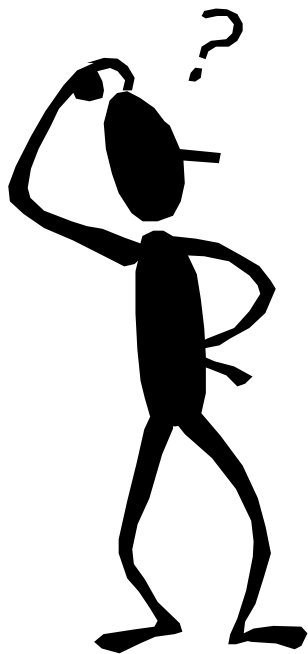


Engenharia de Software

Marcelle Mussalli Cordeiro
{mmussalli@gmail.com}

Reflexão



O que é software??

O que é Software?

- Programa
- Dados de configuração
- Dados de documentação
- *Tudo que esteja envolvido com a operação correta do programa.*

Categorias para aplicações de Software

- Software de Sistemas (Software Básico)
 - Programas escritos para servir a outros programas. Ex: Compiladores, Sistemas Operacionais.
- Software de Aplicação
 - Programas que resolvem uma necessidade específica de um negócio. Ex: Aplicações comerciais, tempo real, controle de processos, ponto de venda, sistemas de gestão.
- Software para Linhas de Produto
 - Dirigido a clientes. Ex: Office, Entretenimento, Multimídia
- Software Científico e de Engenharia
 - Cálculos Estruturais, Naves espaciais

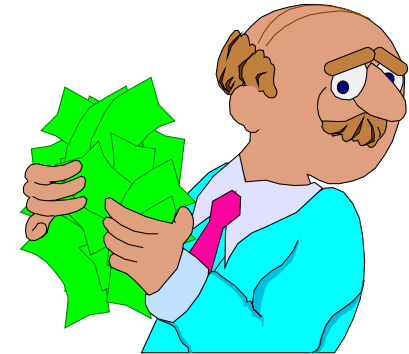
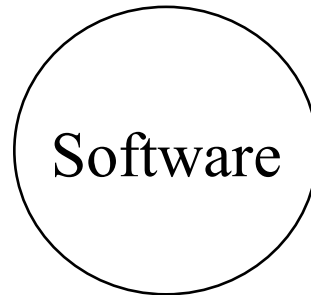
Categorias para aplicações de Software

- Software Embutido
 - Produtos inteligentes: microondas, automóveis.
- Software de Inteligência Artificial
 - Algoritmos não-numéricos, sistemas especialistas, redes neurais.
- Aplicações da WEB
 - B2B, Aplicações de comércio eletrônico, Páginas hipertexto.

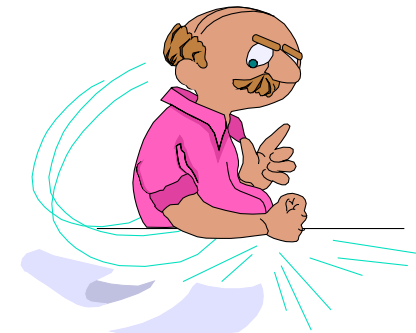
Desenvolvimento de Software



Desenvolvedor



Usuário

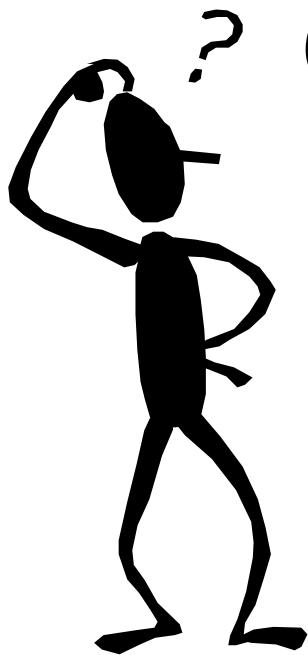


Material de Ricardo Storino

A evolução do Software

- Software cada vez
 - maior
 - mais funcionalidades
 - mais complexo
 - abrangendo mais usuários
 - exigindo a participação de equipes em seu desenvolvimento

Reflexão



O que é Engenharia de Software???

Engenharia de Software

- Causas
 - Evolução significativa de hardware e software.
 - Necessidade de criação de sistemas cada vez maiores e mais complexos.
 - Surgiu em 1968 como uma solução para a “**crise de software**”.

Crise de Software

- Introdução de novo hardware de computador baseado em circuitos integrados.
 - Viabilização de propostas de aplicações até então impossíveis.
 - O software era ordens de grandeza maior e mais complexo.
- A experiência inicial na construção destes sistemas.
- O desenvolvimento informal de software não era suficiente.
 - Grandes atrasos em projetos.
 - Alto custo de software, inclusive manutenção.
 - Desempenho insatisfatório.
- Necessidade de novas técnicas e métodos para controlar a complexidade.

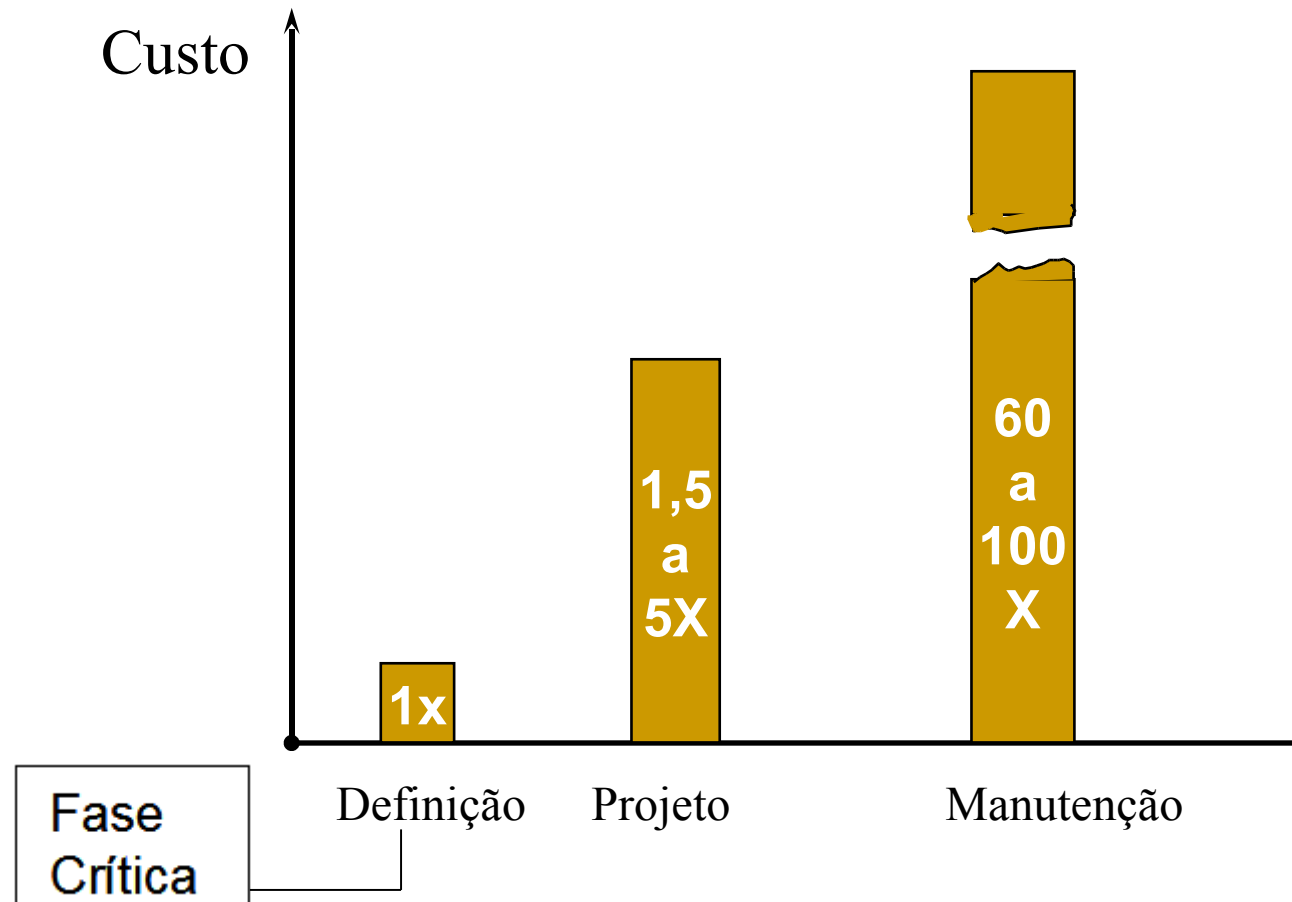
A Crise de Software

- Introdução de erros no processo
 - Má especificação
 - Mau projeto
 - Má implementação
 - Testes incompletos ou mal feitos
- Problemas na gerência
 - Falta de treinamento em novas técnicas de desenvolvimento
 - O processo está evoluindo muito rapidamente em função do aprendizado. É necessário reciclar.

A Crise de Software

- Dificuldades no Trabalho com Software
 - Medidas pobres de eficiência e qualidade
 - Insatisfação do usuário é frequente
 - Pouco entendimento dos requisitos
 - Problemas de Comunicação entre o usuário e o analista
 - A qualidade do software é frequentemente suspeita
 - Poucas medidas e critérios de qualidade
 - Software existente é muito difícil de manter
 - E tem que ser mantido até ser substituído

Custo de Mudanças



Material de Ricardo Storino

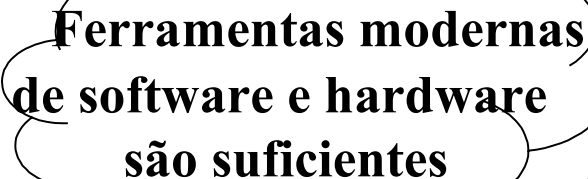
Mitos...

- Da Gerência...

Desatualizados, obsoletos

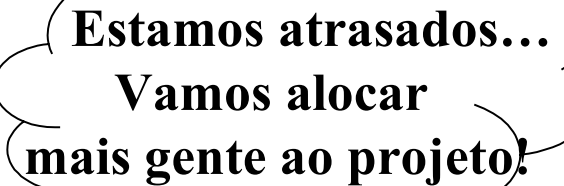


**Manuais de Regras
e Procedimentos**



**Ferramentas modernas
de software e hardware
são suficientes**

O uso eficiente de ferramental
exige conhecimento



**Estamos atrasados...
Vamos alocar
mais gente ao projeto!**

Custos de treinamento,
gerência e
entendimento do
processo de trabalho

Material de Ricardo Storino

Mitos...

- Do Desenvolvedor / Programador...

Especificação, projeto,
plano de trabalho

**O único produto de um
projeto de software é o
conjunto de programas**

**Até que o programa
esteja “rodando” não
há como medir
sua qualidade**

**Programa escrito
e testado!
Acabei!**

Revisões anteriores à
codificação

Quanto mais cedo você escrever o
código, mais tempo irá demorar
para completá-lo.

De 50 a 70 % do custo de produção
de um software vai ser gasto para
operacionalizá-lo para o usuário



Mitos...

- Do Cliente...

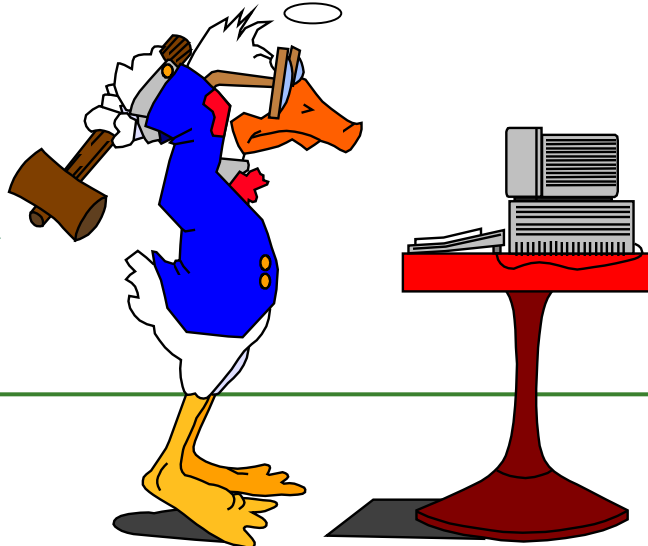
Uma lista de intenções (boas) é suficiente para começar a produzir o software

A Especificação do Software é a fase mais crítica do processo

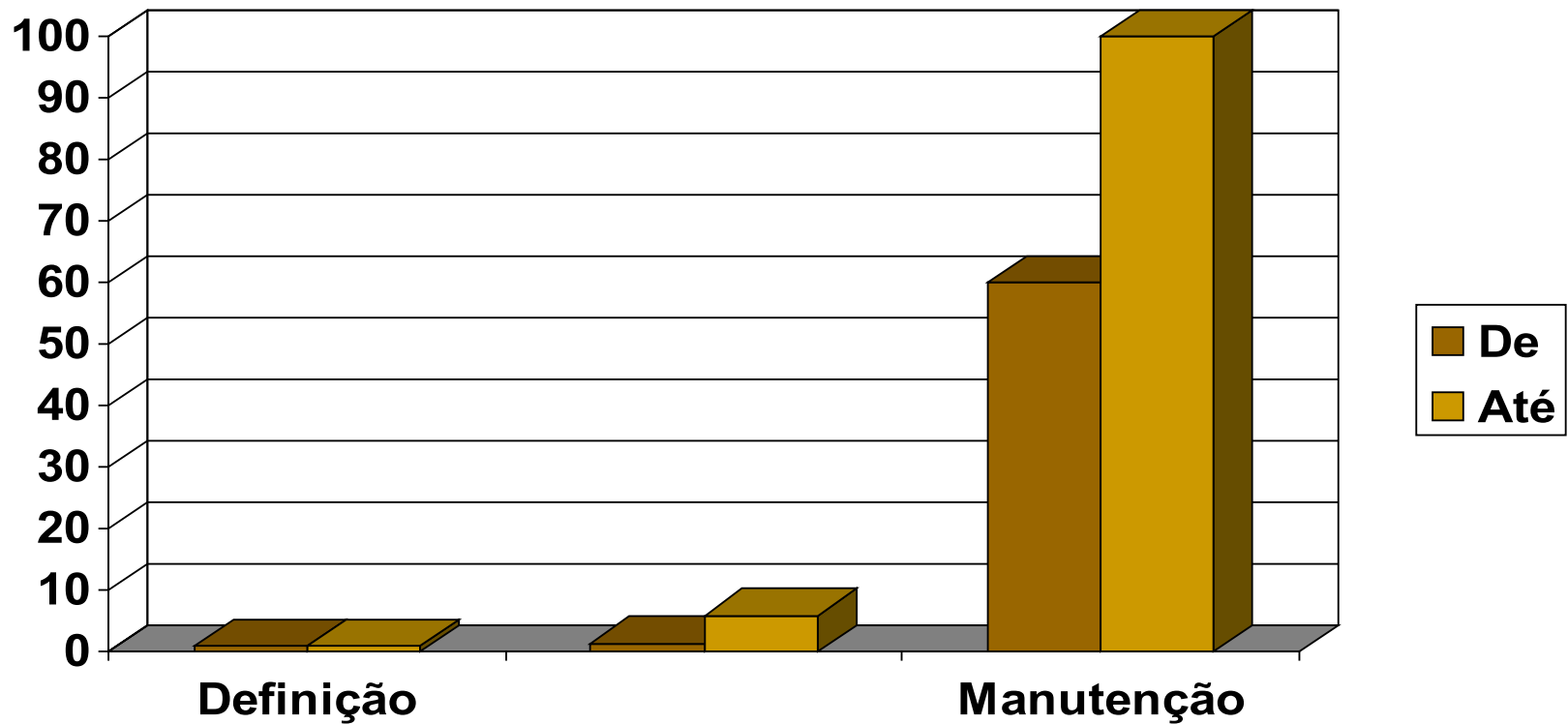
Erros na fase inicial têm um custo muito alto de correção

Minhas necessidades vão mudar... Mas mudanças são fáceis de introduzir porque software é bastante flexível

Custo de mudanças é muito alto



Custo de Introdução de Mudanças



Material de Ricardo Storino

Engenharia de Software

- *Produzir e manter o software dentro de custos adequados é essencial para o funcionamento da economia nacional e internacional.*
- Ramo da engenharia cujo foco é o desenvolvimento dentro de **custos adequados** de sistemas de software de **alta qualidade**.
- Entendimento das atividades envolvidas no desenvolvimento.
- Desenvolvimento de métodos eficazes de especificação, projeto e implementação de software.
- Novas notações e ferramentas reduzem o esforço necessário para produzir sistemas complexos e de grande porte.

Engenharia de Software

- Avanços mais importantes na Engenharia de Software:
 - Aparecimento da UML como padrão para a descrição de sistemas orientados a objetos.
 - Desenvolvimento de métodos ágeis, como o *Extreme Programming*.
 - Desenvolvimento rápido de sistemas.
 - Envolvimento dos usuários na equipe.
 - Redução de documentos e burocracia no processo de software.

Engenharia Civil X Engenharia de Software

- Porque pontes são geralmente construídas a tempo e a um custo conhecido... e softwares raramente os são?
 - 3000 anos de experiência a mais
 - Engenharia Civil: Clara separação entre especificação e execução
 - Especificação detalhada antes de “colocar a mão na massa”
 - Uma vez que a especificação é aceita, pouco se pode fazer para alterar
 - Terrenos não estão constantemente mudando
 - Engenharia de Software: a especificação faz parte da execução
 - Códigos fontes são também especificações
 - Tendência a clientes solicitar mudanças constantemente
 - Em negócios, mudança é uma necessidade
- Conclusão
 - Projetos de software possuem maiores riscos e devem ser gerenciados com atenção voltada às mudanças

Engenharia de Software em camadas

- FERRAMENTAS
 - Apoio automatizado para o processo e para os métodos
- MÉTODOS (Notações)
 - Técnica de como fazer para construir software. Conjunto de tarefas que incluem comunicação, análise de requisitos, modelagem de projeto, construção de programas, testes e manutenção.
- PROCESSO (Procedimentos)
 - Alicerce da ES. Arcabouço para utilização da tecnologia. Permite o desenvolvimento racional de softwares. Base de controle gerencial de projetos de software. Onde os métodos são aplicados. Roteiro
- FOCO NA QUALIDADE
 - Base da ES. Gestão de qualidade total, cultura de processo contínuo de qualidade.

Material de Ricardo Storino

Processos

- Quais os passos para construir o software?
- Seqüência dos métodos
 - seqüência de passos e atividades
- Resultados
 - documentos, relatórios, módulos
- Controles
 - coordenação, critérios de qualidade, pontos de controle
- Equipe

Processos

- Atividades
 - Especificação de Software.
 - Desenvolvimento de Software
 - Projeto e Implementação.
 - Validação do Software
 - Verificação de acordo com os requisitos do cliente.
 - Evolução do Software
 - Modificações para se adaptar às mudanças dos requisitos do cliente e do mercado.

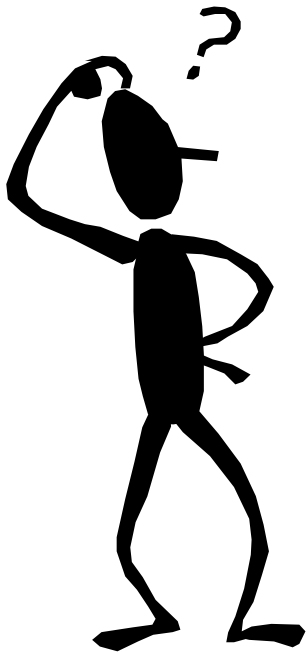
Métodos

- Como construir o software?
- Abordagem estruturada para desenvolvimento de software.
- Base:
 - Modelos de desenvolvimento de sistema, que pode ser representado graficamente.
 - Uso dos modelos como especificação e no projeto de sistema.
- Análise Estruturada (De Marco, 78)
- JSD (Jackson, 83)
- UML (Booch, Rumbaugh, Jacobson, 97)

Ferramentas Case

- *Computer-Aided Software Engineering*
- É preciso muito software para desenvolver software!
- Programas usados para dar apoio às atividades do processo de software.
- Associada ao método, executa uma verificação com relação a notações e conceitos específicos do método.
- São editores de diagramas.
- Podem conter um gerador de código.

Reflexão



O que é um bom software???

O que é um bom software?

- **Depende!**
 - Sistema bancário – Segurança.
 - Jogo interativo – Resposta ágil.
 - Sistema de comutação telefônica – Confiável.
- **Atributos essenciais**
 - Facilidade de manutenção – Mudança de requisitos.
 - Confiança

O que é um bom software?

- Atributos essenciais
 - Facilidade de manutenção
 - Mudança de requisitos.
 - Confiança
 - Confiabilidade, Proteção e Segurança.
 - O software não deve causar danos físicos ou econômicos em caso de falha.
 - Eficiência
 - Não desperdiçar recursos do sistema, como memória e processamento.
 - Usabilidade
 - Interface com o usuário e documentação adequadas.

Engenharia de Software

Marcelle Mussalli Cordeiro
{mmussalli@gmail.com}