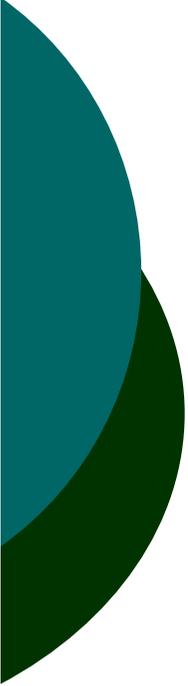


Engenharia de Software Introdução

Ricardo Argenton Ramos 
ricargentonramos@gmail.com

Engenharia de Software I – 2012.2



Tópicos

- Apresentação da Disciplina
- A importância do Software
- Software
- Aplicações de Software
- Paradigmas de Projetos
- Mitos de Software
- Processo de Software
- Modelos de Processo de Software

Ementa da Disciplina

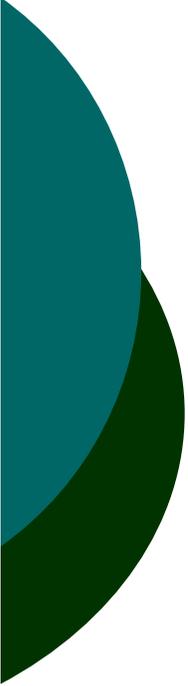
○ EMENTA:

- Conceitos gerais de Engenharia de Software;
- Modelos de Processo:
 - Cascata, Prototipação, Espiral, Desenvolvimento Ágil e outros.
- Modelar um projeto seguindo as etapas de um modelo de processo;
- Engenharia de Requisitos (Plano de Negócio, Planejamento estratégico, Requisitos básicos do Software, Prototipação; Estudar os principais paradigmas de projeto (Fluxo de Dados, Estruturado, Tempo Real, Baseado em Funções e OO);
- Etapa de Projeto: Desenvolver a etapa de projeto preliminar e detalhado utilizando uma ferramenta Case.
- Estudar as principais etapas de Apoio para o desenvolvimento de Software com Qualidade (Revisão, Validação e Teste).

○ REFERÊNCIAS:

- PRESSMAN, Roger. Engenharia de Software. Rio de Janeiro: McGraw Hill, 2006.
- SOMERVILLE, Ian. Engenharia de Software. Addison Wesley Editora, 2007.



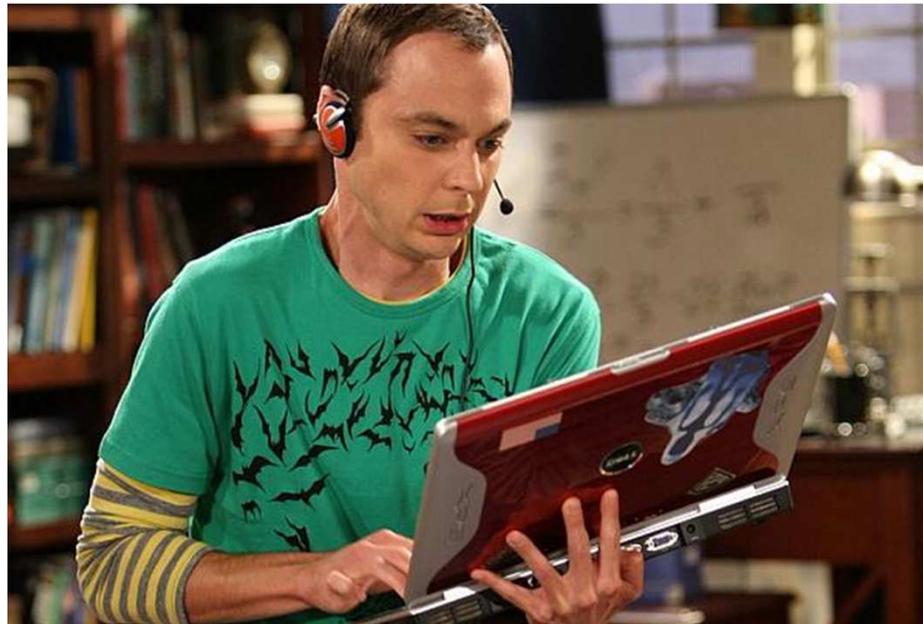


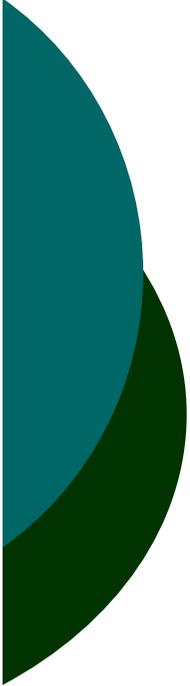
Avaliações da Disciplina

- A nota final será composta pela média aritmética das seguintes avaliações:
 - AV1 - 1ª avaliação (05/02/2013 às 08:00);
 - AV2 - 2ª avaliação (26/03/2013 às 08:00);
 - Média das Avaliações dos exercícios:
 - T1 - 20/01/2013 (Moodle);
 - T2 - 16/02/2013 (Moodle);
 - T3 - 17/03/2013 (Moodle);
 - T4 - (12/03/2013) Entrega e apresentação do Trabalho de Junit;
- Fórmula para calcular a Nota Final:

$$\mathbf{NF = (AV1 + AV2 + (T1+T2+T3+T4)/4)/3}$$

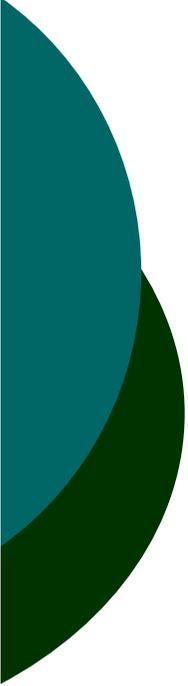
A importância do Software





A Importância do SOFTWARE

- **INSTRUÇÕES:** que quando executadas produzem a função com desempenho desejados.
- **ESTRUTURAS DE DADOS:** que possibilitam que os programas manipulem a informação de maneira adequada e segura.
- **DOCUMENTOS :** descrição das operações e o do uso dos programas.



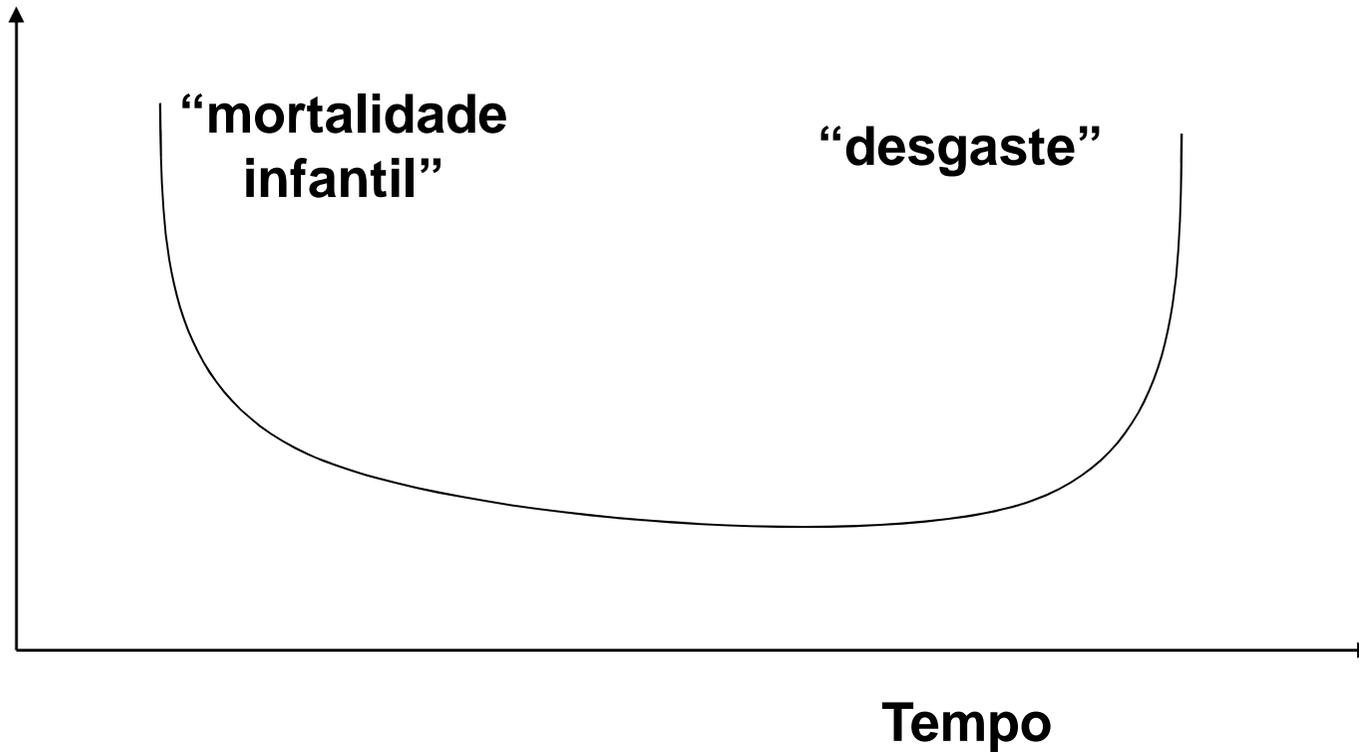
Características do software

- Não se desgasta com o tempo, mas pode se deteriorar;
- Pode ser desenvolvido ou projetado pela engenharia, não manufaturado no sentido clássico;
- Principal destaque na melhoria da performance de uma Empresa.

Hardware

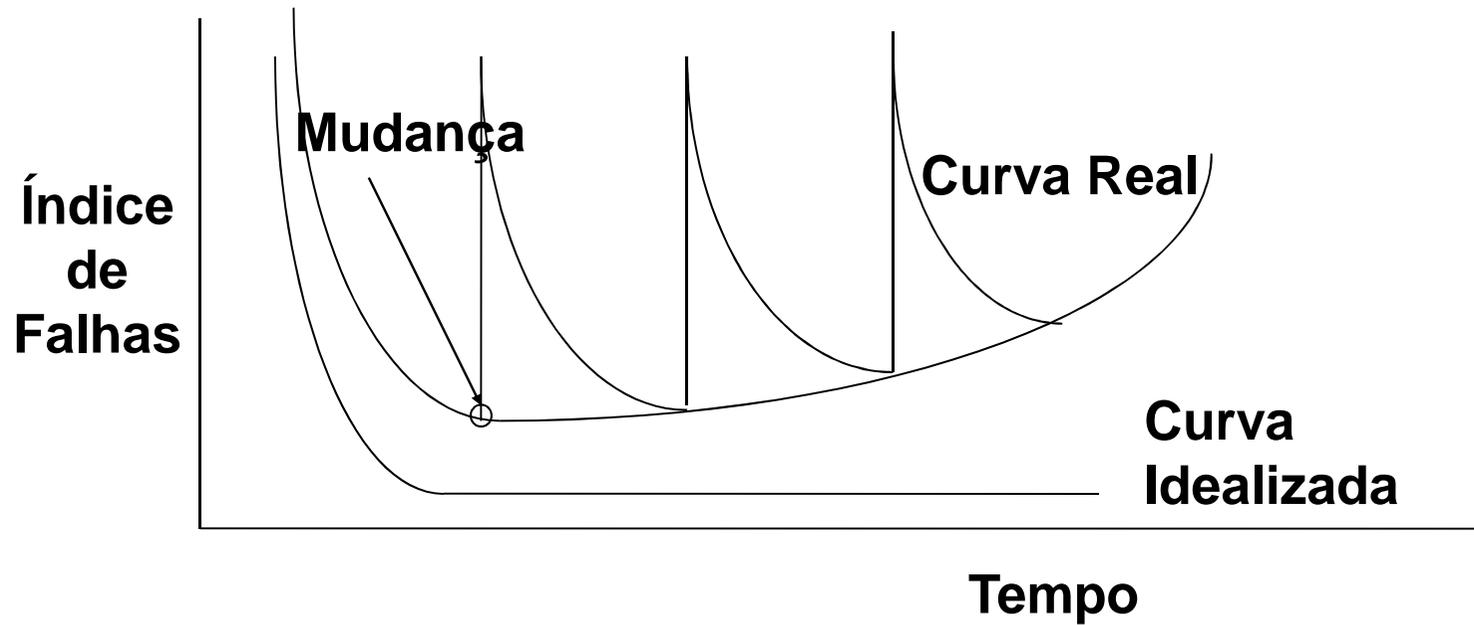
Curva de desgaste

Índice de Falhas



Software

Curva de Desgaste



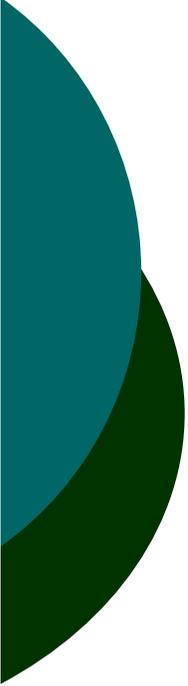
Características - Software

- A maioria é feita sob medida em vez de ser montada a partir de componentes existentes;



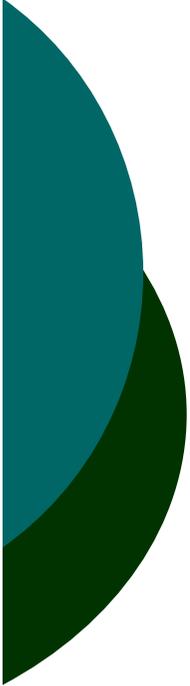
- Grande parte dos softwares produzidos no passado não possuem documentação e nem planejamento.





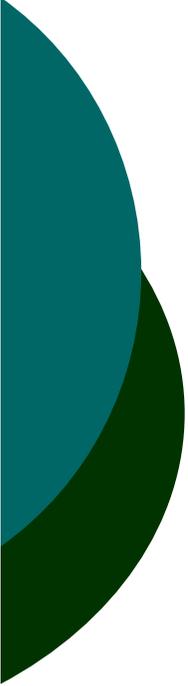
Aplicações dos Softwares

- **BÁSICO** coleção de programas escritos para dar apoio a outros programas;
- **TEMPO REAL** software que monitora, analisa e controla eventos do mundo real;
- **CIENTÍFICO E DE ENGENHARIA** caracterizado por algoritmos de processamento de números



Aplicações dos Softwares

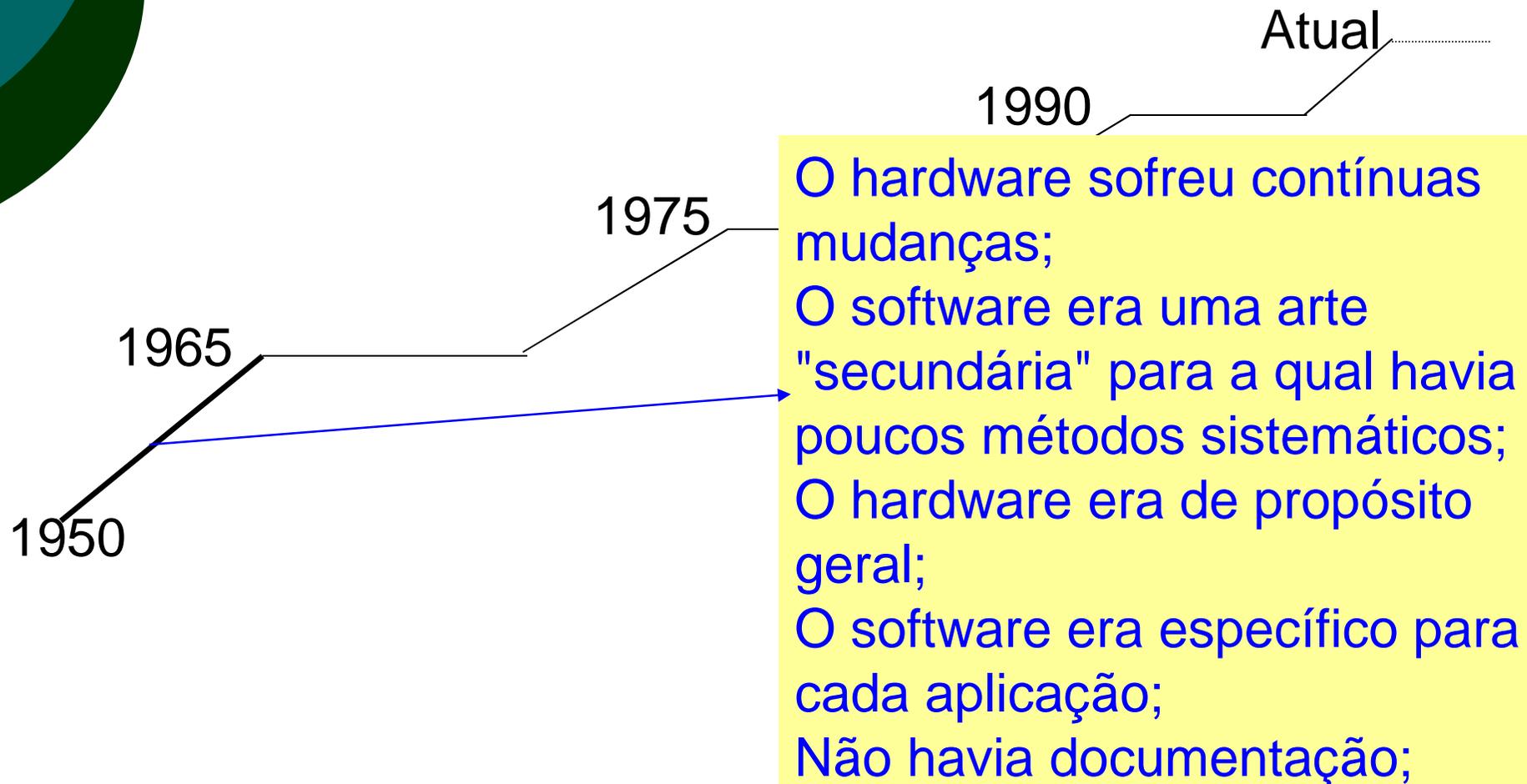
- **COMERCIAL ou EMPRESARIAL**
sistemas de operações comerciais e tomadas de decisões administrativas;
- **EMBUTIDO ou EMBARCADO**
usado para controlar produtos e sistemas para os mercados industriais e de consumo
- **DE COMPUTADOR PESSOAL**
envolve processamento de textos, planilhas eletrônicas, diversões, etc.



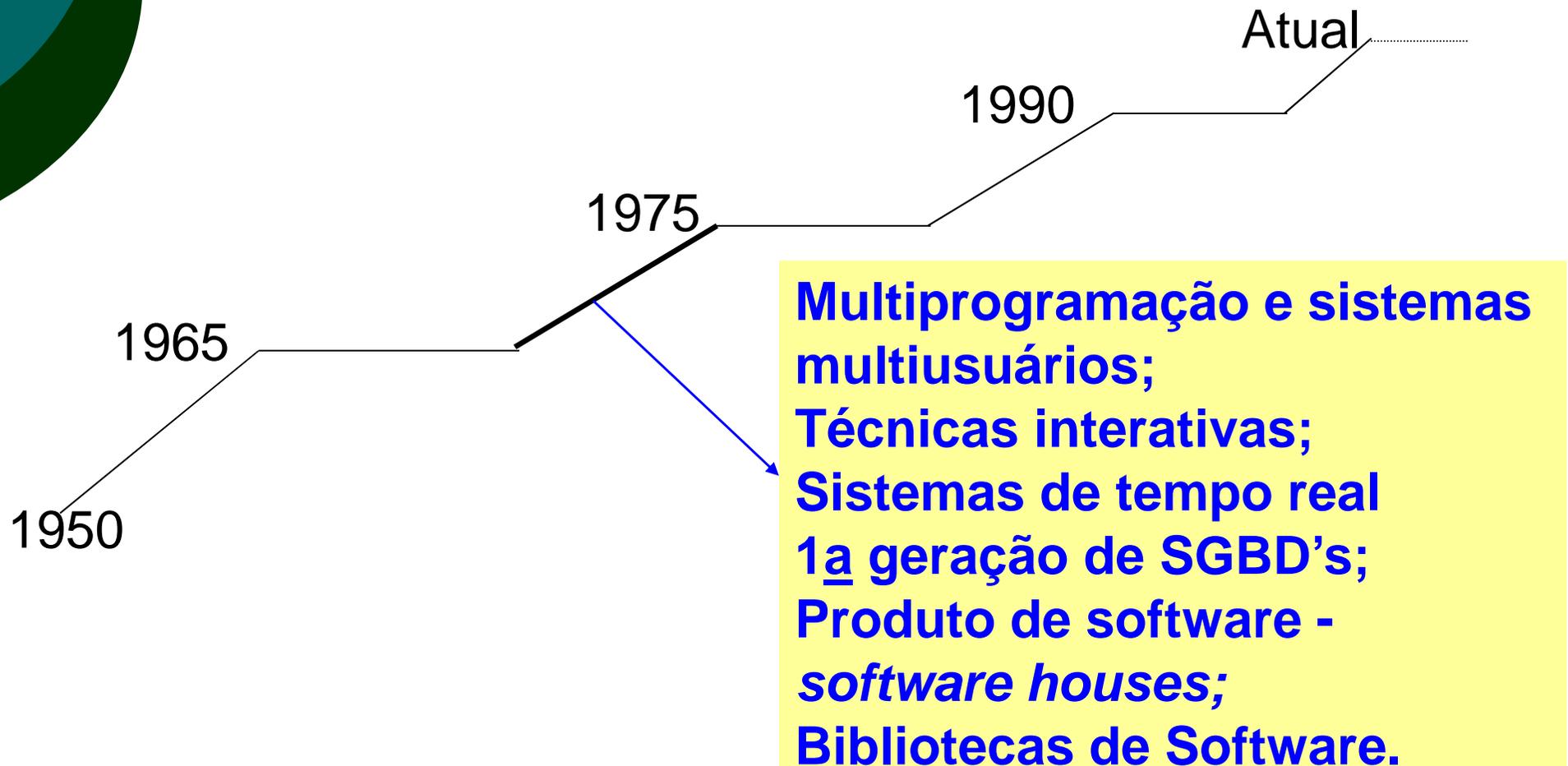
Aplicações dos Softwares

- **DE INTELIGÊNCIA ARTIFICIAL**
faz uso de algoritmos não numéricos para resolver problemas que não sejam favoráveis à computação ou à análise direta;

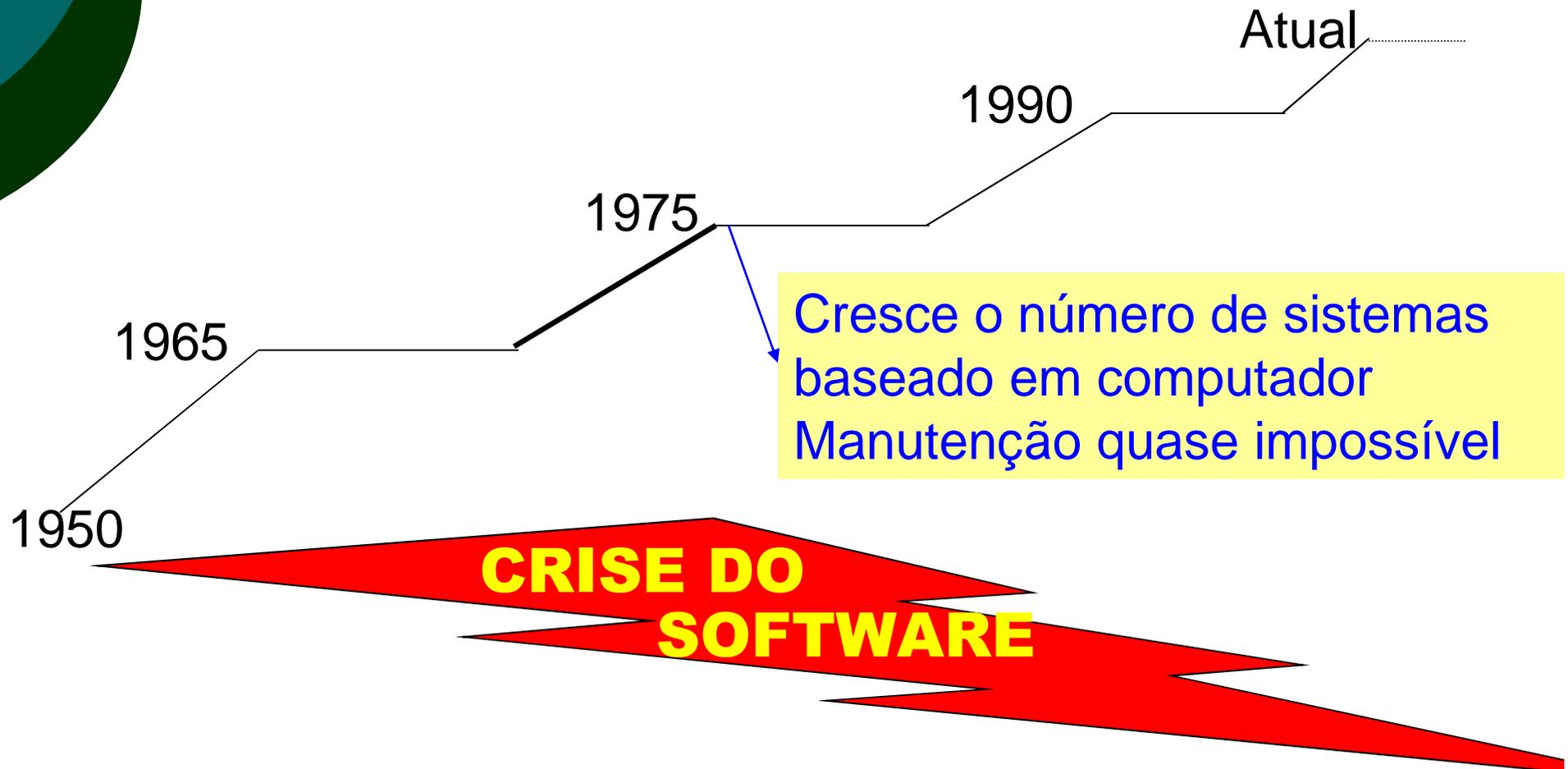
A Evolução do Software



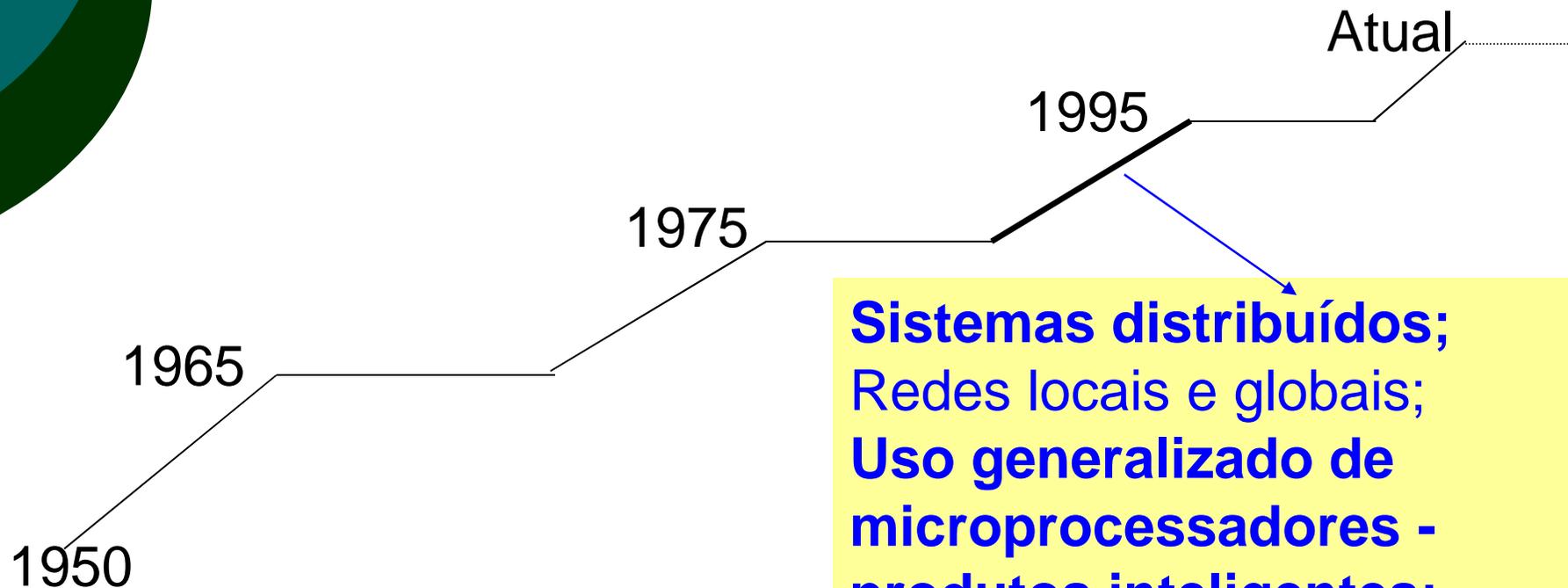
A Evolução do Software



A Evolução do Software



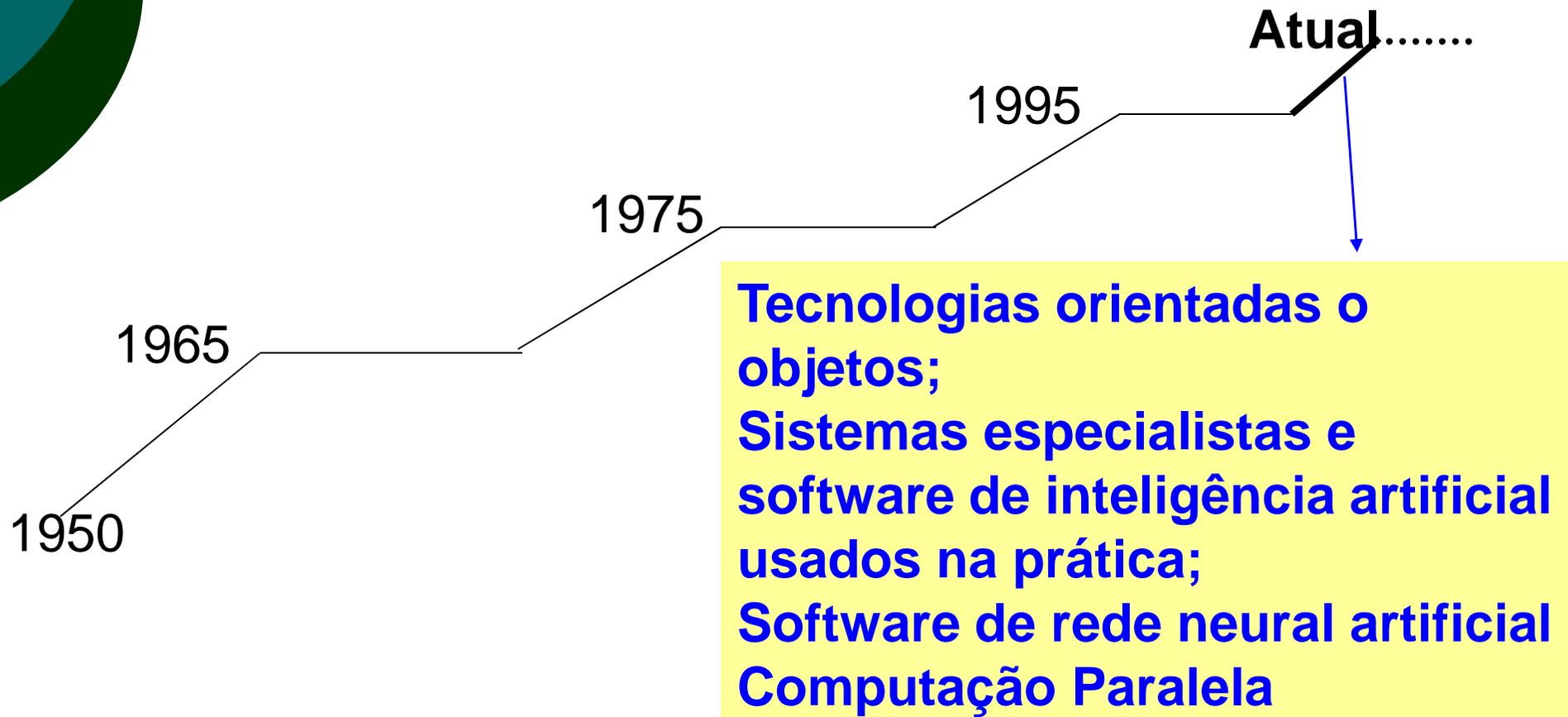
A Evolução do Software



Sistemas distribuídos;
Redes locais e globais;
Uso generalizado de
microprocessadores -
produtos inteligentes;
Hardware de baixo custo;
Impacto de consumo.

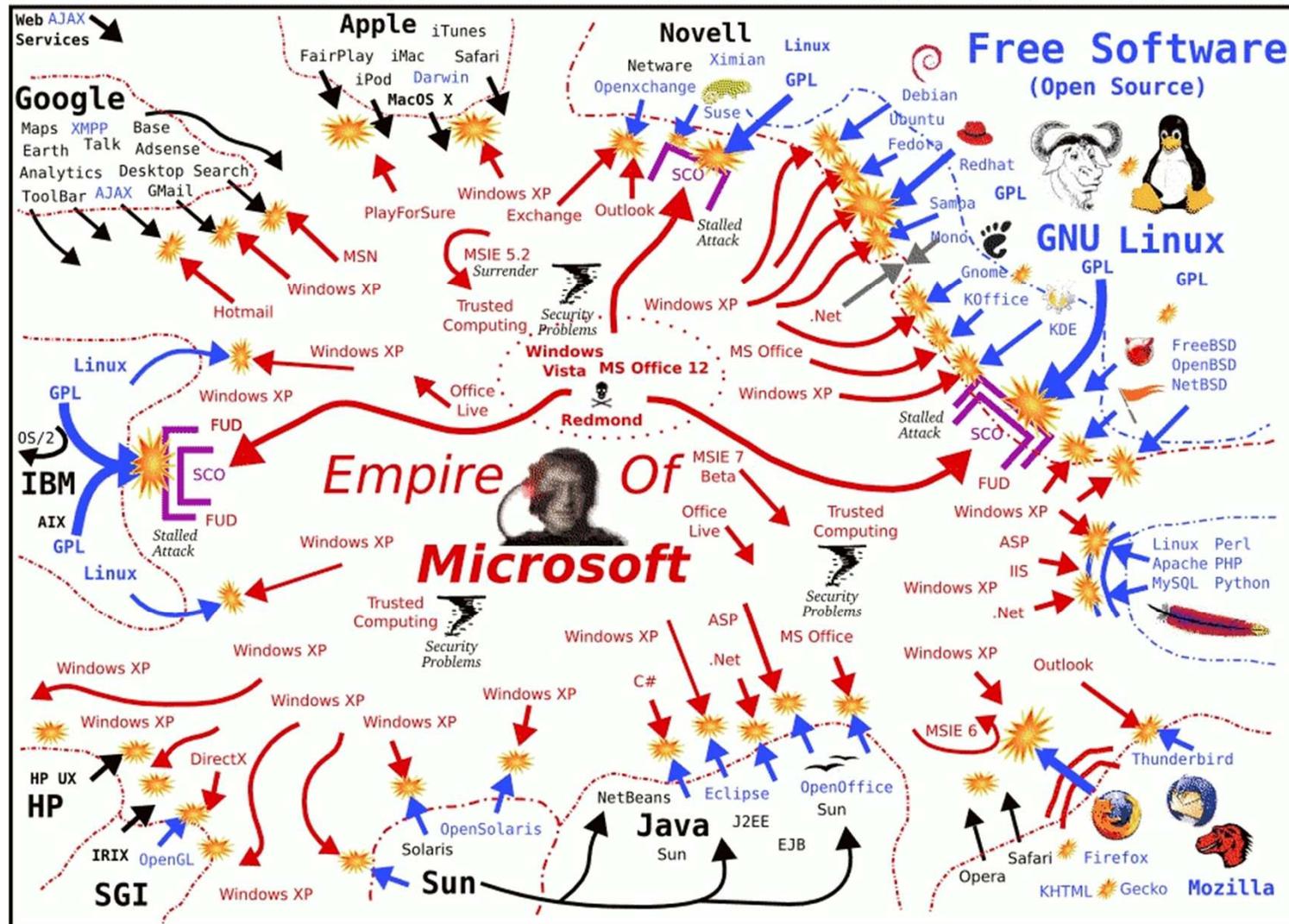
A Evolução do Software

(Quarta era da evolução)



A Guerra dos Softwares !

Software Wars http://mshiltonj.com/software_wars/ **Update: 2006-01-29** All trademarks are property of thier respective owners.
Copyright 2006 - Steven Hilton <mshiltonj@gmail.com> Permission to copy is granted if copyright notice is preserved.

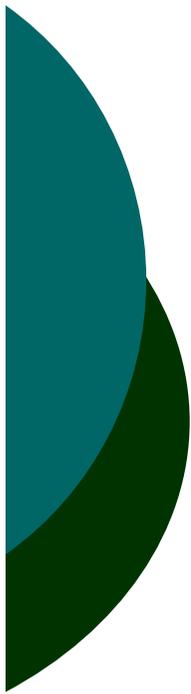


Crise do Software

Refere-se a um conjunto de problemas encontrados no desenvolvimento de software e na etapa de Manutenção.



© Scott Adams, Inc./Dist. by UFS, Inc.



Como o cliente explicou



Como o lider de projeto entendeu



Como o analista planejou



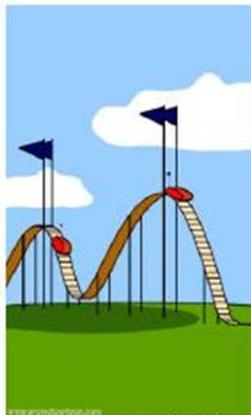
Como o programador codificou



O que os beta testers receberam



Como o consultor de negocios descreveu



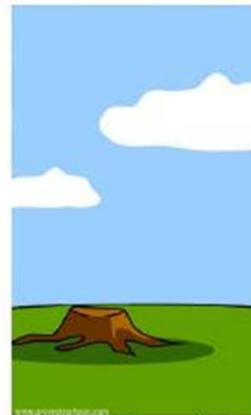
Valor que o cliente pagou



Como o projeto foi documentado



O que a assistencia tecnica instalou



Como foi suportado



Quando foi entregue



O que o cliente realmente necessitava

Crise de Software

Problemas

1- As estimativas de prazo e de custo freqüentemente são imprecisas



- **“ Falta de dedicação na etapa de coletas de dados no processo de desenvolvimento de software”**
- **“Vaga indicação de produtividade não se pode avaliar com precisão a eficácia de novas ferramentas, métodos ou padrões”**

Crise de Software Problemas



2- Insatisfação do cliente com o sistema concluído



- **“Projetos de desenvolvimento de software são efetuados com um vago indício das exigências do cliente”**



Crise de Software

Problemas

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**



Crise de Software Problemas

4- O software existente é muito difícil de manter (Sem Mantenabilidade)

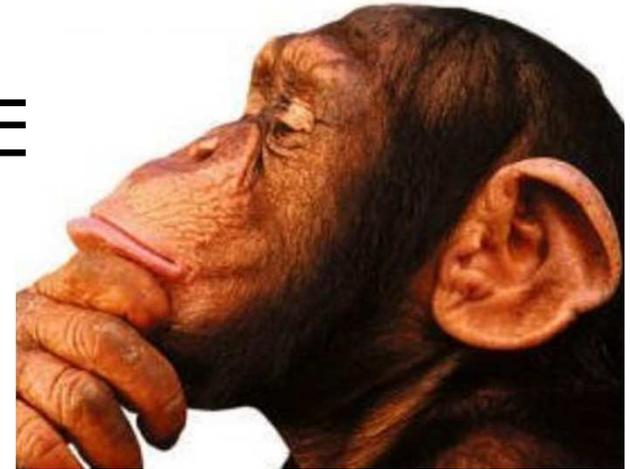


- **A tarefa de manutenção devora o orçamento destinado ao software**
- **A facilidade de manutenção não foi enfatizada como um critério importante**

Problemas associados a Crise do Software

1- CARACTERÍSTICAS PRÓPRIAS DO SOFTWARE

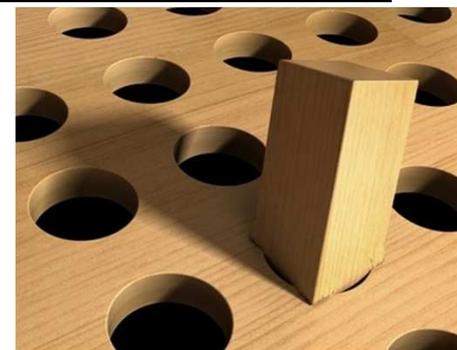
O software é um elemento de sistema lógico e não físico. Conseqüentemente o sucesso é medido pela qualidade de uma única entidade e não pela qualidade de muitas entidades manufaturadas



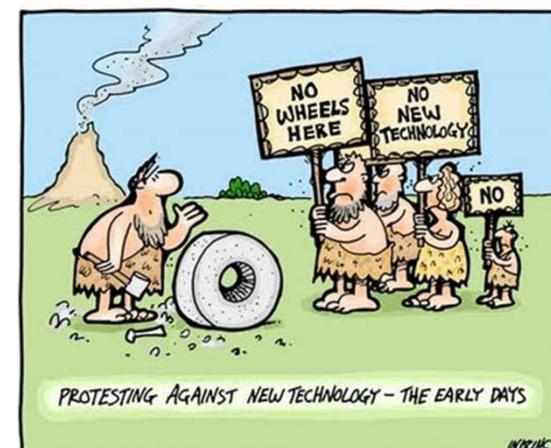
O software não se desgasta, mas se deteriora com o tempo.

Problemas associados a Crise do Software

2- FALHAS DAS PESSOAS RESPONSÁVEIS PELO DESENVOLVIMENTO DE SOFTWARE



- ❑ Gerentes sem nenhum background em software
- ❑ Profissionais da área de software têm pouco treinamento formal em novas técnicas para o desenvolvimento de software
- ❑ Resistência a mudanças



Problemas associados a Crise do Software

3- MITOS DO SOFTWARE

Falta de Planejamento e Comunicação

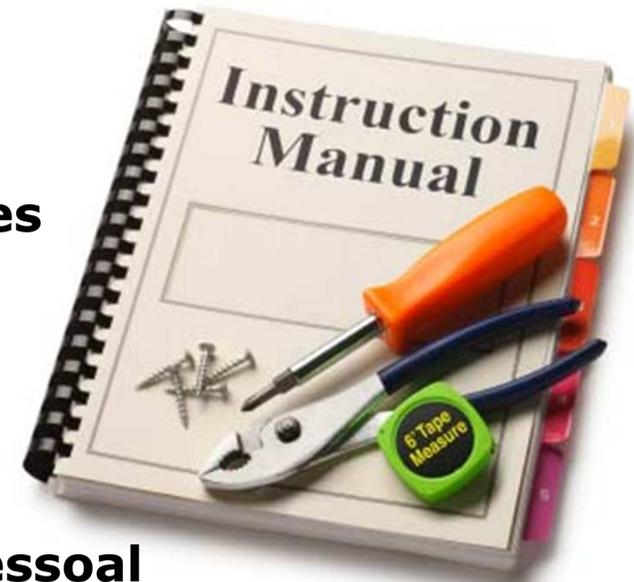
- ❑ ***administrativos***
- ❑ ***cliente***
- ❑ ***profissional***



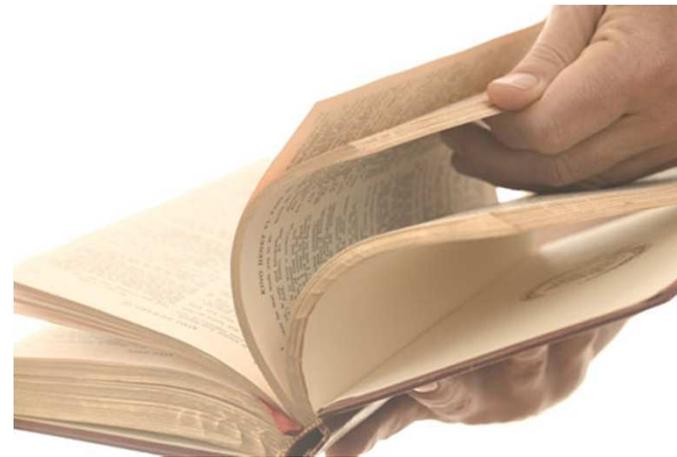
Mitos do software

○ Administrativo – Mito 1

Já temos um manual repleto de padrões e procedimentos para a construção de software.



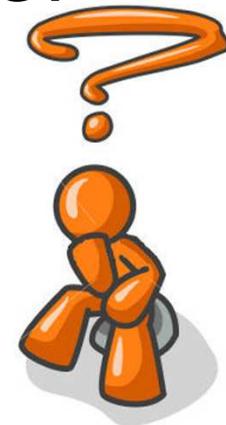
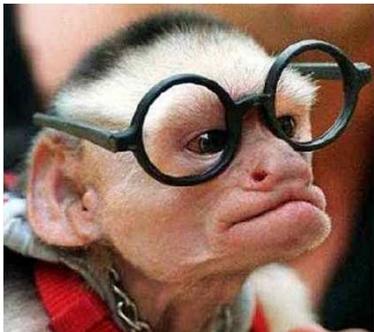
Isso não oferecerá ao meu pessoal tudo o que eles precisam saber?



Mitos do software

Realidade - Mito 1:

- *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

○ Administrativo *Mito 2* :

- Meu pessoal tem Ferramentas de desenvolvimento de software de última geração.



Realidade:

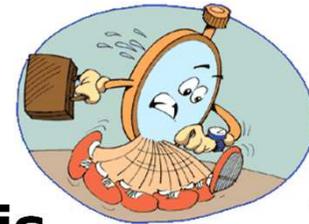
É preciso muito mais do que os mais recentes computadores e as melhores ferramentas para se fazer um desenvolvimento de software de alta qualidade.



Mitos do software

○ Administrativo Mito 3:

- Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e recuperar o atraso



ATENÇÃO

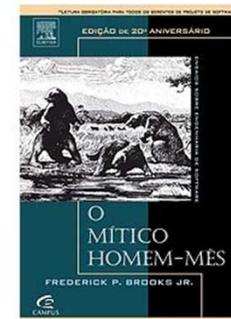


OBRAS EM ATRASO

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 começar a escrever programas - podemos preencher os detalhes mais tarde.



Mitos do software



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.

Os requisitos de usuário incluem 400 características

Você percebe que nenhum humano seria capaz de usar um produto com esse nível de complexidade

DILBERT by Scott Adams



Bom ponto. É melhor eu adicionar "fácil de usar" à lista.

Mitos do software

CLIENTE: Mito 2:

- Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o software é flexível.



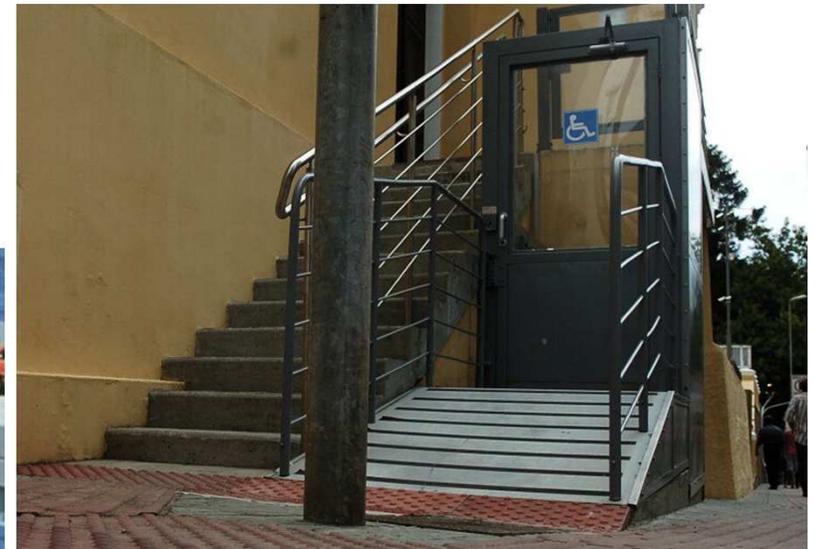
Mitos do software

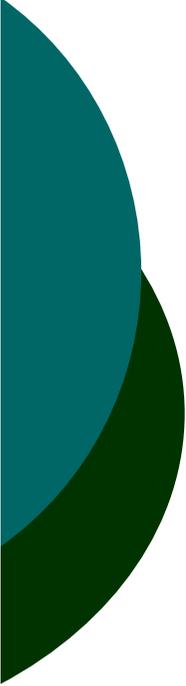
CLIENTE:



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

MAGNITUDE DAS MUDANÇAS

FASES	CUSTO DE MANUTENÇÃO
D EFINIÇÃO	1 x
D ESENVOLVIMENTO	1.5 - 6x
M ANUTENÇÃO	60 - 100x

Mitos do software

PROFISSIONAL: Mito 1:

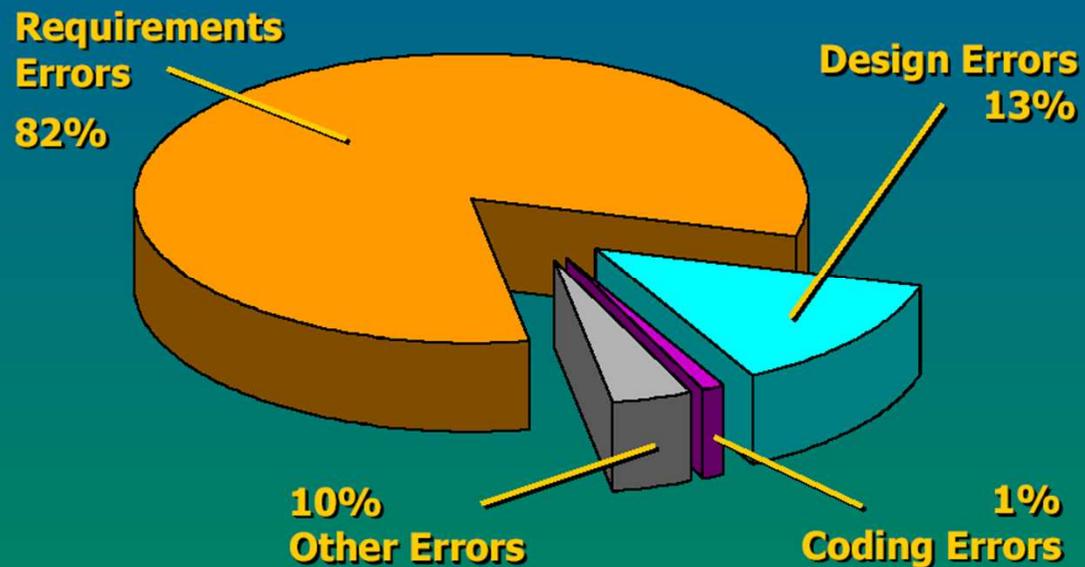
- Assim que escrevermos o programa e o colocarmos em funcionamento nosso trabalho estará completo.



Mitos do software

PROFISSIONAL: *Realidade:*

The Cost to Fix Software Defects



Mitos do software

PROFISSIONAL: Mito 2 :

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

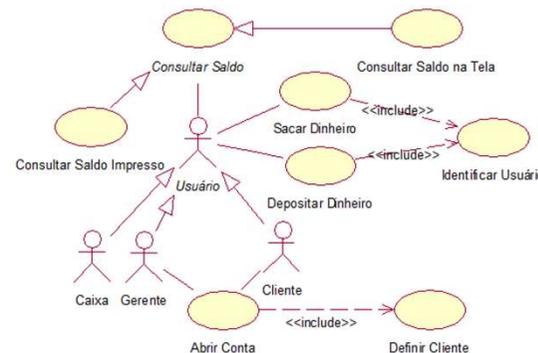
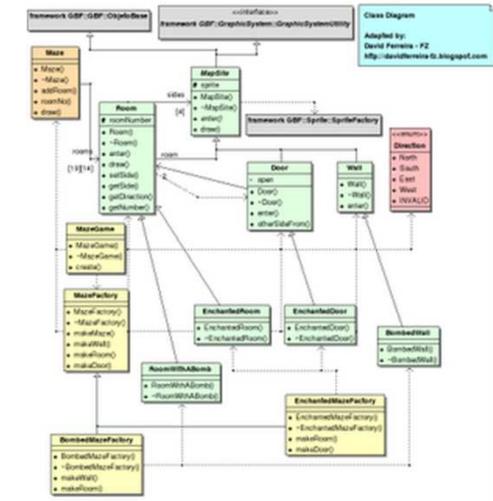


Mitos do software

PROFISSIONAL:

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.*



Evolução do Software

CRISE DE SOFTWARE

Refere-se a um conjunto de problemas encontrados no desenvolvimento de software



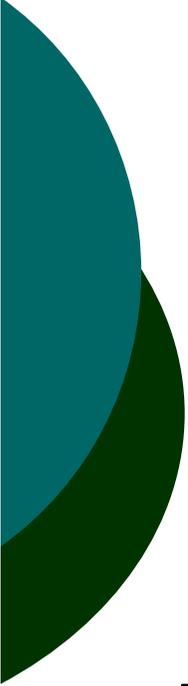
Resposta à Crise de Software

Engenharia de Software



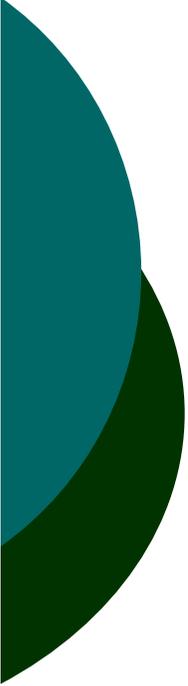
PROCESSO DE SOFTWARE

A aplicação de uma **abordagem** sistemática, disciplinada e possível de ser medida para o desenvolvimento, operação e manutenção do software (*IEEE*).



O Processo de Software

- Abrange um conjunto de três elementos fundamentais: **Métodos, Ferramentas e Procedimentos** para projetar, construir e manter grandes sistemas de software de forma profissional



O Processo de Software

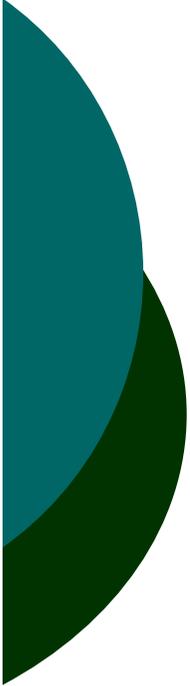
- **MÉTODOS:** fornecem os detalhes sobre como fazer para construir o software
 - **Planejamento e estimativa de projeto**
 - **Análise de requisitos de software e de sistemas**
 - **Projeto da estrutura de dados**
 - **Algoritmos de processamento**
 - **Codificação**
 - **Teste**
 - **Manutenção**

O Processo de Software

- **FERRAMENTAS: dão suporte automatizado aos métodos.**



- **Existem atualmente ferramentas para sustentar cada método**
- **Quando as ferramentas são integradas, é estabelecido um sistema de suporte ao desenvolvimento de software chamado *CASE - Computer Aided Software Engineering***



O Processo de Software

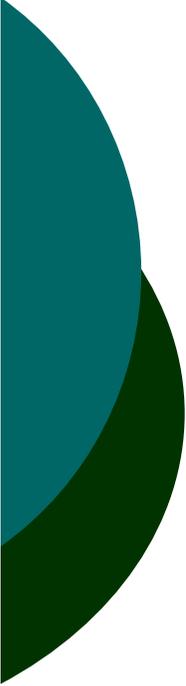
- PROCEDIMENTOS: constituem o elo de ligação entre os métodos e as ferramentas
 - Seqüência em que os métodos serão aplicados
 - Produtos que se exige que sejam entregues
 - Controles que ajudam a assegurar a qualidade e coordenar as alterações
 - Marcos de referência que possibilitam administrar o progresso do software.

Um Processo de Software com Qualidade

- **A Qualidade do Processo de Software** está relacionada à extensão na qual um processo de software específico é *eficiente* e é explicitamente *definido, gerenciado, medido e controlado*.
- **A Qualidade de Processo de Software** também implica em um *potencial* para crescimento na capacidade do processo de software e a consistência com a qual ele é aplicado em projetos por toda a organização.

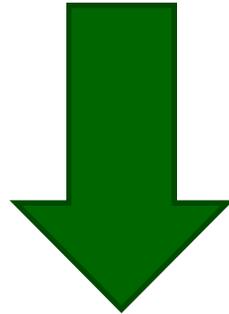


**Teremos uma aula sobre
Padrões de Qualidade**

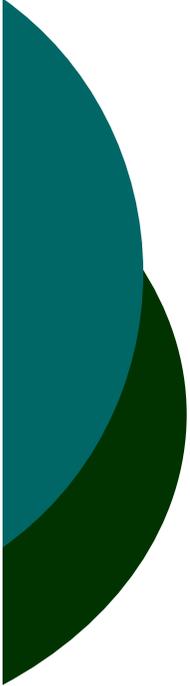


Um Processo de Software com Qualidade *(SOMMERVILLE)*

Que pode ser facilmente compreendido



- **Inteligibilidade**
 - o processo é definido e inteligível
- **Visibilidade**
 - o progresso do processo é visível externamente
- **Suportabilidade**
 - o processo pode ser apoiado por ferramentas CASE



Um Processo de Software com Qualidade *(SOMMERVILLE)*

- Aceitabilidade
 - o processo é aceito por todos envolvidos nele
- Confiabilidade
 - os erros do processo são descobertos antes que resultem em erros no produto
- Robustez
 - o processo pode continuar a despeito de problemas inesperados

Um Processo de Software com Qualidade (SOMMERVILLE)

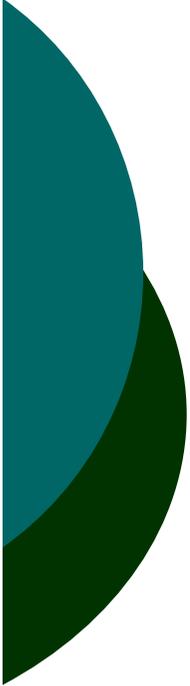
- **Manutenibilidade**

- o processo pode evoluir para atender alterações de necessidades organizacionais

- **Velocidade**

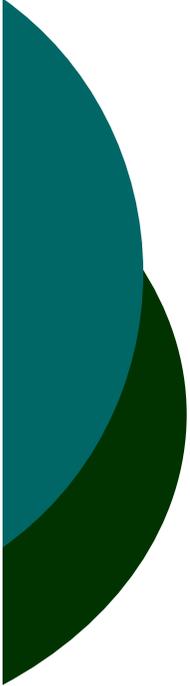
- quanto rápido o sistema pode ser produzido





Fases Genéricas dos Modelos de Processo de ENGENHARIA

- **Especificação** - estabelecer os requisitos e restrições do sistema
- **Projeto** - produzir um modelo documentado do sistema
- **Implementação** - construir o sistema
- **Teste** - verificar se o sistema atende às especificações requeridas
- **Instalação (ou implantação)** - liberar o sistema para o cliente e garantir que ele se torne operacional
- **Manutenção** - eliminar defeitos e evoluir o sistema conforme demanda.



Fases Genéricas dos Modelos de Processo de SOFTWARE

- Independentemente da natureza do projeto e aplicação os modelos de processo de software possuem:
 - **fase de definição**
 - **fase de desenvolvimento**
 - **fase de manutenção**
 - **atividades de apoio**

Fase de Definição do Processo de Software

focaliza "o que" será desenvolvido



- que informação vai ser processada
- que função e desempenho são desejados
- que comportamento pode ser esperado do sistema
- que interfaces vão ser estabelecidas
- que restrições de projeto existem
- que critérios de validação são exigidos para definir um sistema bem sucedido
- que tarefas serão realizadas

Fase de Desenvolvimento do Processo de Software

Focaliza "como" o software será desenvolvido



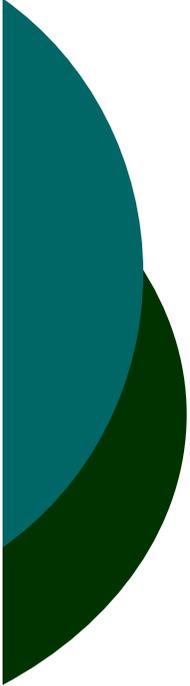
- como os dados vão ser estruturados
- como a função vai ser implementada em uma arquitetura de software
- como os detalhes procedimentais vão ser implementados
- como as interfaces vão ser caracterizadas
- como o projeto será traduzido em uma linguagem de programação
- como os testes serão efetuados

Fase de Manutenção do Processo de Software

focaliza as "mudanças" que ocorrerão depois que o software for liberado para uso operacional



A fase de manutenção reaplica os passos das fases de definição e desenvolvimento, mas faz isso no contexto de um software existente.



Fase de Manutenção do Processo de Software

As mudanças estão associadas com:

- *correção de erros/defeitos*
- *adaptações exigidas conforme o ambiente do software evolui*
- *Aperfeiçoamentos para incluir funções adicionais que trarão benefícios para os clientes*
- *Modificações preventivas para evitar erros e facilitar modificações futuras.*

Atividades de Apoio ao Processo de Software

As três fases genéricas do processo de software são complementadas por uma série de atividades de apoio.



As atividades de apoio são aplicadas durante toda a engenharia do software

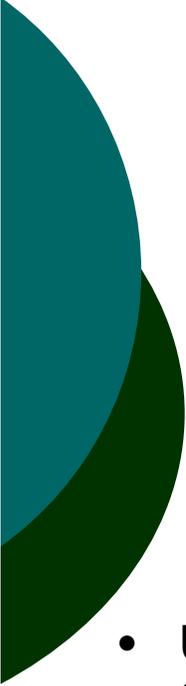
Atividades de Apoio ao Processo de Software

Atividades típicas de Apoio são:

- Controle e Acompanhamento do Projeto de Software
- Revisões Técnicas Formais
- Garantia de Qualidade de Software



- Gestão de Configuração de Software
- Preparação e Produção de Documentos
- Gestão de reutilização
- Medições
- Gestão de risco

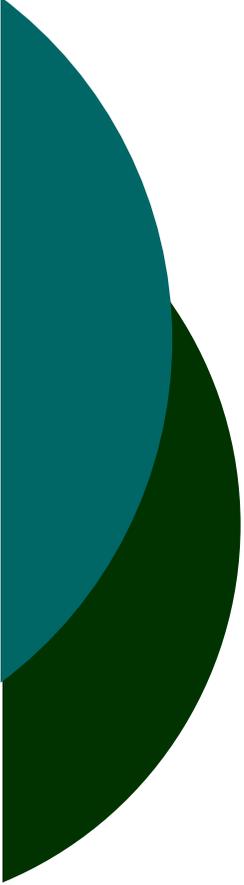


Exercício

- Baseando-se nas atividades Genéricas de um processo de Software, que acabamos de ver, crie um modelo para a seguinte situação:
 - Uma equipe iniciante de desenvolvedores de Software de 15 pessoas.
 - Um cliente pede para construir um software para o controle de estoque e de vendas da empresa. A empresa é considerada de médio porte, com cerca de 800 funcionários.
 - O prazo de entrega é de 6 meses

Descreva as fases do seu Modelo e faça uma figura para ilustrar o fluxo de uma fase para a outra

- Para lembrar das fases:
 - fase de definição
 - fase de desenvolvimento
 - fase de manutenção
 - atividades de apoio



Modelos de Processo de Software

Modelos de Processo de Software

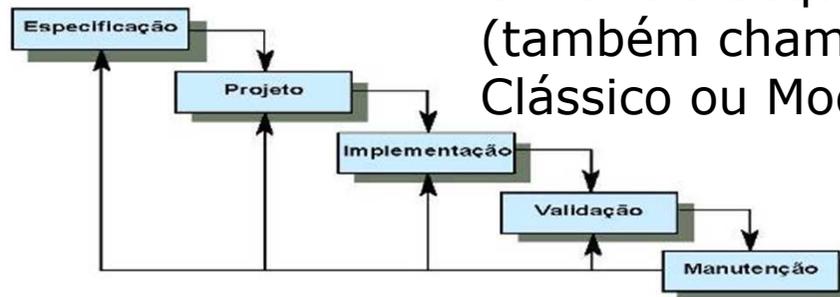
Existem vários modelos de processo de software (ou paradigmas de engenharia de software)



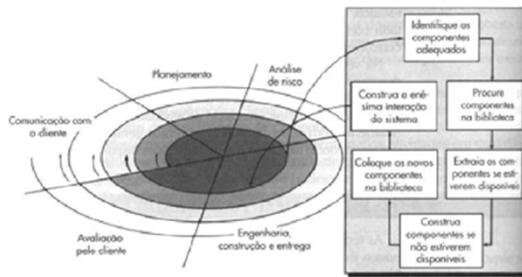
Cada um representa uma tentativa de colocar ordem em uma atividade inerentemente caótica

Modelos de Processo de Software

O Modelo Seqüencial Linear
(também chamado Ciclo de Vida Clássico ou Modelo Cascata)

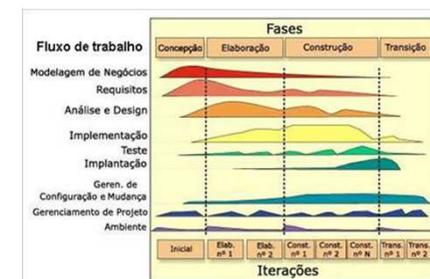


O Modelo Espiral



O Modelo Baseado em Componentes

O Paradigma de Prototipação



Processo Unificado