

Engenharia de Software

Aula 01

Apresentação Pessoal

Apresentação da Disciplina

Fundamentos da Engenharia de Software

Evolução do Software

Crise do Software

Mitos da Engenharia de Software

Prof. MSc. Vinícius Camargo Andrade

Formação



Graduação:

Tecnologia em Análise e Desenvolvimento de Sistemas - UTFPR
(2008 - 2010)

Iniciação Científica:

Projeto: Identificação dos Pontos de Estabilidade e Flexibilidade no Modelo de Requisitos dos Métodos de Preço de Venda. (2009-2010)
Orientadora: Prof^a Dr^a Simone Nasser Matos.

Mestrado:

Projeto: Transformação de Modelos de Diagrama de Sequência UML contemplando restrições de Tempo e Energia para Rede de Petri Temporal. UFPR (2011-2012)
Orientadora: Prof^a Dr^a Letícia Mara Peres.

Atuação Profissional



2013 – 2014:

UTFPR – Câmpus Ponta Grossa.
Professor Colaborador – 40h

2014 – 2015:

UTFPR – Câmpus Santa Helena.
Professor do Magistério Superior – Dedicação Exclusiva.

2015 – atual:

UTFPR – Câmpus Ponta Grossa.
Professor do Magistério Superior – Dedicação Exclusiva.

Áreas de Pesquisa



Grupo de Pesquisa Sistemas de Informação

Linha de Pesquisa Engenharia de Software

Pesquisas:

- Modelagem de Software;
- Linhas de Produto de Software;
- Framework de Domínio;
- Desenvolvimento Orientado a Modelos;
- Padrões de Projeto.

Projeto de Extensão



Manutenção Laboratório do Programa de Inclusão Digital

Coordenador: Prof. MSc. Vinícius Camargo Andrade

Este projeto tem como objetivo realizar a manutenção dos equipamentos utilizados pelos alunos, tais como: notebooks, desktops, roteadores, hubs, cabeamento, bem como instalação de softwares necessários para o andamento das disciplinas ofertadas pelo Programa de Inclusão Digital.

Sobre a Avaliação

$$Nota Final = \sqrt[10]{\left(\frac{1}{2} \sum_{i=1}^2 AT\right)^6 * (Arq^1) * (Impl^1) * (Testes^1) * (Manut^1)}$$

Onde:

- AT: Avaliação Teórica;
- Arq: Arquitetura de Software;
- Impl: Implementação do Sistema;
- Testes: Testes de Software;
- Manut: Manutenção de Software.

$$Nota Final Recuperada = Sub * 0,6 + Nota Final * 0,4$$

Onde:

- Sub: Avaliação Substitutiva;
- Nota Final: Nota obtida pelo aluno no decorrer do semestre.

Programa da Disciplina

- Fundamentos da Engenharia de Software.
- Processo de desenvolvimento de software.
- Modelos de processos.
- Arquitetura de software.
- Engenharia de requisitos.
- Planejamento e gestão de projetos.
- Verificação e validação.
- Qualidade de software.
- Testes de software.

Referências

- Básica

- PRESSMAN, Roger S. Engenharia de software: uma abordagem profissional. 7. ed. Porto Alegre, RS: AMGH, 2011. 780 p.
- PFLEEGER, Shari Lawrence. Engenharia de software: teoria e prática. 2. ed. São Paulo, SP: Prentice-Hall, 2004. 537 p.
- SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo, SP: Pearson Prentice Hall, 2011. 529 p.

Referências

- Complementar
 - BEHFOROOZ, Ali; HUDSON, Frederick J. Software engineering fundamentals. New York: Oxford University Press, 1996. 661 p.
 - CARVALHO, Ariadne Maria Brito Rizzoni; CHIOSSI, Thelma Cecília dos Santos. Introdução à engenharia de software. Campinas, SP: UNICAMP- Universidade Estadual de Campinas, 2001.148p.
 - MENDES, Sueli Bandeira Teixeira; AGUIAR, Teresa Cristina de. Métodos para especificação de sistemas. São Paulo: Edgard Blücher, 1989. 183 p.
 - ROCHA, Ana Regina Cavalcanti da; MALDONADO, José Carlos; WEBER, Kival Chaves. Qualidade de software: teoria e prática. São Paulo: Prentice-Hall, 2001. 303 p.
 - SCHACH, Stephen R. Engenharia de software: os paradigmas clássicos& orientado a objetos. 7. ed. São Paulo: McGraw-Hill, 2009. 618 p.

Engenharia de Software (definição 1)

- Segundo Martins (1995), Engenharia de Software é o desenvolvimento e aplicação de ciência, matemática, técnicas, métodos, metodologias e ferramentas para o desenvolvimento e a manutenção econômica de software de qualidade, previsível e controlável, operando de modo econômico em máquinas e ambientes reais.

Engenharia de Software (definição 2)

- Segundo Boehm (2002), a engenharia de software é a “aplicação prática do conhecimento científico ao projeto e construção de programas e da documentação necessária para o desenvolvimento, operação e manutenção dos mesmos.”

Engenharia de Software (definição 3)

- Engenharia de Software é um procedimento sistemático que é usado para atender a um conjunto de metas para análise, projeto, implementação e manutenção de software. O software resultante deve ser eficiente, confiável, usável, modificável, portátil, testável, reutilizável, fácil de manter, interoperável e correto. Estes termos referem-se tanto aos sistemas como aos seus componentes (THIRY, 2002).

Engenharia de Software

APRESENTAÇÃO | Engenharia de Software

A ENGENHARIA DE SOFTWARE (ES) lida com métodos, metodologias, ferramentas e disciplinas, que colaboram para o desenvolvimento de sistemas. Por isto, a ES acaba sendo uma ponte que integra ainda áreas da Computação, como Banco de Dados, Linguagens de Programação, Interface Homem Máquina, entre outras, e outras áreas que transcendem a Computação. As principais instituições de ensino e pesquisa no país têm departamentos e centros da área de Computação com fortes grupos da área de Engenharia de Software. Todos notamos que o software está cada vez mais presente em tudo que nos rodeia, daí a frase (meio provocativa, meio séria) que costumamos usar afirmando que “Tudo é Software”.

Engenharia de Software

- Engenheiros Civis fazem Plantas antes de construírem prédios;
- Engenheiros Eletrônicos fazem Esquemas antes de montarem aparelhos;
- Engenheiros Mecânicos fazem Desenhos antes de produzirem máquinas;
- Engenheiros de Software são superdotados pela Mãe Natureza, e não precisam de nada disso.

Comparação entre Áreas da Engenharia

- O ser humano faz casas e abrigos há milhões de anos;
- O ser humano lida com eletricidade há centenas de anos;
- O ser humano produz máquinas e ferramentas há outros milhares de anos;
- O ser humano faz software há 50/60 anos.

Aplicações da Engenharia de Software

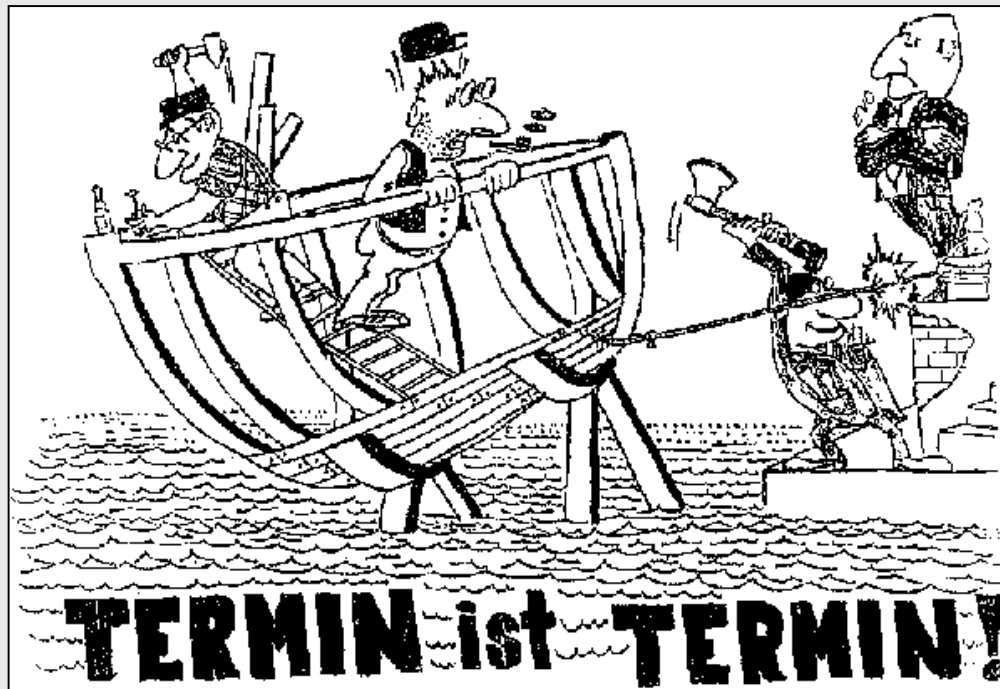
- Software de Sistema (Compilador)
- Software Científico/de engenharia (Cálculo Matemático)
- Software Embutido (Funções digitais)
- Software para linha de produtos (Múltiplos clientes)
- Aplicações para web (WebApps) Aplicações Comerciais

Aplicações da Engenharia de Software

- Software de inteligência artificial (reconhecimento de padrões – imagens ...)
- Netsourcing (recursos via internet) Uso pessoal que forneça benefício ao mercado mundial
- Software Aberto (SO, Frameworks...)

Realidades da Engenharia de Software

- Prazos Audaciosos



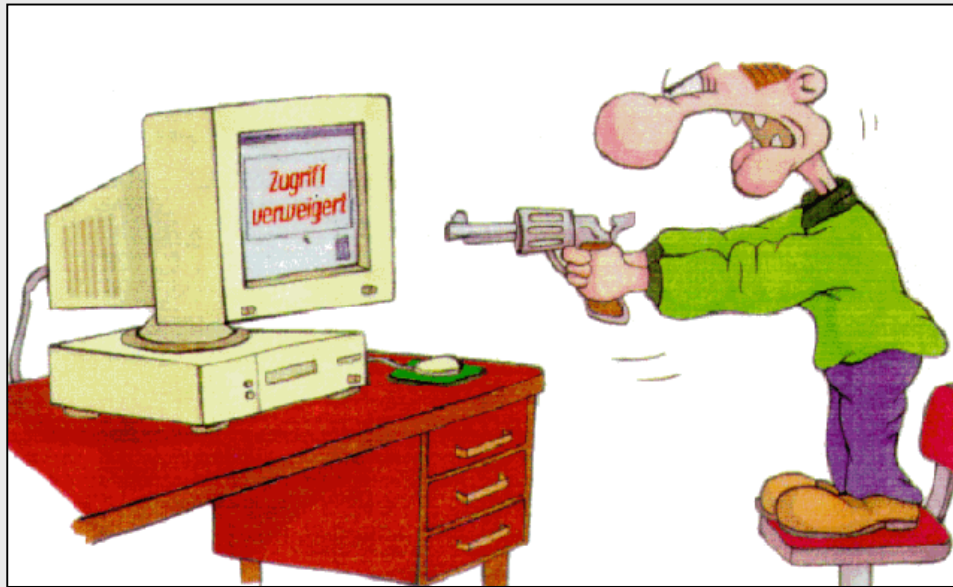
Realidades da Engenharia de Software

- Ouvir o mercado



Realidades da Engenharia de Software

- Usuários odeiam Bugs



Realidades da Engenharia de Software

- Experiência em Simuladores ajuda



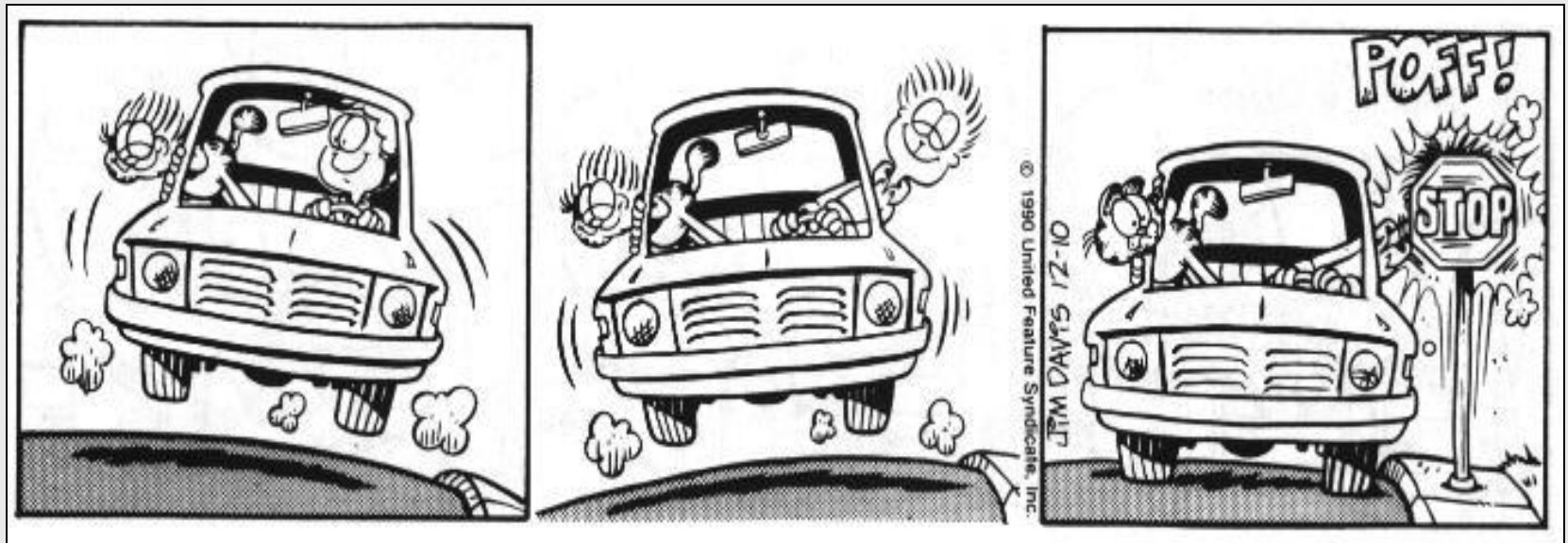
Realidades da Engenharia de Software

- Nem todo o Projeto será um Sucesso



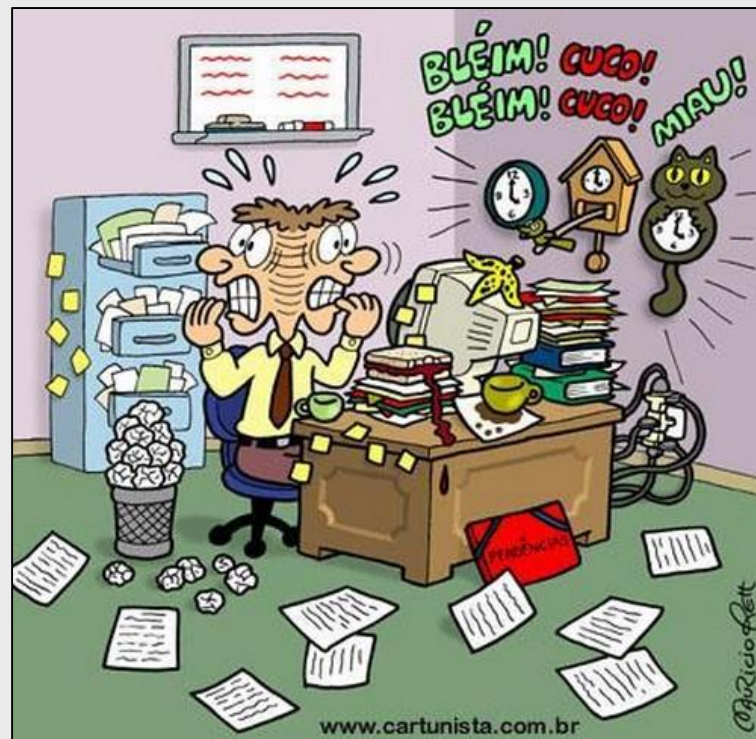
Realidades da Engenharia de Software

- O que serve para um, pode não servir para outros



Realidades da Engenharia de Software

- Buscar soluções eficientes para o problema



Realidades da Engenharia de Software

- Suporte: Ferramenta correta para o problema



Realidades da Engenharia de Software

- Prever e otimizar são tarefas complexas



Evolução do Software

- A Evolução do Software inicia em 1950 e atualmente é dividida em 5 eras:
 - Primeira Era (1950 a início de 1960):
 - Aplicações científicas e de Engenharia.
 - Segunda Era (1960 a meados de 1980):
 - Aplicações comerciais em grande-porte (sistemas de informações BD).
 - Terceira Era (meados de 1970 e década de 80):
 - Aplicativos pessoais em microcomputadores.

Evolução do Software

- Quarta Era (meados de 1980 a início de 1990):
 - Aplicativos com interfaces gráficas.
 - Redes e arquitetura Cliente-Servidor.
- Quinta Era (meados de 1990 até a atualidade):
 - Software distribuído, internet, groupwares e intranets.

Crise do Software

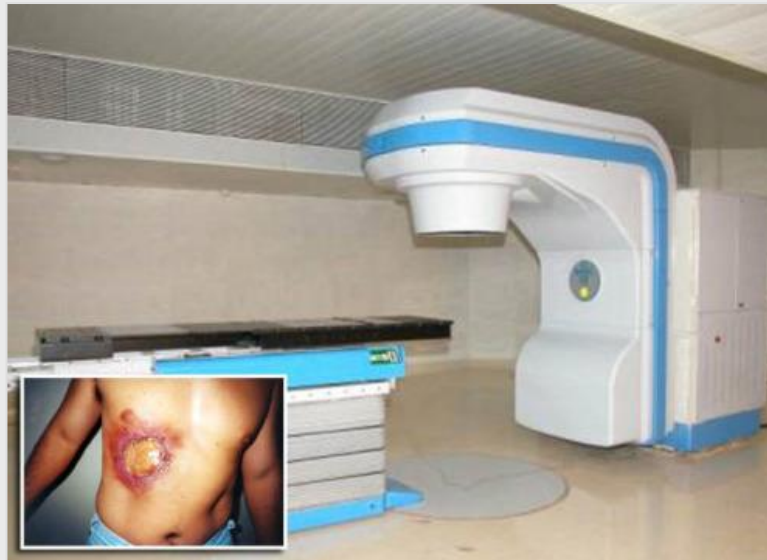
- Termo utilizado nos anos 70, porém emergiu no final dos anos 60;
- Engenharia de Software inexistente.
- O termo expressava:
 - Dificuldades do desenvolvimento de software;
 - Crescimento da demanda por software;
 - Complexidade dos problemas a serem resolvidos;
 - Inexistência de técnicas estabelecidas para o desenvolvimento.

Crise na Engenharia de Software

- Crise acompanha a Engenharia de Software há aproximadamente 40 anos.
 - Projetos estourando prazos;
 - Software de baixa qualidade;
 - Software muitas vezes não atingem os requisitos;
 - Projetos ingerenciáveis e difícil de manter.
- Problemas não estão apenas na funcionalidade dos sistemas, mas sim no desenvolvimento, testes, implantação e manutenção.

Crise na Engenharia de Software

- Therac-25.
 - Se envolveu em 6 acidentes, causando mortes por overdoses de radiação.
 - Software foi adaptado de uma antecessora, Therac-6:
 - Falhas por falta de testes integrados
 - Falta de documentação



Crise na Engenharia de Software

- Pepsi
 - A promoção era a seguinte : Quem tirasse a tampinha com o número 349 impresso, ganharia uma premiação em dinheiro.
 - Um problema no sistema das máquinas de impressão resultou na distribuição de 800 mil tampinhas com a numeração premiada nas Filipinas.
 - Na época, a empresa não entregou os prêmios, o que provocou bastante revolta.



1991

Crise na Engenharia de Software

- Denver International Airport
 - Custo do projeto: US\$ 4.9 bilhões
 - 100 mil passageiros por dia
 - 1,200 voos
 - 53 milhas quadradas
 - 94 portões de embarque e desembarque
 - 6 pistas de pouso / decolagem



Crise na Engenharia de Software

- Denver International Airport
 - Erros no sistema automático de transporte de bagagens :
 - Atraso na abertura do aeroporto com custo total estimado em US\$360 Milhões
 - 86 milhões para consertar o sistema



Crise na Engenharia de Software

- Honda
 - Por causa de um defeito de programação no sistema dos carros, a Honda teve que realizar um *recall* em mais de 2 milhões de automóveis.
 - O fato custou alguns milhões de dólares à empresa japonesa.
 - O airbag era ativado com muita força e pelo componente errado.



Crise na Engenharia de Software

- Ariane 5
 - Projeto da Agência Espacial Européia que custou:
 - 10 anos.
 - US\$ 8 Bilhões.
 - Capacidade 6 toneladas.
 - Garante supremacia européia no espaço.



Crise na Engenharia de Software

- Ariane 5
 - o veículo detonou suas cargas explosivas de autodestruição e explodiu no ar porque estava se quebrando devido às forças aerodinâmicas.
 - O foguete tinha perdido o controle de direção (atitude) devido os computadores principal e backup se desligarem ao mesmo tempo.
 - O desligamento ocorreu devido a um *run time error* (out of range, overflow, ou outro) e ambos computadores se desligaram. O erro se originou de um programa que convertia um valor em ponto flutuante para um inteiro de 16 bits que recebeu como entrada um valor que estava fora da faixa permitida.

Crise na Engenharia de Software

- Ariane 5
 - Resultado:
 - Explosão 40 segundos após a decolagem.
 - Destruição do foguete e carga avaliada em US\$ 500 milhões.



Problemas na Engenharia de Software

1. As estimativas de prazo e de custo frequentemente são imprecisas
 - Não dedicamos tempo para coletar dados sobre o processo de desenvolvimento de software;
 - Sem nenhuma indicação sólida de produtividade, não podemos avaliar com precisão a eficácia de novas ferramentas, métodos ou padrões;
 - Análise de ponto de função
 - Resulta em definições “no olho”.



Problemas na Engenharia de Software

2. Insatisfação do cliente com o sistema concluído

- Os projetos de desenvolvimento de software normalmente são efetuados apenas com um vago indício das exigências do cliente
 - Comunicação fraca com o cliente



Problemas na Engenharia de Software

3. A qualidade de software às vezes é menos que adequada
- Só recentemente começam a surgir conceitos quantitativos sólidos de garantia de qualidade de software
 - Ex: Microsoft – avaliação de níveis de erros (versões: Alfa, Beta, Final ou Comercial)



Problemas na Engenharia de Software

4. O software existente é muito difícil de manter
- A tarefa de manutenção devora o orçamento destinado ao software
 - A facilidade de manutenção não foi enfatizada como um critério importante



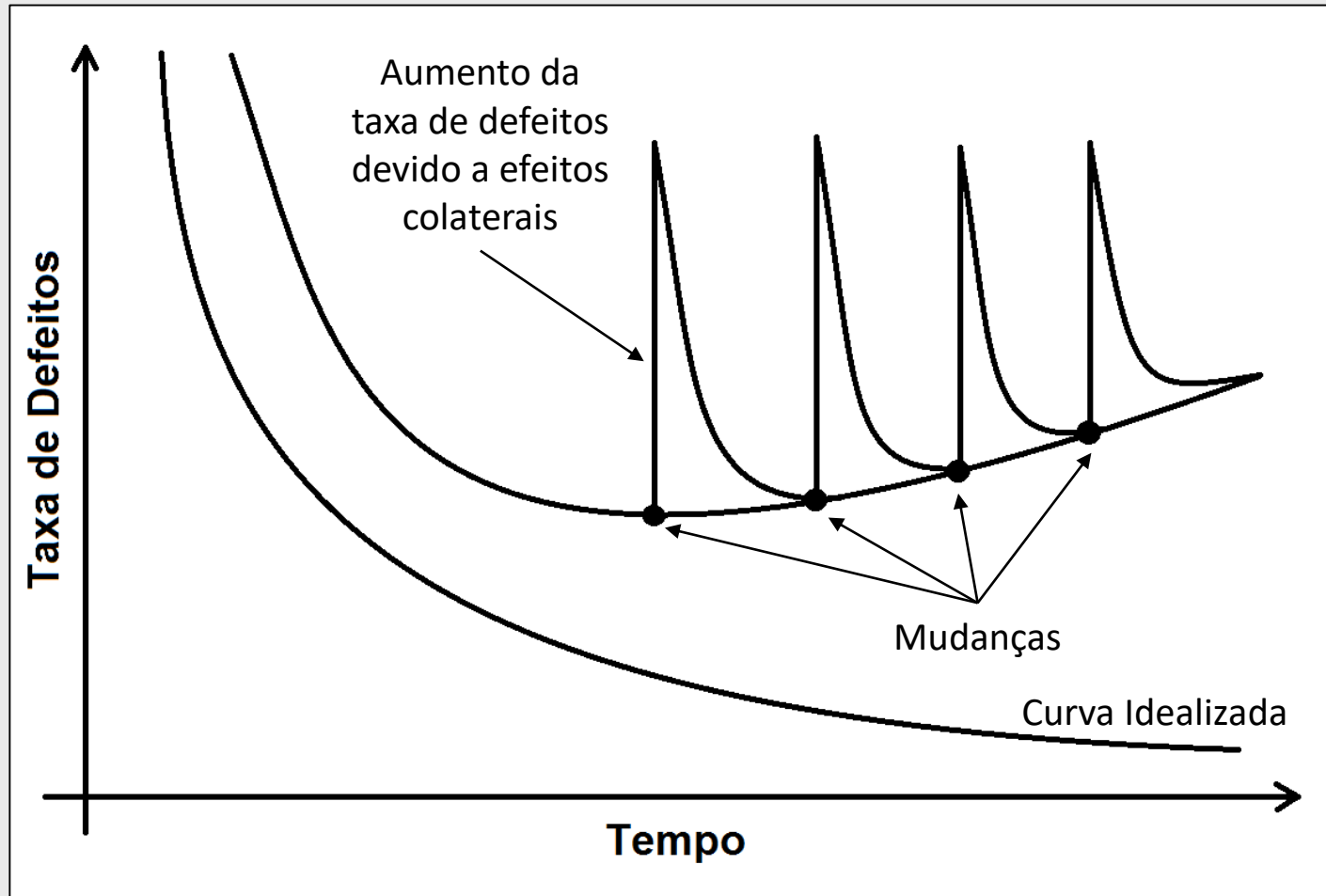
Causas dos Problemas

1. Próprio caráter do software

- O software é um elemento de sistema lógico e não físico. Consequentemente o sucesso é medido pela qualidade de uma única entidade e não pela qualidade de muitas entidades manufaturadas.

O software não se desgasta, mas se deteriora.

Curva de Desgaste do Software



Causas dos Problemas

2. Falhas das pessoas responsáveis pelo desenvolvimento de software
 - Gerentes sem nenhuma experiência em software;
 - Profissionais da área de software têm pouco treinamento formal em novas técnicas para o desenvolvimento de software;
 - Resistência a mudanças.

Causas dos Problemas

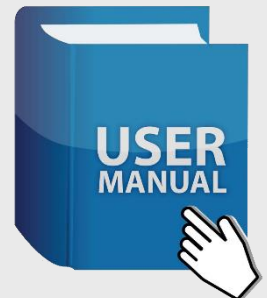
3. Mitos do software

- Propagaram desinformação e confusão
 - Administrativos
 - Cliente
 - Profissional

Mitos do Software - Administrativos

- Mito 1

Já temos um manual repleto de padrões e procedimentos para a construção de software.
Isso não oferecerá ao meu pessoal tudo o que eles precisam saber?
- Realidade
 - Será que o manual é usado?
 - Os profissionais sabem que ele existe?
 - Ele reflete a prática moderna de desenvolvimento de software?
 - Ele é completo?



Mitos do Software - Administrativos

- Mito 2
Computadores de última geração solucionam problemas de desenvolvimento.
- Realidade
É preciso muito mais do que os mais recentes computadores para se fazer um desenvolvimento de software de alta qualidade.
Ex: ferramentas CASE



Mitos do Software - Administrativos

- Mito 3:

Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e tirar o atraso.

- Realidade:

O desenvolvimento de software não é um processo mecânico igual à manufatura. Acrescentar pessoas em um projeto torna-o ainda mais atrasado.

Pessoas podem ser acrescentadas, mas somente de uma forma planejada.



Mitos do Software - Cliente

- Mito 1

Uma declaração geral dos objetivos é suficiente para se começar a escrever programas

- Podemos preencher os detalhes mais tarde.

- Realidade

Uma definição inicial ruim é a principal causa de fracassos dos esforços de desenvolvimento de software.

É fundamental uma descrição formal e detalhada do domínio da informação, função, desempenho, interfaces, restrições de projeto e critérios de validação.

Definição dos requisitos funcionais



Mitos do Software - Cliente

- Mito 2

Os requisitos de projeto modificam-se continuamente e as mudanças podem ser facilmente acomodadas visto que o software é flexível.

- Realidade

Uma mudança, quando solicitada tardiamente num projeto, pode ser maior do que a ordem de magnitude mais dispendiosa da mesma mudança solicitada nas fases iniciais.



Mitos do Software - Cliente

- Magnitude das mudanças

| Fases | Custo de Manutenção |
|-----------------|---------------------|
| Definição | 1x |
| Desenvolvimento | 1,5 – 6x |
| Manutenção | 60 – 100x |

Mitos do Software - Profissional

- Mito 1

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

- Realidade

Os dados da indústria indicam que entre 50 e 70% de todo esforço gasto num programa serão despendidos depois que ele for entregue pela primeira vez ao cliente.

“quanto mais cedo se começa a escrever o código, mais tempo se demora para que se consiga terminá-lo”



Mitos do Software - Profissional

- Mito 2

Enquanto não tiver o programa "funcionando", eu não terei realmente nenhuma maneira de avaliar sua qualidade.

- Realidade

Um programa funcionando é somente uma parte de uma Configuração de Software que inclui todos os itens de informação produzidos durante a construção e manutenção do software.



Configuração de Software

