentar pessoas em projetos atrasados torna-o ainda mais atrasado
 - Quando novas pessoas são acrescentadas, as pessoas que estavam trabalhando devem gastar tempo educando os recém-chegados, o que reduz o tempo despendido num esforço de desenvolvimento produtivo. Pessoas podem ser acrescentadas mas de forma planejada e coordenada

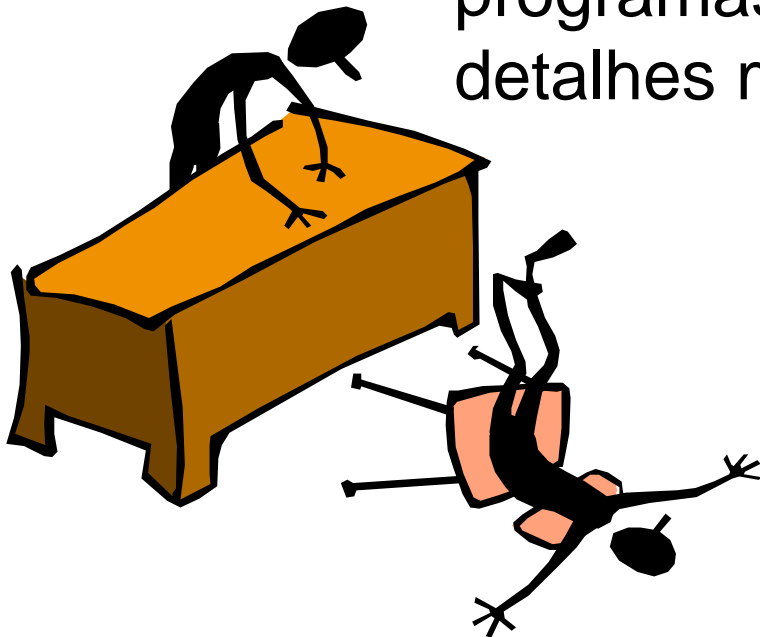


Mitos do Cliente

- Um cliente que exige sw pode ser um departamento de marketing, um grupo técnico da sala de reuniões, uma empresa de fora,
 - Em muitos casos, o cliente acredita em mitos, porque os gerentes e profissionais pouco fazem para corrigir a desinformação
 - Os mitos levam a falsas expectativas (por parte do cliente) e em última análise, à insatisfação com o desenvolvedor

Mitos do Cliente

- Mito:
 - Uma declaração geral dos objetivos é suficiente para se começar a escrever programas – podemos preencher o detalhes mais tarde





Mitos do Cliente

- Realidade:
 - Uma definição inicial ruim é a principal causa de fracasso dos esforços de desenvolvimento de sw. Uma definição formal e detalhada do domínio da informação, função, interfaces, validação, ..., é fundamental. Essas características podem ser determinadas somente depois de cuidadosa comunicação entre o cliente e o desenvolvedor



Mitos do Cliente

- Mito:
 - Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodados pois o sw é flexível



Mitos do Cliente

- Realidade:
 - Os requisitos de sw se modificam sim, mas o impacto de mudança varia de acordo com o tempo em que ela é introduzida
 - Se uma séria atenção for dada à definição inicial, os primeiros pedidos de mudança podem ser acomodados facilmente. O cliente pode rever as exigências e recomendar modificações sem causar grande impacto



Mitos do Cliente

- Realidade (cont.)
 - Quando são exigidas mudanças durante o projeto, o impacto sobre os custos eleva-se rapidamente. Recursos foram comprometidos e uma estrutura de projeto foi estabelecida. A mudança pode causar sublevações que exijam recursos adicionais e grandes modificações, isto é, custo adicional
 - Mudanças na função, desempenho, interfaces, ..., durante a implementação (código e teste) exercem forte impacto sobre o custo



Mitos do Profissional

- Mitos têm sido fomentados por várias décadas de cultura de programação. Durante os primórdios do sw, a programação era vista como uma ferramenta de arte. Velhas maneiras e atitudes dificilmente morrem



Mitos do Profissional

- Mito:

- Assim que escrevermos o programa e o colocarmos em funcionamento, nosso trabalho estará completo

- Realidade:

- Alguns dados indicam que entre 50 e 70% de todo o esforço gasto num programa serão despendidos depois que ele for entregue pela primeira vez ao cliente

Mitos do Profissional

- Mito:
 - Enquanto não tiver o programa “funcionando”, eu não terei nenhuma maneira de avaliar sua qualidade





Mitos do Profissional

- Realidade:
 - Um dos mecanismos mais efetivos de garantia de qualidade pode ser aplicado desde o começo de um projeto – a revisão técnica formal, que tem sido considerada mais eficiente do que a realização de testes para a descoberta de certas classes de defeitos de sw

Mitos do Profissional

- Mito:
 - A única coisa a ser entregue em um projeto bem sucedido é o programa funcionando





Mitos do Profissional

- Realidade:
 - Um programa funcionando é somente uma parte de uma configuração de sw que inclui vários elementos: plano, especificação de requisitos, projeto, estrutura de dados, teste, ...
 - A documentação forma os alicerces para um desenvolvimento bem sucedido e, o que é mais importante, fornece um guia para a tarefa de manutenção de sw.