



Engenharia de Software



Bibliografia

- Engenharia de Software, *Roger Pressman*, 6.ed. McGraw-Hill, 2006.
- Engenharia de Software. *Sommerville*, 8.ed. Addison Wesley, 2007.



Introdução

- Hoje em dia todos dependem dos sistemas com base em computadores.
- O Software representa uma grande e crescente proporção do custo total do Sistema Computacional.



Introdução

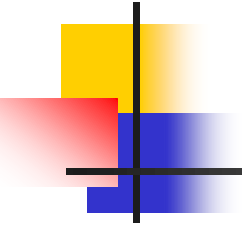
- Ainda hoje se desenvolve software sem qualidade e com sérios problemas de usabilidade. O que acarreta custo elevado e insatisfação.
- Produzir software de um modo que apresente uma boa relação custo-benefício é essencial: QUALIDADE!



A importância do Software

Durante as 3 primeiras décadas da era do computador, o principal desafio era desenvolver um **HARDWARE** de baixo custo e alto desempenho.

Hoje, o desafio é melhorar a qualidade e reduzir os custos das soluções baseadas em **SOFTWARE!**



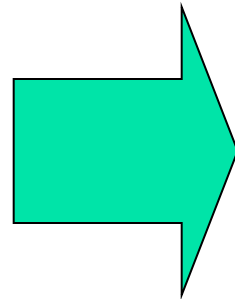
"O Software ultrapassou o Hardware como chave para o sucesso de muitos sistemas baseados em computador"

(Pressman)



A Evolução do Software

Computação



- Nova
Revolução
Industrial
(*Toffler*)
- 3a. Onda



Revolução Industrial: Primeira Onda

- Ferro
- Máquina a vapor:
 - Mecanização da indústria têxtil
 - Tear Mecânico (Cartão Perfurado)

★ Têxteis, Carvão e Ferro



Revolução Industrial: Segunda Onda

- Aço
- Locomotiva a Vapor (Rede de Transporte)
- Máquina de Costura (SINGER)
- Motor a combustão interna
- Desemprego e fim da escravidão



Revolução Industrial: Terceira Onda

- Energia Nuclear
- Uso Industrial/Comercial da Eletricidade
- Computadores Eletrônicos
- Transistor

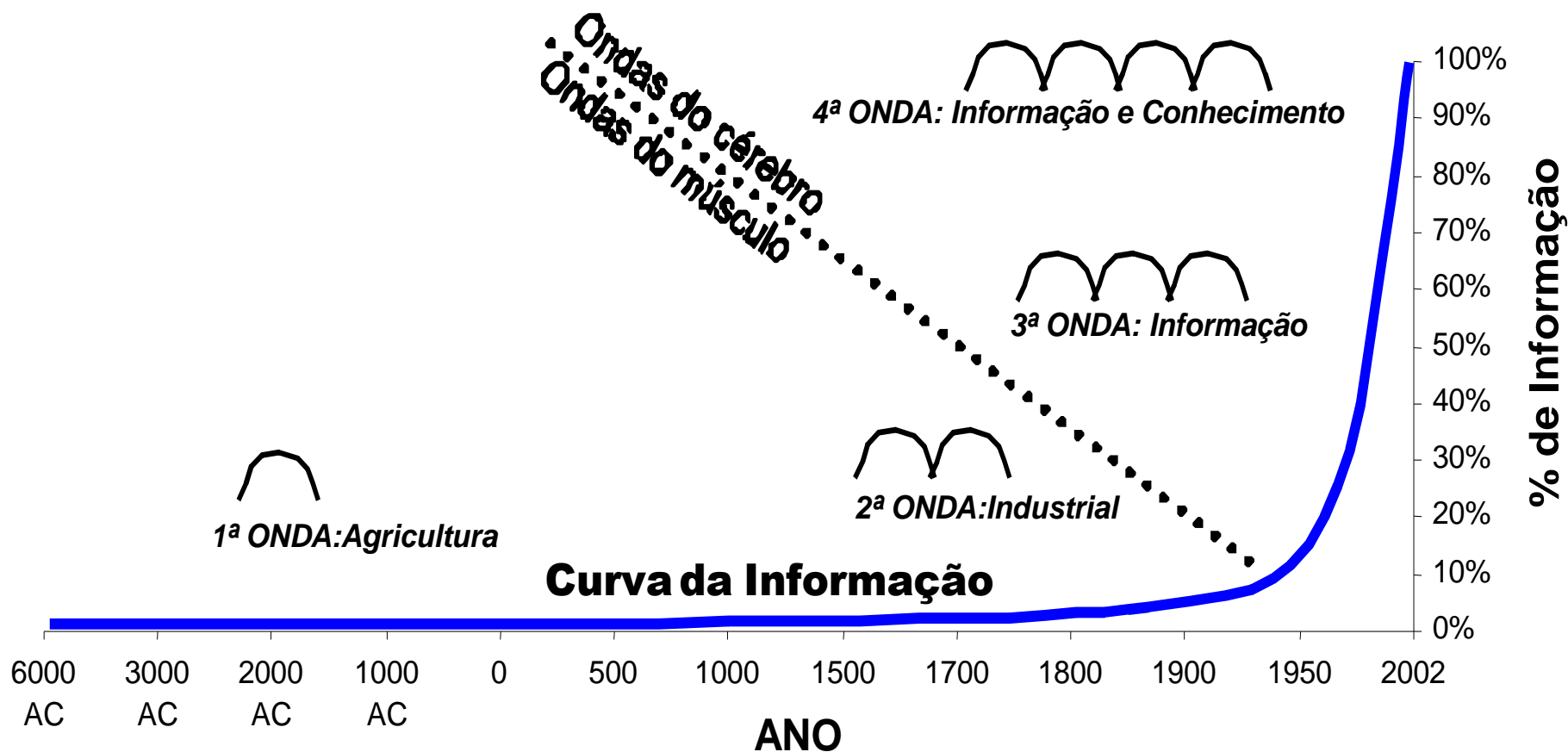
Sociedade
Industrial



Sociedade
da Informação

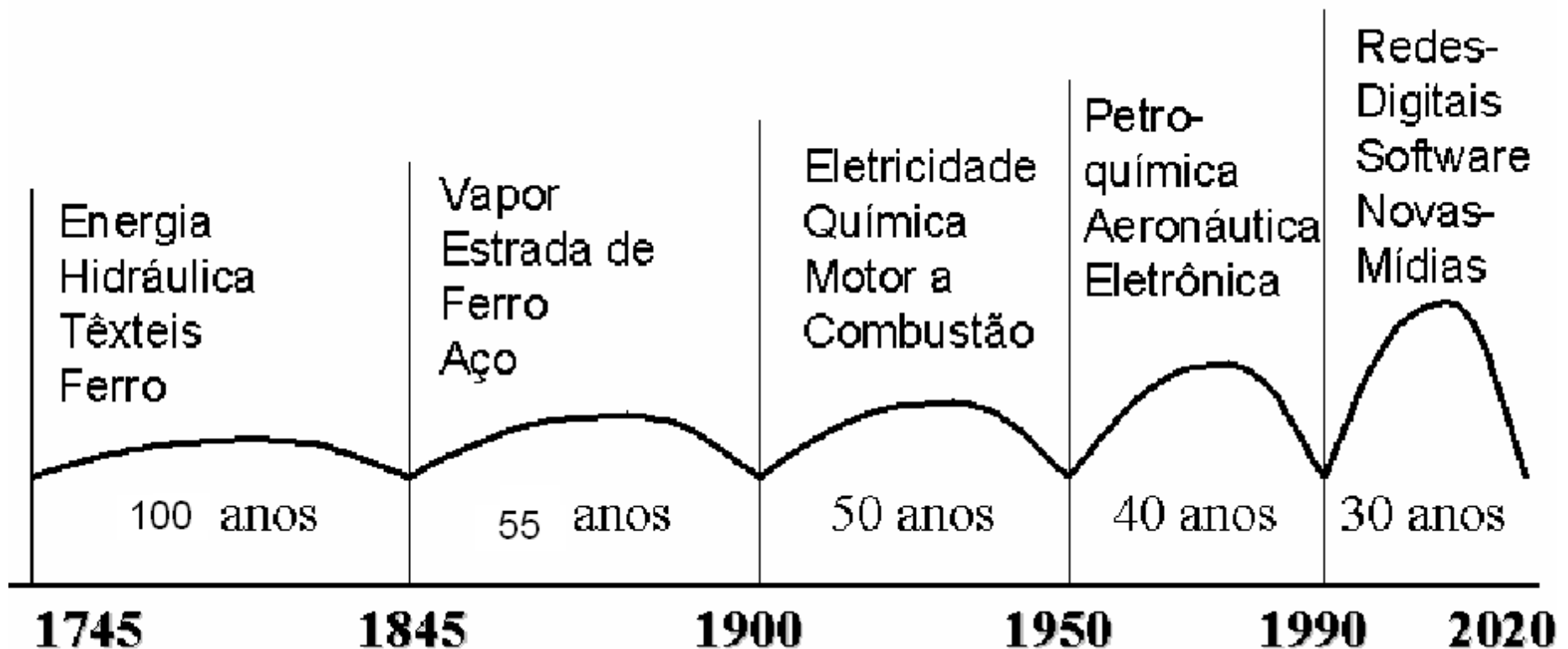
Conhecimento: Quarta Onda

■ Informação na Gestão do Conhecimento





Ondas de Schumpeter





Economia baseada na Informação

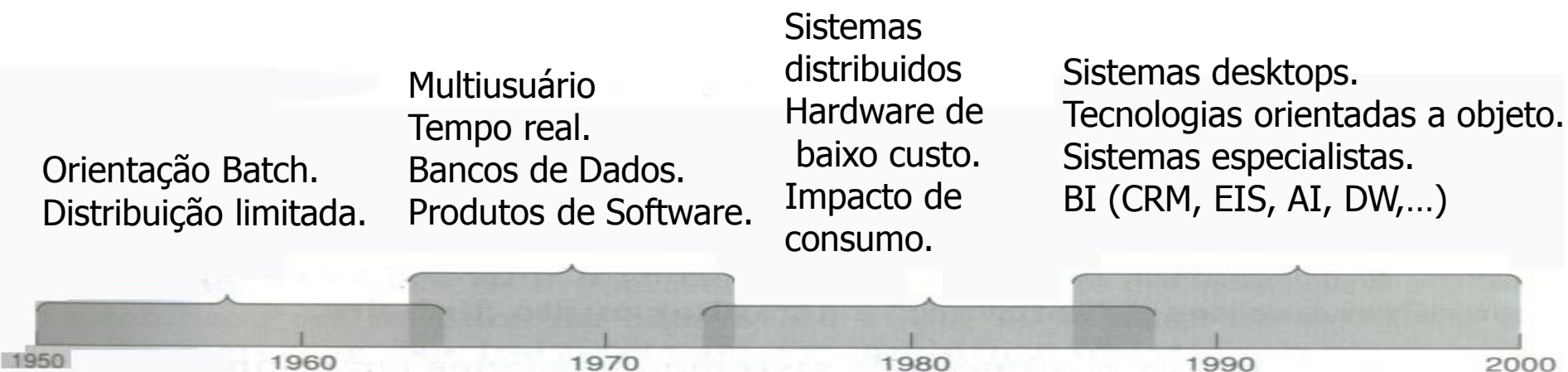
- A mudança de uma sociedade industrial para uma baseada na informação é uma Radical Mudança Econômica:
 - Informação possui mais valor que o bem produzido

Informação:

Estratégia empresarial → COMPETITIVIDADE

A evolução do software

- Software é dividida em 4 Eras:
 - Primeiros anos 1950 - 1965
 - Segunda Era 1965 - 1975
 - Terceira Era 1975 - 1988
 - Quarta Era 1988 - ...





O que é Software?

- Definição e composição:
 - 1º instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados;
 - 2º estruturas de dados que permitem a manipulação das informações;
 - 3º documentos que descrevem a operação e uso dos programas.



Software

Software = Parte Lógica do Computador.

Software não é apenas o programa mas também toda a documentação associada e os dados de configuração.

Os Engenheiros de Software se ocupam em desenvolver **produtos de software** que possam ser **vendidos** a um cliente:

- Produtos Genéricos: Sistemas produzidos por uma organização e vendidos no mercado a qualquer cliente capaz de adquiri-los.
- Produtos Sob-Encomenda: São sistemas encomendados a um cliente em particular.



Características do Software

- Engenharia:
 - Arte das construções, com base no conhecimento científico e empírico (experiência), adequada ao atendimento das necessidades humanas.



Características do Software

- O Software é desenvolvido ou projetado por **engenharia**, não manufaturado no sentido clássico:
 - Custos são concentrados no trabalho de engenharia.
 - Projetos de software não podem ser geridos como projetos de manufatura.
 - “Fábrica de Software”



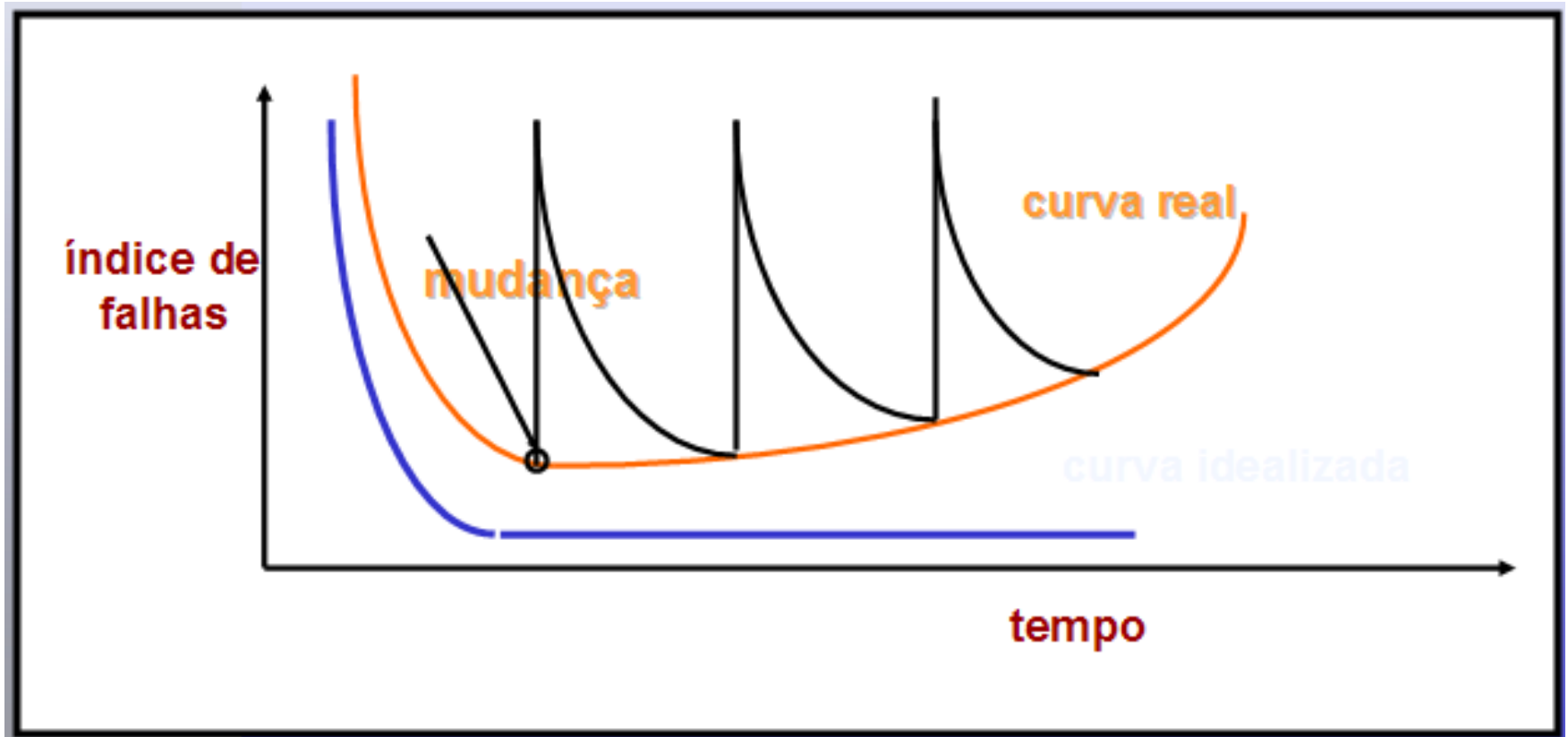
Características do Software

- Software não desgasta!
- Software não é sensível aos problemas ambientais que fazem com que o hardware se desgaste.
- Toda **falha** indica **erro de projeto ou implementação**: manutenção do Software é mais complicada que a do Hardware.

Curva de falhas do HW



Curva de falhas do SW





Características do Software

- A maioria dos softwares é feita sob medida e não montada a partir de componentes existentes.
- Diferentes Hardwares.
- Situação esta mudando:
 - Reusabilidade (diminui custos e melhora projetos).
 - Orientação a objetos



Atributos de um Bom Software

Atributos que refletem a qualidade do software.

Facilidade de Manutenção	O Software deve ser escrito de modo que possa evoluir para atender as necessidades mutáveis dos clientes.
Nível de Confiança	O software confiável não deve ocasionar danos físicos ou econômicos no caso de defeito.
Eficiência	Não deve desperdiçar os recursos do sistema, como memória e ciclo de processador.
Facilidade de Uso	Deve ser utilizado pelo usuário, para quem foi projetado, sem esforços indevidos. Interface e Documentação.



Aplicações de software

- Software de sistemas
 - Coleção de programas escritos para servir outros programas
 - Compiladores, editores.
- Software de tempo real
 - Monitora, analisa e controla eventos do mundo real à medida que eles ocorrem.
- Software comercial
 - Maior área de aplicação
 - Folha de pagamento, controle de estoque,...



Aplicações de software

- Software científico e de engenharia
 - Softwares que processam números
 - Astronomia, biologia molecular, aero-espacial, mecânica,...
- Software embutido
 - Reside nas memórias ROM
 - Funções limitadas e particulares – controle de teclado para um forno, funções digitais em automóvel.
- Software para Web
 - Executam scripts, procuras,...



Uma Crise no horizonte

- A indústria de Software tem tido uma “crise” que a acompanha há quase 30 anos.
- Menos de 40 anos de experiência.
- Problemas não se limitam ao software que não funciona adequadamente, mas abrange:
 - desenvolvimento, testes, manutenção, etc.
- **Tudo deve ser feito para ontem** → Sistemas de Software não acompanharam a velocidade do mundo dos negócios.

Ariane 5



Ariane 5

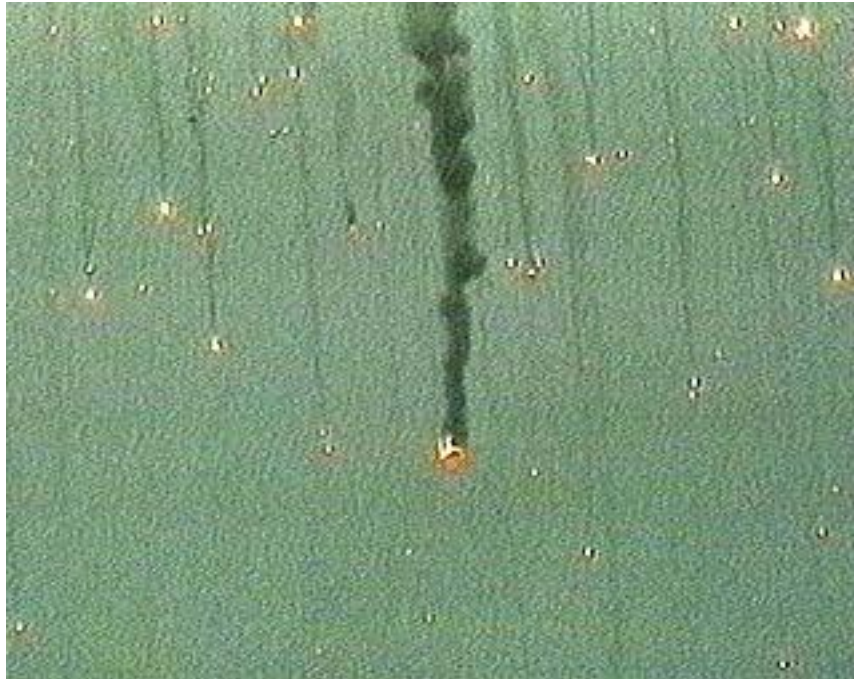


- Projeto da Agência Espacial Européia que custou:
 - 10 anos.
 - US\$ 8 Bilhões.
- Capacidade 6 toneladas.
- Garante supremacia européia no espaço.

Vôo inaugural em 4/7/96



BUUMMMmmm



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



O que aconteceu?

- Fato: o veículo detonou suas cargas explosivas de autodestruição e explodiu no ar. Por que?
- Porque ele estava se quebrando devido às forças aerodinâmicas. Mas por que?
- O foguete tinha perdido o controle de direção. Causa disso?
- Os computadores principal e back-up deram shut-down ao mesmo tempo.
- Por que o Shut-down ? Ocorreria um *run time error* (out of range, overflow , ou outro) e ambos computadores se desligaram. De onde veio este erro?
- Um programa que convertia um valor em ponto flutuante para um inteiro de 16 bits recebeu como entrada um valor que estava fora da faixa permitida (maior que 16 bits).



Therac-25

- Equipamento de Radioterapia.
- Entre 1985 e 1987 se envolveu em 6 acidentes, causando mortes por overdoses de radiação.
- Software foi adaptado de uma antecessora, Therac-6:
 - falhas por falta de testes
 - falta de documentação

Denver International Airport



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

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



Satélite – monitorava Marte

- Outubro 1999: acreditou-se que o satélite que monitorava o clima de Marte de US\$125mi, estava perdido no espaço devido a um erro de uma simples conversão de dados onde foram utilizadas unidades inglesas ao invés de métricas.



A quase guerra nuclear

- Erros de software no módulo de pré-aviso de um sistema de monitoração, quase detonaram uma guerra nuclear em 1983.
- O software não considerou falsos os sinais de mísseis causados pelo reflexo do sol da manhã no topo de determinadas formações de nuvens.
- Um comandante soviético simplesmente desconfiou que poderia ser alarme falso e evitou a guerra.



Quais são os problemas?

- A sofisticação do software ultrapassou nossa capacidade de construção.
- Nossa capacidade de construir programas não acompanha a demanda por novos programas.
- Nossa capacidade de manter programas é ameaçada por projetos ruins.



Causas óbvias

- Não dedicamos tempo para coletar dados sobre o desenvolvimento do software - resulta em estimativas “a olho”.
- Comunicação entre o cliente e o desenvolvedor é muito fraca.
- Falta de testes sistemáticos e completos.



Causas menos óbvias

- O Software é desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico.
- Gerentes sem “*background*” em desenvolvimento de Software.
- Profissionais recebem pouco treinamento formal.
- Falta de métodos e automação.



Mitos do Software - Administrativos

- Um manual oferece tudo que se precisa saber
- Computadores de última geração solucionam problemas de desenvolvimento
- Ferramentas que são o estado-da-arte
- Estamos atrasados: contrate mais programadores
- Relaxe, vamos terceirizar o projeto de software



Mitos do Software - Cliente

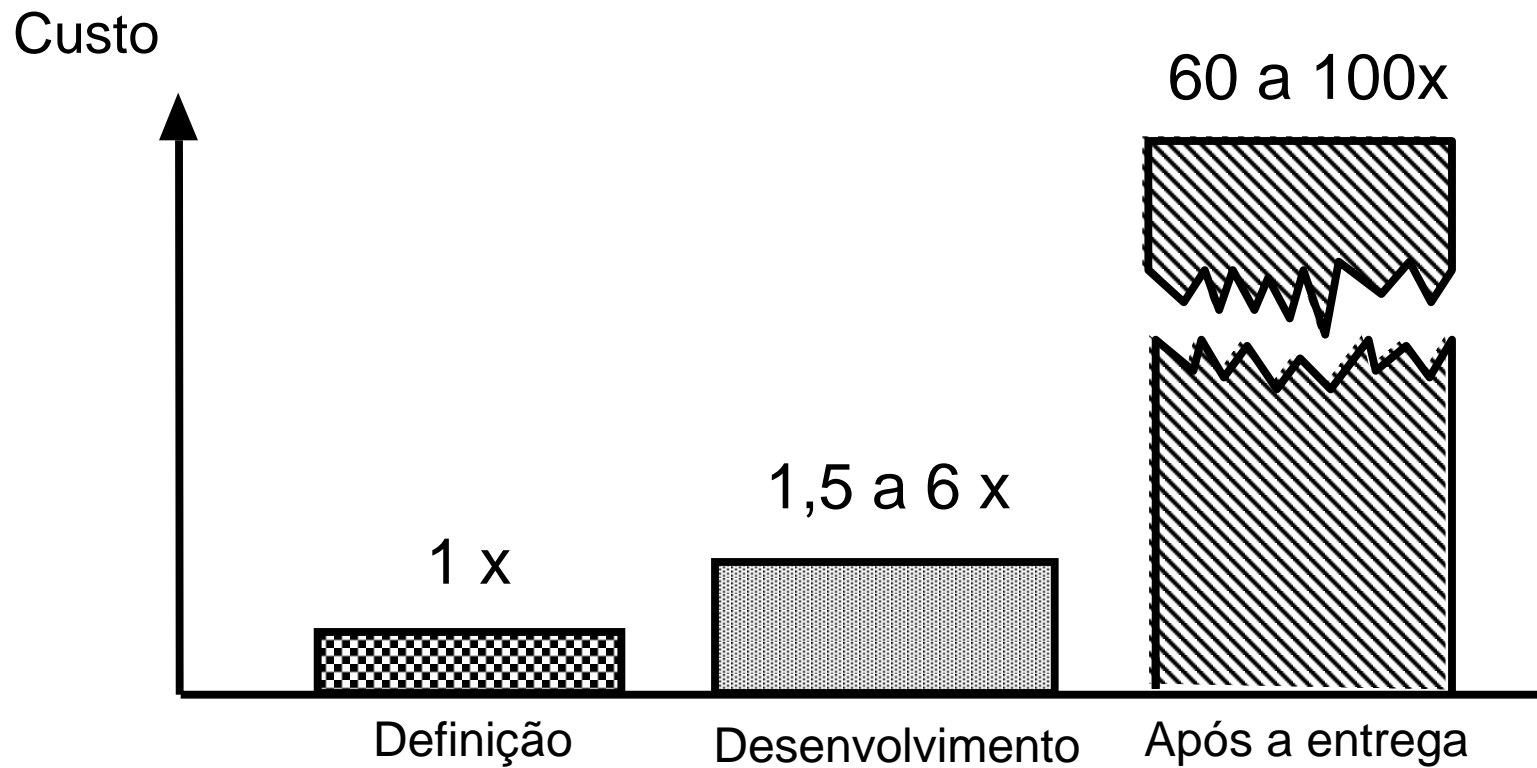
- Estabelecimento de objetivos é suficiente para começar o software.
- Requisitos mudam constantemente, mas o software é flexível.



Mitos do Software - do Profissional

- Um programa está terminado ao funcionar.
- Quanto mais cedo escrever o código, mais rápido terminarei o programa.
- Só posso avaliar a qualidade de um programa em funcionamento.
- O único produto a ser entregue, é o software

Impacto da modificação





Engenharia de Software

Definição

Engenharia de Software é o estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais



Engenharia de Software.

É uma disciplina da Engenharia que se ocupa de todos os aspectos da produção de software.

- **Disciplina da Engenharia:**

Uso de processos, métodos, teorias e ferramentas na procura de soluções de problemas.

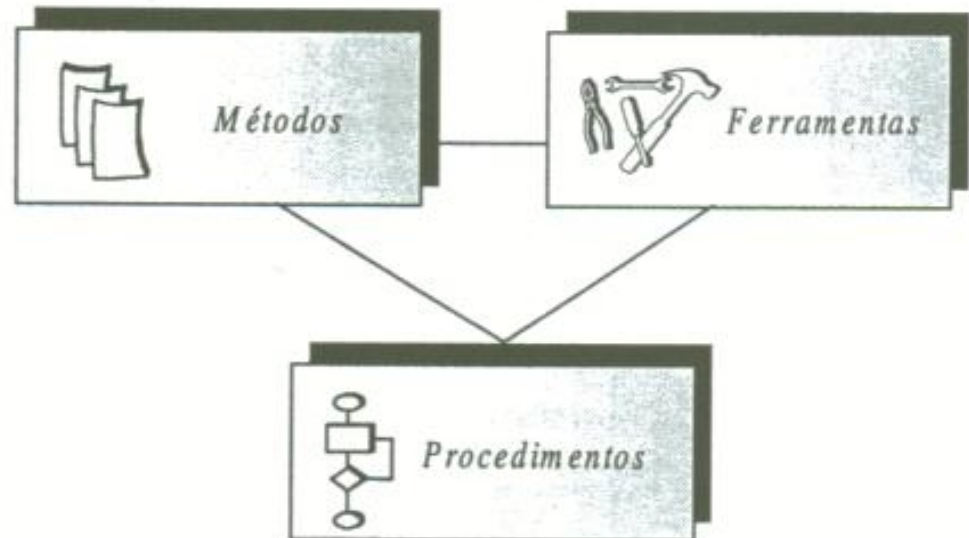
- **Todos os Aspectos da Produção de Software:**

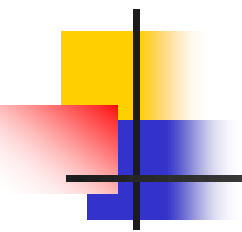
Não se dedica só ao processo técnico, mas também a atividades como o gerenciamento de projetos de software e o desenvolvimento de ferramentas, métodos e teorias que ajudem a produção de software.

Engenharia de Software: Abrangência

■ 3 elementos fundamentais:

- **Métodos: “como fazer”** – Fornecem a técnica para construção do software, dentre os quais se destacam o método **estruturado** e o **orientado a objeto**.
- **Ferramentas:** apoio automatizado aos métodos, dentre os quais podem ser citadas as ferramentas CASE.
- **Procedimentos:** elo de ligação entre os métodos e as ferramentas, permitindo o desenvolvimento de forma racional e no prazo estipulado.





Quais são os desafios da Engenharia de software ?

- Sistemas legados
 - Sistemas velhos que devem ser mantidos e atualizados.
- Heterogeneidade
 - Os sistemas são distribuídos e incluem uma combinação de HW e SW.
- Entrega
 - Há uma constante pressão para uma entrega mais rápida de software.



Perguntas que Engenharia de Software quer responder:

- Porque demora tanto para concluir um projeto (não cumprimos prazos) ?
- Porque custa tanto (uma ordem de magnitude a mais) ?
- Porque não descobrimos os erros antes de entregar o software ao cliente ?
- Porque temos dificuldade em medir o progresso enquanto o software está sendo desenvolvido ?



Quais são os custos da engenharia de software?

- 60% dos custos são custos de desenvolvimento, 40% são custos de testes.
- Os custos dependem do tipo de sistema a ser desenvolvido e dos requisitos do sistema tais como desempenho e viabilidade.
- A distribuição dos custos depende do modelo de desenvolvimento usado.



Responsabilidade ética e profissional

- A engenharia de software envolve mais responsabilidades do que a simples aplicação das aptidões técnicas.
- engenheiros de software devem comportar-se de forma honesta e eticamente responsável se quiserem ser profissionais respeitáveis.
- O comportamento ético é muito mais do que apenas cumprir a lei.
 - *Confidencialidade.*
 - *Direitos de propriedade intelectual*



Processo de um Software

É um conjunto de atividades e resultados associados que geram um produto de software. Há 4 atividades de processo fundamentais;

- **Especificação do Software:** A funcionalidade do software e as restrições em sua operação devem ser definidas.
- **Desenvolvimento de Software:** O software deve ser produzido de modo que atenda a suas especificações.
- **Validação do Software:** O software tem de ser validado para garantir que ele faz o que o cliente deseja.
- **Evolução do Software:** O software deve evoluir para atender às necessidades mutáveis do cliente.

Paradigmas de Engenharia de Software

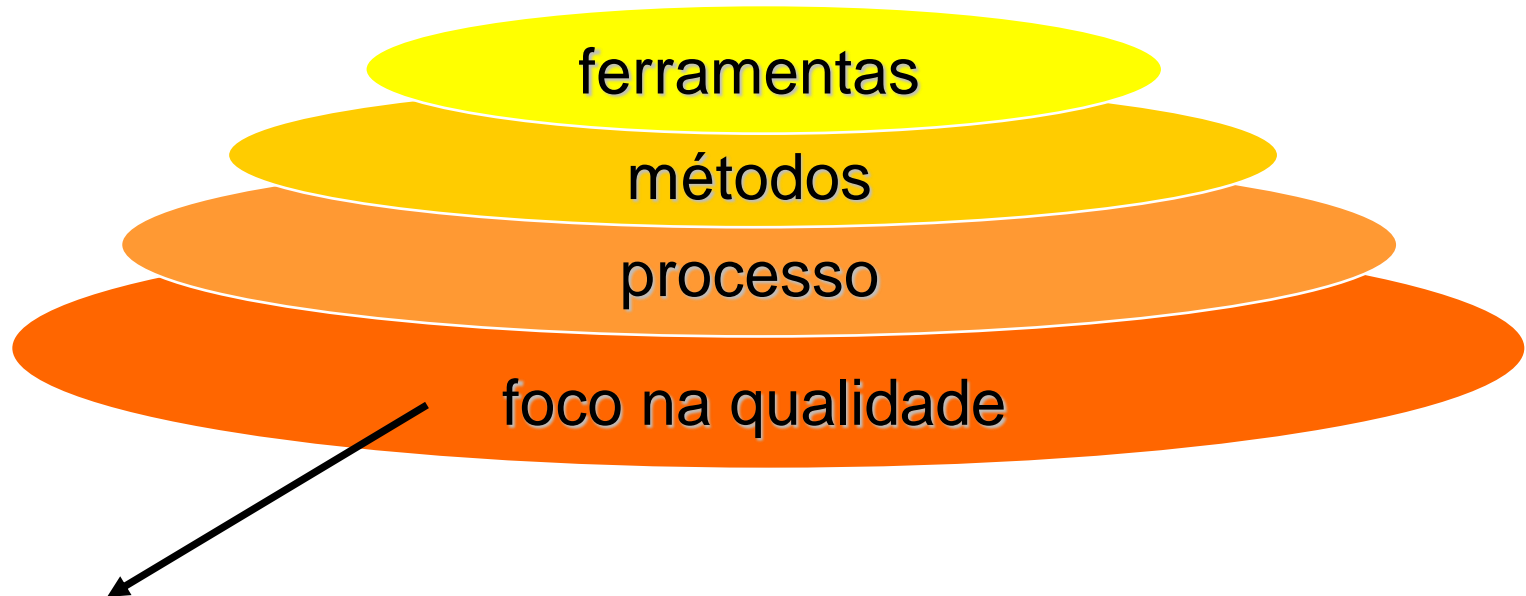
- Tecnologia em camadas:

Engenharia de Software



Paradigmas de Engenharia de Software

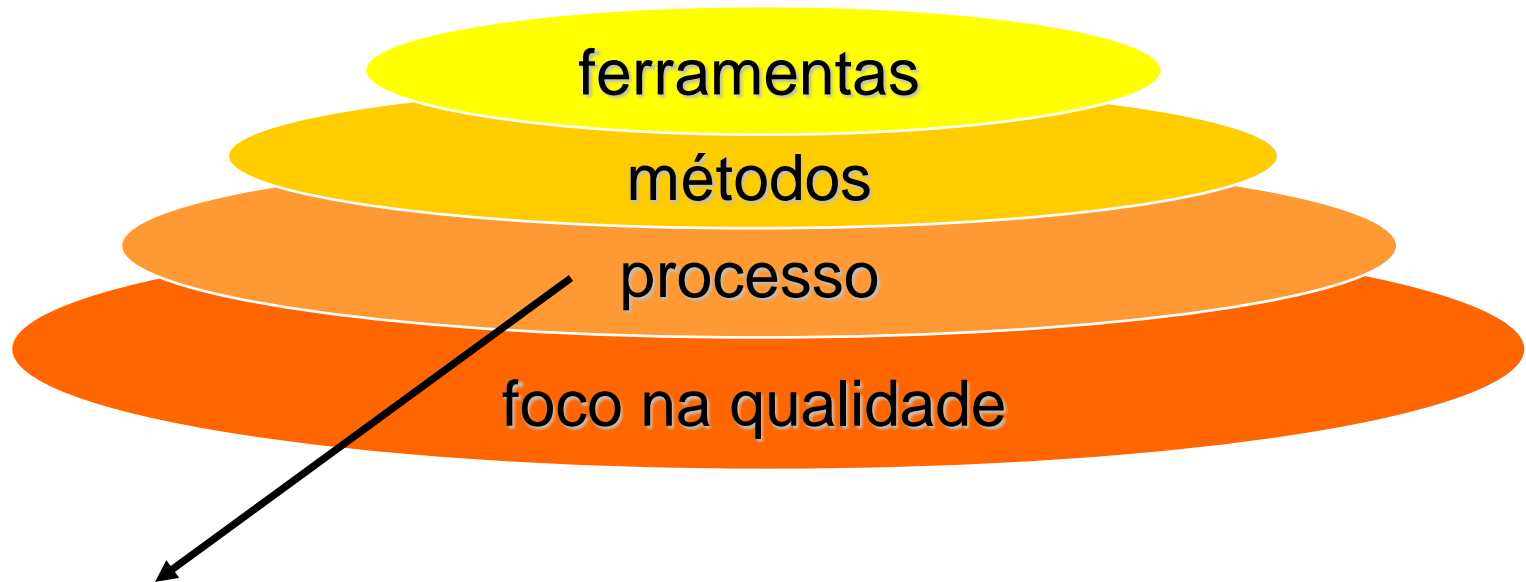
- Tecnologia em camadas:



É a base e o porque de utilizarmos um processo de software

Paradigmas de Engenharia de Software

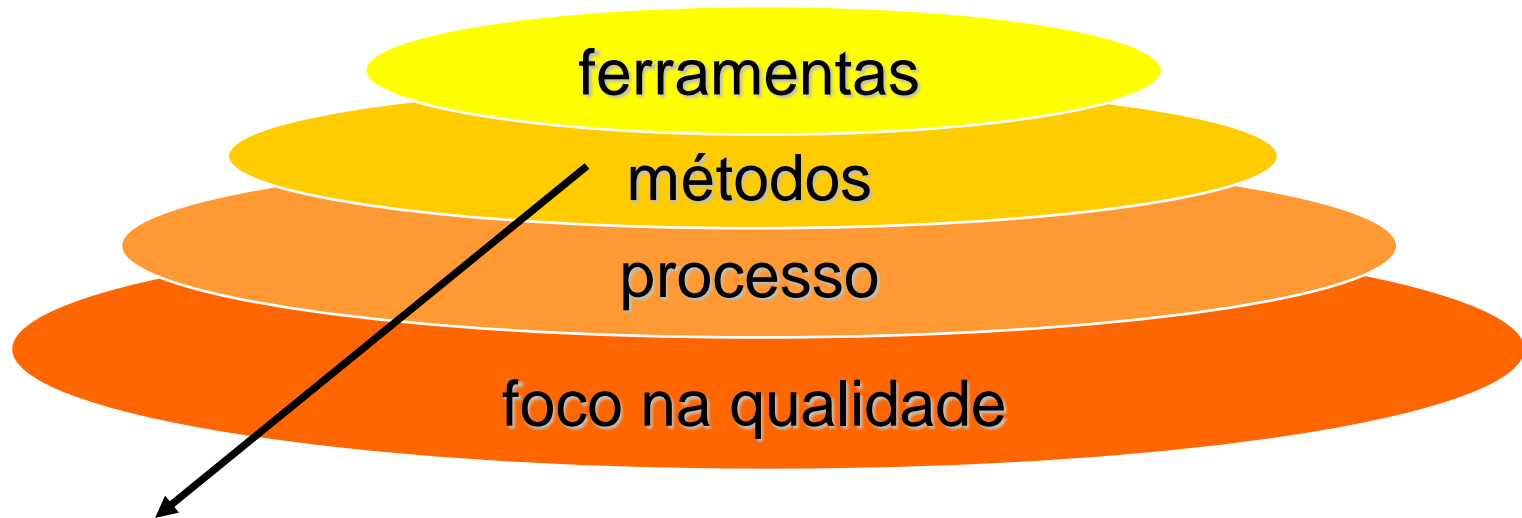
- Tecnologia em camadas:



Permite um desenvolvimento de software racional e em tempo

Paradigmas de Engenharia de Software

- Tecnologia em camadas:

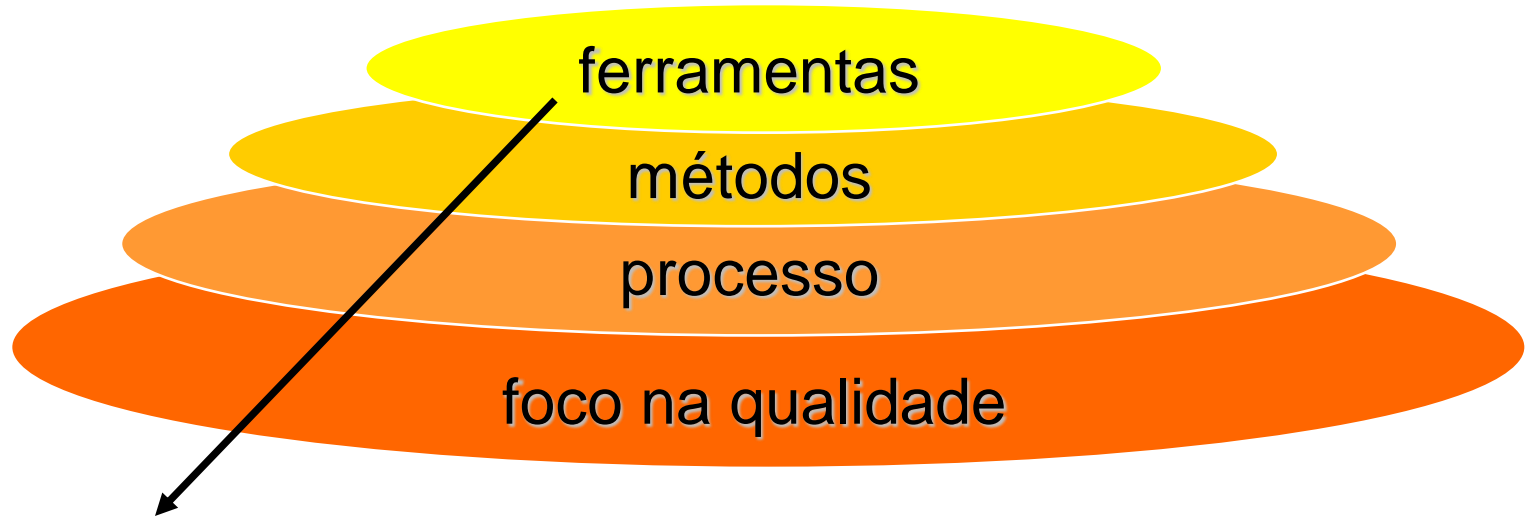


É o “como fazer”.

Englobam um conjunto de tarefas que inclui análise de requisitos, projeto, implementação, teste e manutenção.

Paradigmas de Engenharia de Software

- Tecnologia em camadas:



Dão suporte automatizado ou semi-automatizado aos métodos (CASE).



CASE

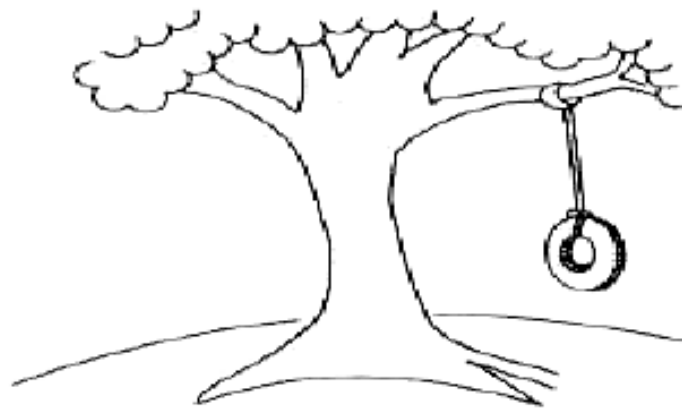
- Computer Aided Software Engineering.
(Engenharia de Software com auxílio do Computador)
- Diferentes tipos de programas utilizados para apoiar as atividades de processo de software, como a análise de requisitos, a modelagem do sistema, a depuração, testes,...
- As ferramentas CASE podem também incluir um gerador de códigos que, automaticamente, origina código-fonte a partir do modelo de sistemas e alguma orientação de processo.



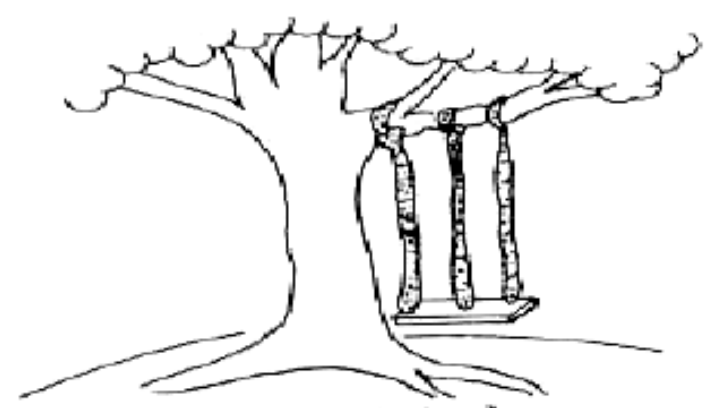
Paradigmas de Engenharia de Software

- Existem dezenas
- Alguns deles:
 - Ciclo de Vida Clássico
 - Prototipagem
 - Espiral
 - Open Source
 - RUP

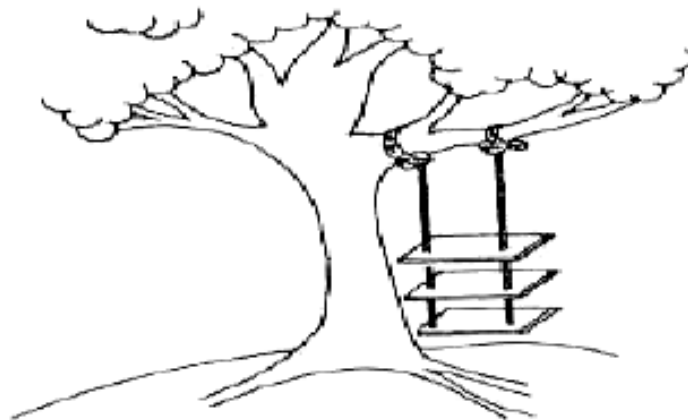
Mas todos tentam resolver o seguinte problema...



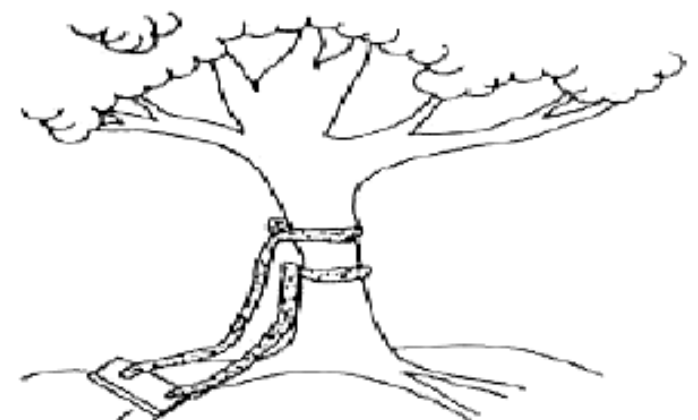
O que o usuário queria



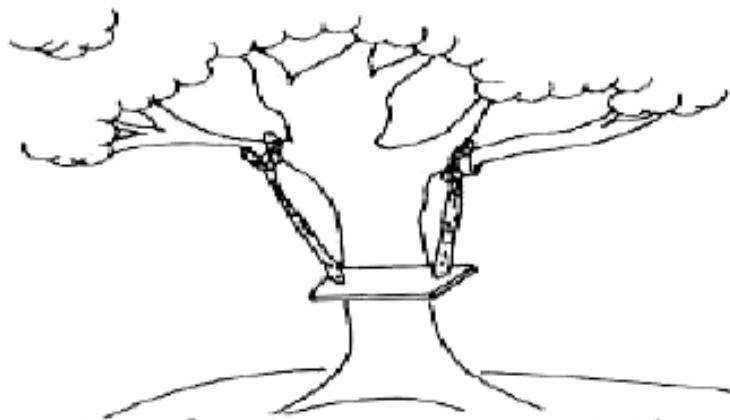
O que foi solicitado



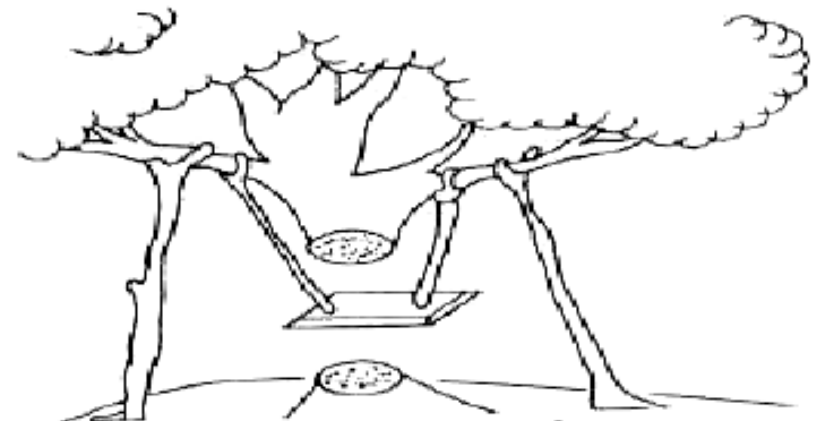
O que foi proposto



O que foi projetado



O que foi desenvolvido



O que foi implantado



Ciclo de Vida Clássico: modelo *Cascata* (*Waterfall*)

- Baseado em projetos de engenharia clássicos
- Fases:
 1. Levantamento de requisitos
 2. Análise
 3. Projeto
 4. Implementação
 5. Testes
 6. Manutenção



Ciclo de Vida Clássico

modelo *Cascata (Waterfall)*

- Modelo mais antigo e mais usado.
- Requer uma abordagem seqüencial para o desenvolvimento de um software.



Ciclo de Vida Clássico modelo *Cascata (Waterfall)*

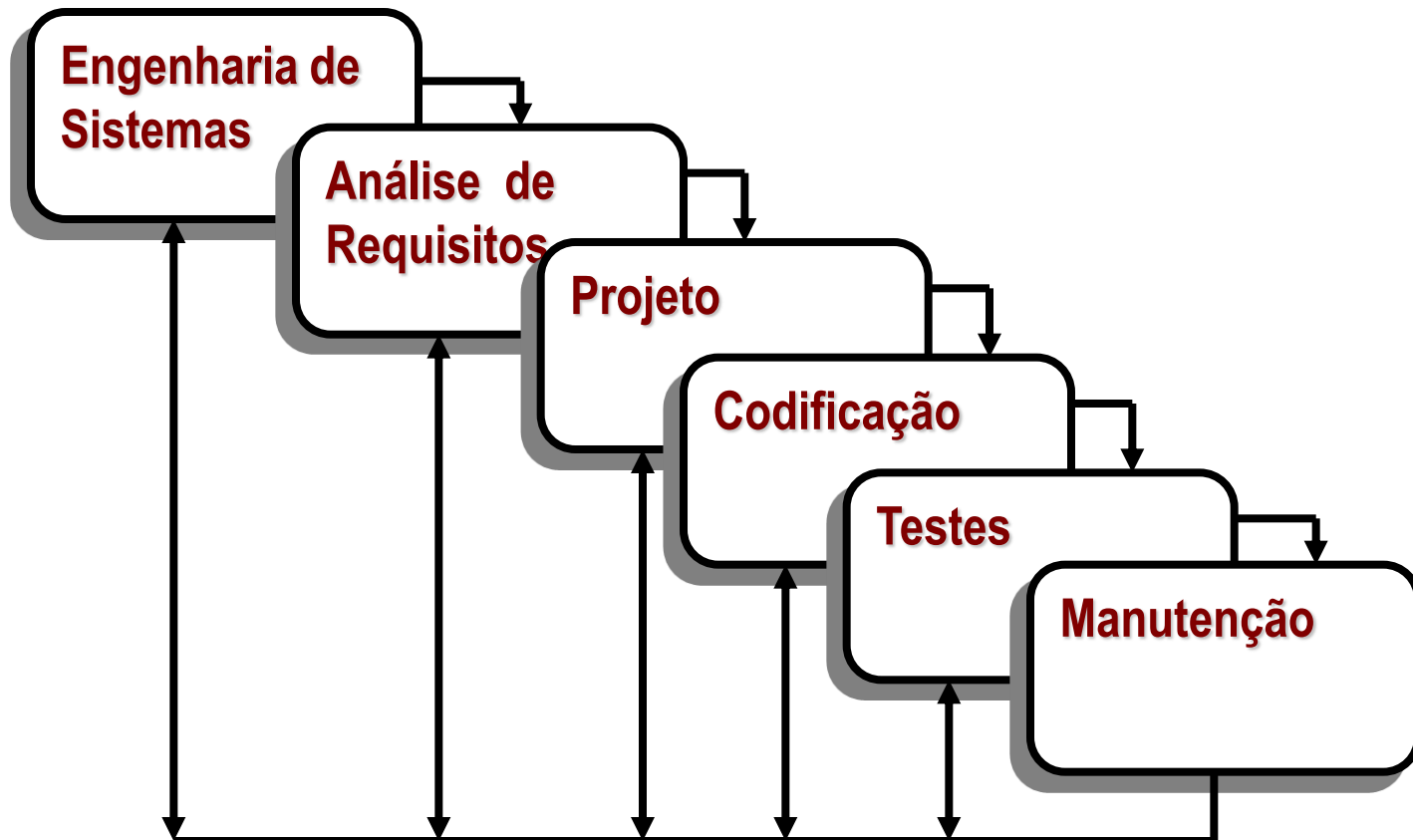
- Uso absurdamente linear



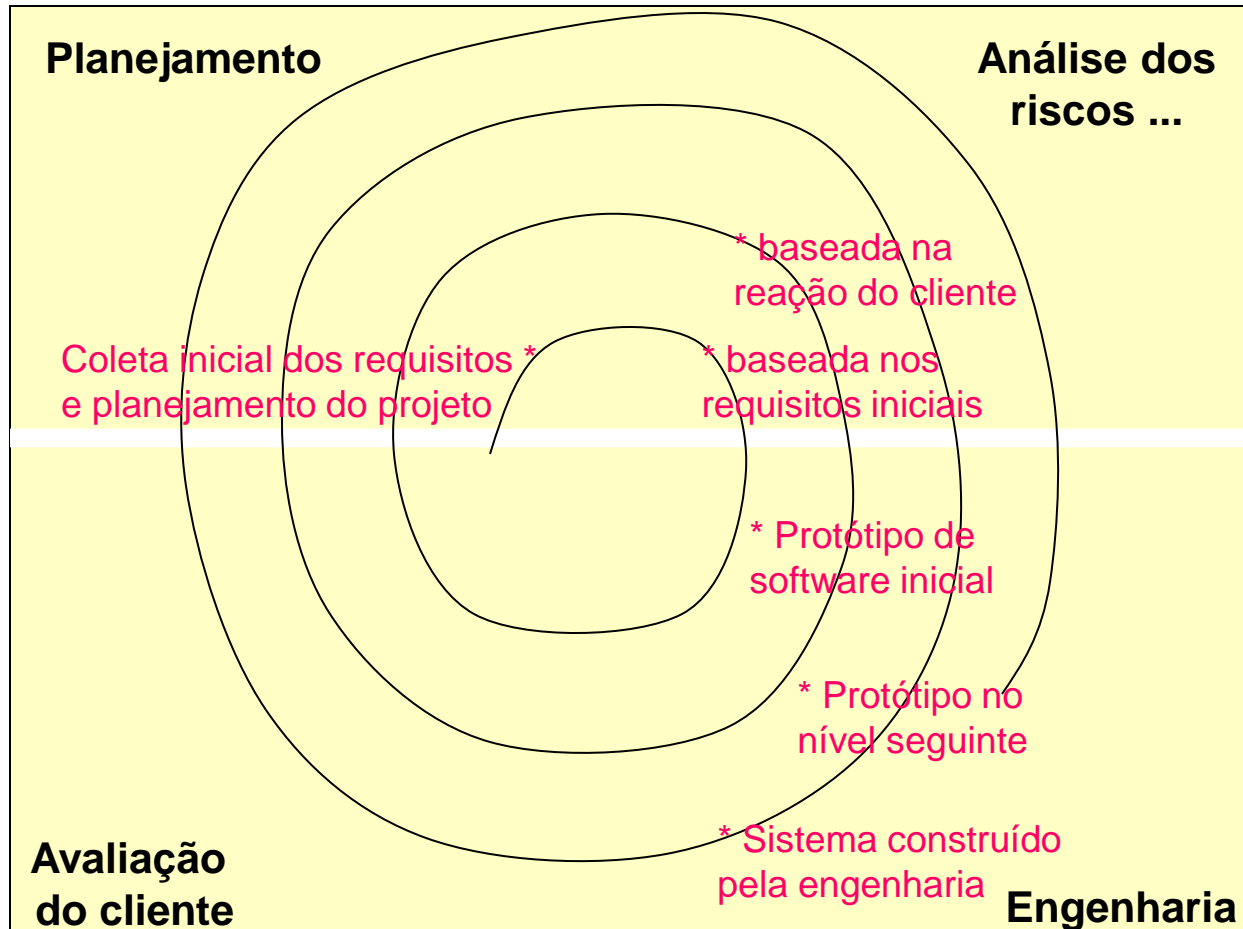
Ciclo de Vida Clássico

modelo *Cascata* (*Waterfall*)

- Modelo prevê voltas



Modelo Espiral



Em cada arco da espiral, a conclusão da análise dos riscos resulta numa decisão de prosseguir/ não prosseguir.



Fases do modelo Espiral

- Definição dos objetivos
 - Especificação dos objetivos específicos desta fase.
- Análise dos riscos
 - Identificação e solução dos principais riscos
- Desenvolvimento e validação
- Planejamento
 - O projeto é revisto e se define planos para a próxima “volta da espiral”



Conclusão

- Software é elemento chave para o sucesso.
Mas:
 - Software não é hardware.
 - Software não é fácil.
 - Software mata.
- Valorize o Software...



Conclusão

- Software é elemento chave para o sucesso.
Mas:
 - Somente 26% dos softwares são bem sucedidos
 - 31% nunca são completados
 - 53% custam quase 200% da estimativa inicial
 - São gastos Centenas de Bilhões de Dólares somente nos EUA



Conclusão

Software é o combustível sobre o qual negócios modernos rodam, governos funcionam, e sociedades se tornam melhor conectadas. Software tem nos ajudado a criar, acessar, e visualizar informações em modos e formas antigamente inconcebíveis.

Globalmente, o brilhante avanço do progresso em software tem ajudado a conduzir o crescimento da economia do mundo.



Conclusão

Em escala mais humana, produtos de software intensivo tem ajudado a curar doenças, dar voz aos mudos, mobilidade aos deficientes, e oportunidade aos menos capazes.

A partir de todas estas perspectivas, software é uma parte indispensável de nosso mundo moderno.

Grady Booch



Exercícios – Individual

1. Apresente exemplos de três produtos e de pelo menos um sistema em que o software, não o hardware, é o elemento que faz a diferença.
2. A Engenharia de Software auxiliada por computador é uma indústria crescente, pesquise 3 produtos CASE comercialmente disponíveis e apresente-os.
3. Quais dos paradigmas de engenharia de software seriam os mais indicados às suas aplicações de software?
4. Pesquise duas linguagens de quinta geração e apresente uma discussão resumida: Quão ampla é sua aplicabilidade?