



INTRODUÇÃO AOS SISTEMAS DE INFORMAÇÃO

Profa. Fernanda Oliveira

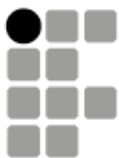
Bacharelado em Ciência da Computação

Instituto Federal de São Paulo Campus São João da Boa Vista

Resgatando conceitos...



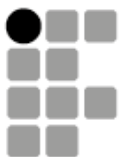
Adaptação do material elaborado pelo
Prof. Breno Lisi Romano



Sistema é como um conjunto de partes que interagem, visando um objetivo específico

Os principais fatores que merecem nossa atenção com relação a sistemas:

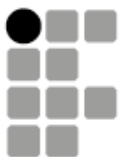
- ❖ Ambiente do sistema
- ❖ Principais objetivos do sistema
- ❖ Recursos utilizados pelo sistema
- ❖ Controle do Sistema



Todo conjunto de programas, procedimentos, dados e documentação associados a um **sistema** de computador, e não somente o programa em si.

Genéricos: sistemas produzidos por uma software house e vendido no mercado para qualquer cliente que queira comprá-lo. Exemplo: Office, Sistemas Contábeis, Pergamum (sistema para Bibliotecas).

Personalizados: encomendados e desenvolvidos especialmente para um cliente. As regras são definidas pelo cliente para atender suas necessidades de negócio. Um exemplo é o SUAP. É um software acadêmico desenvolvido de acordo com as regras do IF em âmbito nacional.



“Sistema de Informação (SI) é um conjunto de componentes inter-relacionados que coletam (ou recuperam), processam, armazenam e distribuem informações destinadas a apoiar a tomada de decisões, a coordenação e o controle em uma organização.”

Curva de falhas para hardware

A Figura mostra o índice de falhas em função do tempo para o **hardware**

A curva da banheira indica que o hardware exibe alto índice de falha no começo de seu ciclo de vida (estas falhas são atribuídas a defeitos de projeto e manufatura)



Os defeitos são corrigidos e o índice de falhas cai para um nível estável durante certo período de tempo

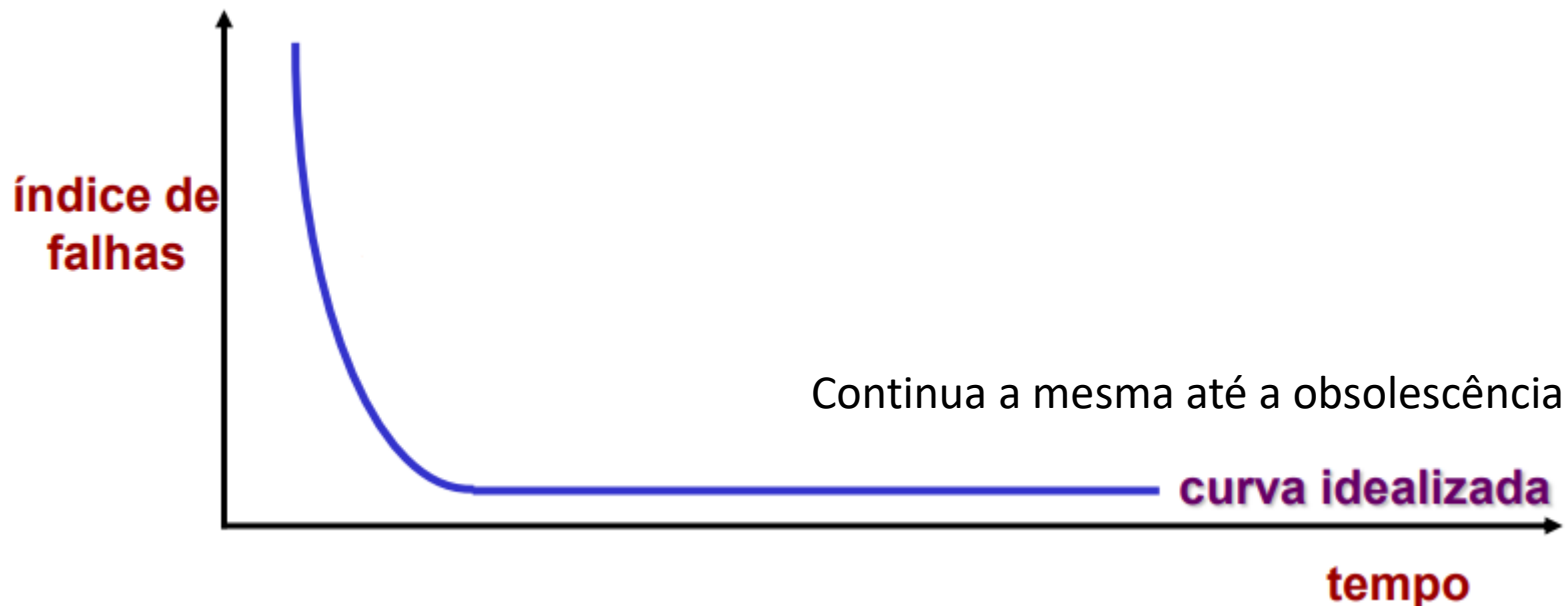
Com o passar do tempo o de falha aumenta conforme os componentes de hardware sofrem os efeitos cumulativos de poeira, vibração, abuso, temperaturas extremas e muitos outros males ambientais

Colocado de maneira simples, o hardware começa a se desgastar

Curva de falhas para software

O **software** não é sensível aos problemas ambientais que fazem com que o hardware se desgaste.

Teoricamente, portanto, a curva do índice de falhas para o software assumiria a forma representada na Figura



O alto índice de falhas no começo do ciclo de vida indica defeitos não identificados durante os testes.

Esses defeitos são corrigidos (espera-se que novos erros não sejam introduzidos) e a curva achata-se e se mantém até que o software se torne obsoleto.

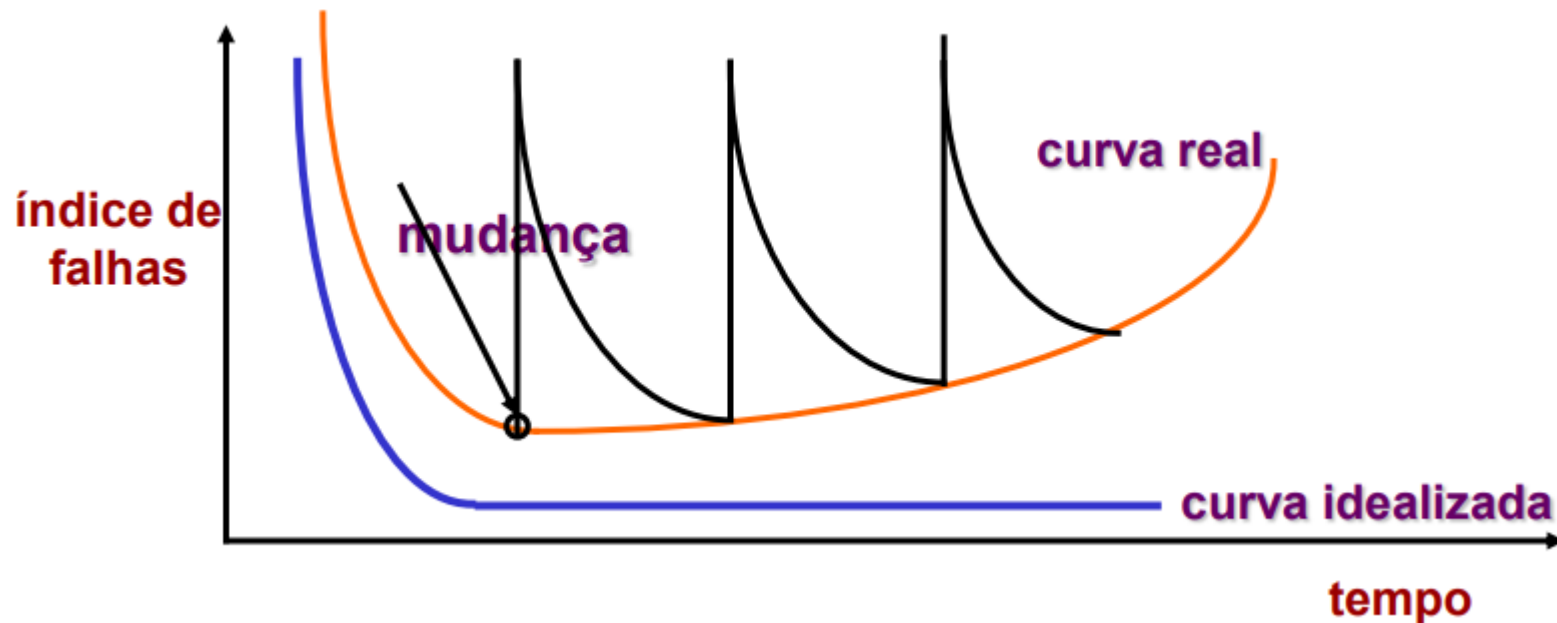
Portanto fica claro que o software não se desgasta. Todavia se deteriora!

Software não se desgasta, mas se deteriora!

Durante sua vida, o software sofrerá manutenção.


Nas atualizações do software novos defeitos são introduzidos e a curva de falhas vai crescendo e apresenta picos conforme ilustra a figura. Com o tempo o custo da manutenção se torna inviável.

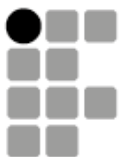
Toda falha de software indica um erro de projeto ou no processo por meio do qual o projeto foi traduzido em código executável por máquina.



Portanto a manutenção do software envolve consideravelmente mais complexidade do que a manutenção de hardware

Bugs históricos

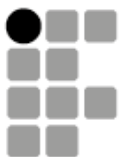
0800 Anttan started
1000 " stopped - anttan ✓
1300 (032) MP - MC ~~1.982147000~~
(033) PRO 2 2.130476415
convd 2.130676415
Relays 6-2 in 033 failed special speed test
in relay 11.000 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult + Adder Test.
1545  Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
1630 Anttan started.
1700 closed down.



Mariner I – 1962

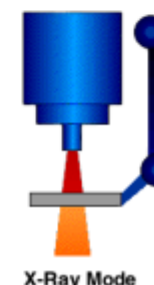
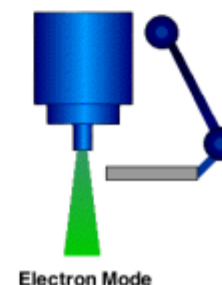
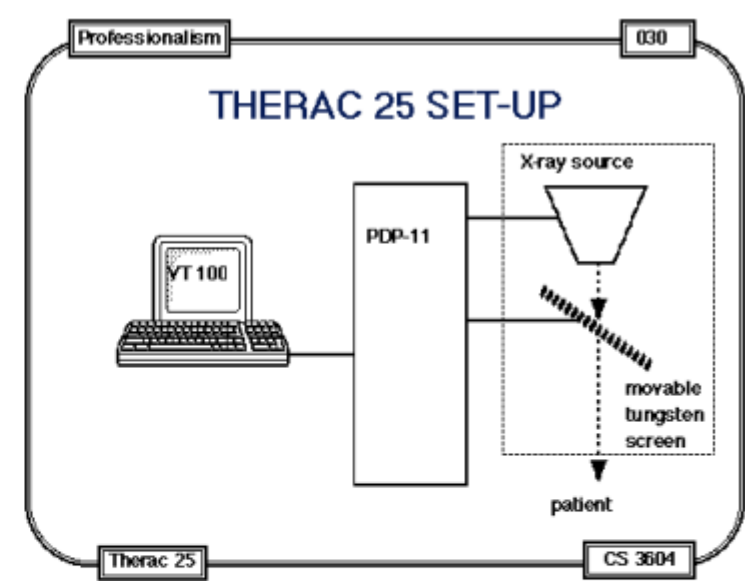
- Missão observar planeta Vênus
- Fórmula matemática foi equivocadamente transcrita para o computador
- Desviou do curso e foi destruído 4 min após lançamento
- Prejuízo: US\$ 18,5 mi

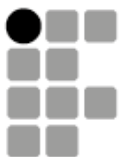




Therac-25 – 1985/1987

- Dispositivo de terapia por radiação sobre células cancerosas
- Libera doses letais de radiação em vários consultórios médicos
- Condição de disputa no SO
- 5 mortes, várias pessoas feridas





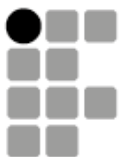
Divisão de pontos flutuantes nos processadores Pentium da Intel – 1993

- Erro em divisões dentro de uma faixa de números (erro ~0,006% no arredondamento)
- 3 a 5 milhões de peças com defeito
- *Recall* para todos que quisessem trocar
- Custou à Intel US\$ 475 milhões



$$\frac{4195835}{3145727} = 1.333820449136241002$$

$$\frac{4195835}{3145727} = 1.333739068902037589$$



INSTITUTO FEDERAL

São Paulo

Campus São João da Boa Vista

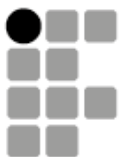
Bugs históricos

<https://encurtador.com.br/dhkzU>

Ariane 5 vôo 501 – 1996

- Levou uma década de desenvolvimento e custou 7 bilhões de dólares.
- Foguete com código reutilizado do Ariane 4 (outro hardware);
- *Overflow* de inteiro: conversão de *float* de 64-bits para inteiro 16-bits com sinal;
- O processador primário do foguete sobrecarregou os motores que se desintegraram em 40 segundos;
- Não tripulado (sem vítimas); prejuízo de US\$ 370 milhões





Bug do milênio (Y2K) – 2000

- Datas com apenas 2 dígitos para o ano
- Uma das maiores histerias da história
- Ao virar o ano 2000, a preocupação era que contasse como 1900
- Entre US\$ 300 e US\$ 500 bi no mundo todo



Por que acontecem?

- ❖ Falta de testes?
- ❖ Projetos mal feitos?
- ❖ Hardware inadequado?
- ❖ Falta de controle de qualidade?
- ❖ O cliente não sabe o que quer?
- ❖ Os desenvolvedores não entendem o que o usuário quer?





Como o cliente explicou...



Como o líder de projeto entendeu...



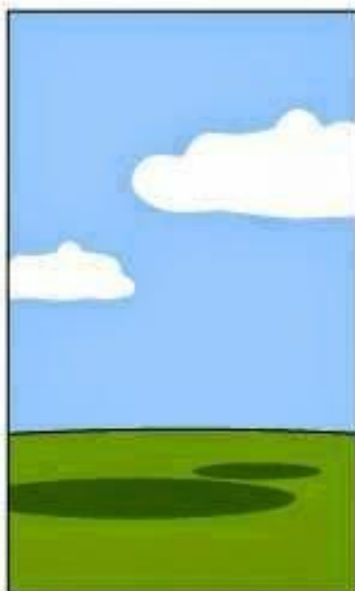
Como o analista projetou...



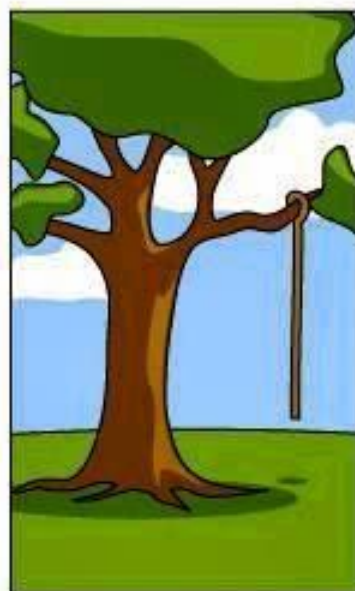
Como o programador construiu...



Como o consultor de negócios descreveu...



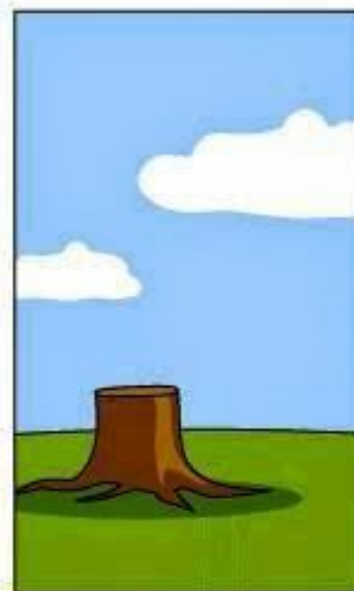
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



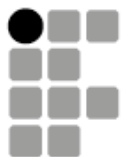
Como foi mantido...



O que o cliente realmente queria.

O que é Engenharia de Software?





engenharia

en·ge·nha·ri·a

sf

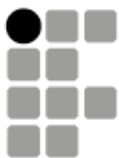
1 Arte de aplicar os conhecimentos científicos à invenção, aperfeiçoamento ou utilização da técnica industrial em todas as suas determinações.

2 **ENG** Ciência ou arte de construções civis, militares e navais.

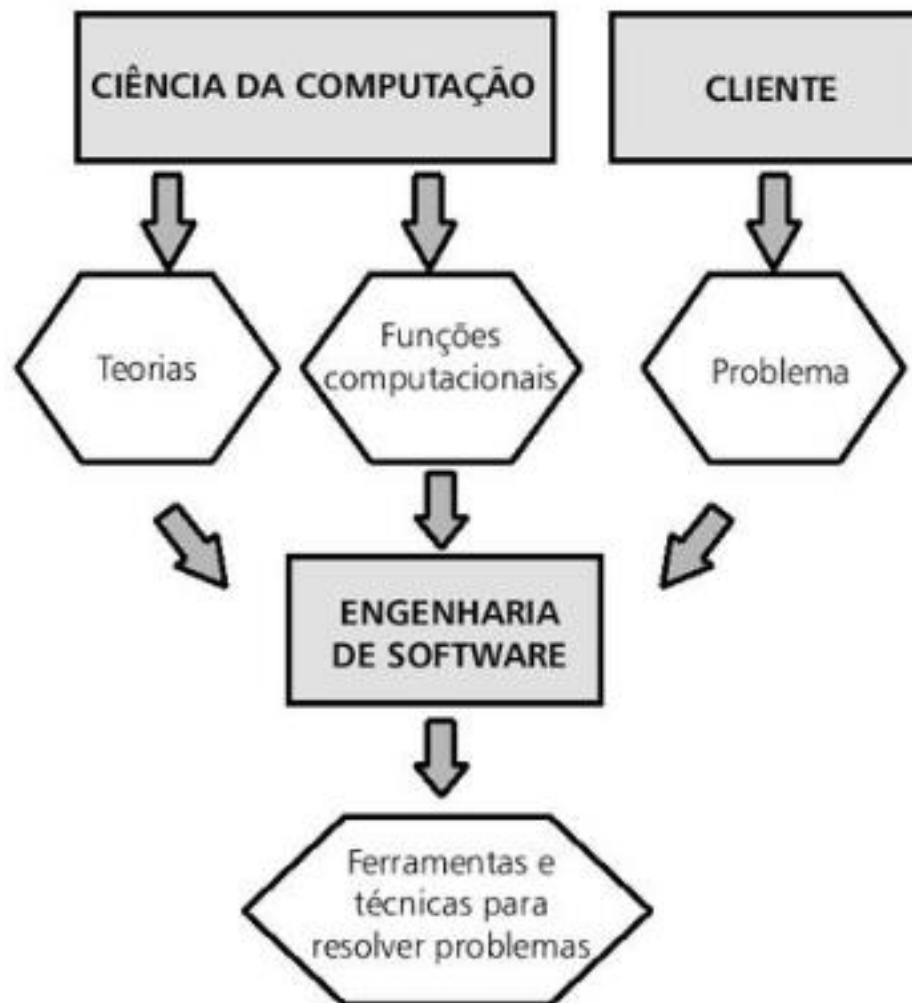
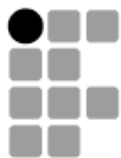
3 A classe dos engenheiros.

4 **MIL** Corpo do Exército que compreende unidades de engenheiros e sapadores.

5 Concepção e execução de algo por alguém dotado de talento, engenho e habilidade: *Alguns estilistas apresentaram trajes de uma insólita engenharia.*



“A aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software”



- Se preocupa com os aspectos da produção de software, desde sua concepção inicial até sua operação e manutenção
- Foca em questões práticas de desenvolver e entregar software útil
- Busca o desenvolver softwares de alta qualidade dentro de custos adequados
- Usar métodos e técnicas é mais barato a longo prazo. Para a maioria dos sistemas, a maior parte do custo é mudar o software depois que ele começa a ser utilizado.
- Custos da Engenharia de Software são de aproximadamente 60% relativos ao desenvolvimento e 40% aos testes

Camadas da Engenharia de Software

Ciência fundamentada na qualidade, que estuda técnicas, ferramentas, procedimentos e paradigmas no desenvolvimento de software.



- **Técnicas ou métodos:** procedimento formal para produzir um resultado.

Por exemplo a resolução de uma expressão matemática $\{[2 + (5 + 4) : 3 - \sqrt{4 + 9}] : 4\}^2$: símbolos (parênteses, colchetes, chaves) e operações (radiciação/potenciação, divisão/multiplicação, soma/subtração).

Seguir essas passos é imprescindível para que o resultado seja correto, no entanto, não importa se os cálculos estão sendo realizados a lápis ou a caneta.

- **Ferramentas:** instrumento ou sistema automatizado utilizado para realizar uma tarefa da melhor maneira (mais preciso, eficiente ou produtivo)

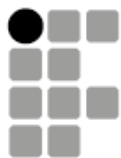
A escrita de um documento pode ser realizada de forma manuscrita ou digital. A utilização de um computador torna a tarefa mais rápida e, além disso, o texto fica mais fácil para leitura do que de forma manuscrita.

➤ **Procedimentos:** combinação de ferramentas e técnicas que, em harmonia, produzem um resultado específico.

Para fazer um bolo, por exemplo, existe uma sequência correta (técnica) para os ingredientes. No preparo, a utilização de uma batedeira (ferramenta) otimiza o tempo e mistura melhor os ingredientes.

➤ **Paradigmas:** Abordagem ou filosofia em particular para a resolver algo. É um estilo.

Em programação, por exemplo, temos o paradigma orientado a objetos (Java) e o procedural (C).



Atributos de um Software

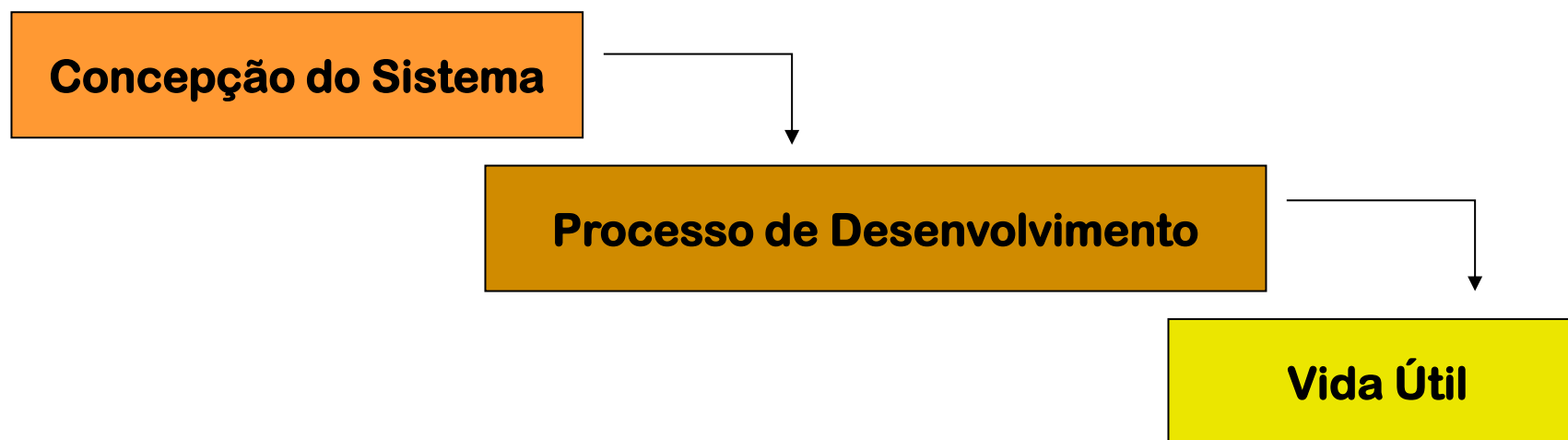
Característica do produto	Descrição
Facilidade de manutenção	O software deve ser escrito de modo que possa evoluir para atender às necessidades de mudança dos clientes. É um atributo fundamental, pois a mudança de software é uma consequência inevitável de um ambiente de negócios em constante mutação.
Confiança	O nível de confiança do software tem uma série de características, incluindo confiabilidade, proteção e segurança. Um software confiável não deve causar danos físicos ou econômicos no caso de falha no sistema.
Eficiência	O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto, a eficiência inclui tempo de resposta, tempo de processamento, utilização de memória etc.
Usabilidade	O software deve ser usável, sem esforço excessivo, pelo tipo de usuário para o qual ele foi projetado. Isso significa que ele deve apresentar uma interface com o usuário e documentação adequadas.

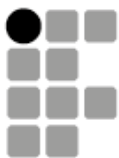
Ciclo de vida de desenvolvimento de um software



Ciclo de Vida do desenvolvimento de um software

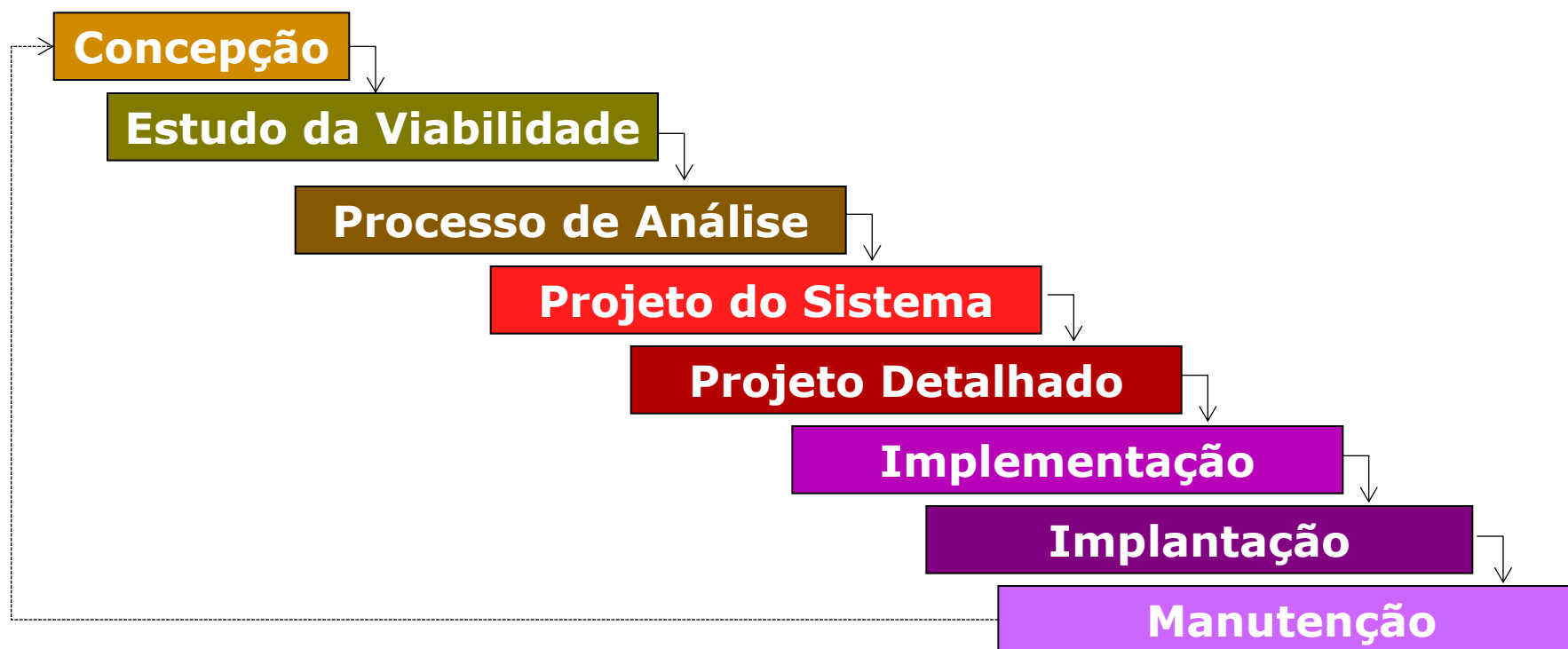
O processo de existência de um produto de software, chamado CICLO DE VIDA, passa por três estágios que são bastante distintos:

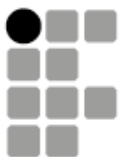




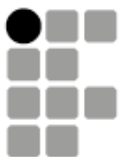
Análise e Projeto Estruturado de Sistemas

- Filosofia de trabalho metódica, gradual e disciplinada.
- Para facilitar o trabalho de desenvolvimento de um sistema, o analista deve ter em mente as seguintes fases que devem ser seguidas:

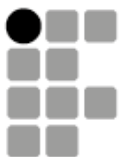




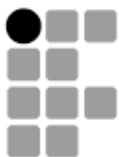
- Embrião do sistema
- Apresentar soluções alternativas
- O analista deve apresentar quais os problemas o sistema irá resolver:
 - Para definir o problema e apresentar soluções alternativas, normalmente é concedido um prazo de 2 a 3 dias, dependendo do tipo e complexidade do sistema proposto
 - Em sistemas muito complexo, concede um prazo maior, nunca ultrapassando uma semana
- A partir da proposta, a empresa define uma posição positiva ou negativa referente ao sistema



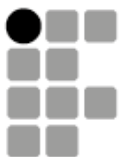
- Mostra a relação direta entre o custo e benefício do sistema proposto
 - Deve-se definir uma **estimativa** de preços (não muito precisa)
 - Algumas decisões a serem tomadas: Será necessário trocar equipamentos, contratar mais pessoas?
 - Em alguns casos, a compra de um sistema pronto pode ser a melhor solução
- Consiste em vender a ideia do sistema
- Deve-se destacar o diferencial competitivo do sistema, a melhoria para seus clientes, melhoria de produtos e serviços e aumento de lucro
- O estudo liga o **analista** → **usuário** → **administração**
- As vezes, o sistema pode “morrer” nesta fase. **Por quê?**



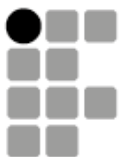
-
- Consiste em ouvir o usuário e administração para levantamento de dados e fatos para
 - Identificar as necessidades dos usuários,
 - Realizar a análise de requisitos (O quê o sistema deverá fazer) e
 - Identificar dos dados que possam gerar informações para tomada de decisão.
 - Transformação de necessidades em especificações, cujo relatório de análise deve conter:
 - a definição dos principais objetivos
 - descrição de funcionamento
 - definição do fluxo de dados
 - descrição dos arquivos
 - definição dos responsáveis pela inserção de dados no sistema
 - processamento de dados e prazo de apresentação dos dados
 - Principal Objetivo: Construção de um modelo lógico do sistema



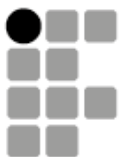
-
- Transformação do modelo lógico para o modelo físico
 - Diagrama de fluxo de dados de alto nível do sistema, Utilizando a Linguagem de Modelagem Unificada (UML)
 - Bancos de dados e arquivos
 - Definição dos relatórios a serem emitidos
 - Definição dos equipamentos a serem utilizados



-
- Consultar o usuário e gerência quais relatórios específicos gostariam de ter, ordem de entrada e saída das informações. Deixar bem claro a todos os envolvidos **como** o sistema resolverá os problemas tratados por ele.
 - Detalhar o fluxo de dados
 - Criar layouts dos relatórios
 - Definir especificações dos programas
 - Revisão do custo
 - Documentar todo sistema em desenvolvimento
 - Definir como será realizado o treinamento
 - Apresentar ao usuário como será o sistema se possível através de protótipos



- Construção física do sistema
- Codificação do sistema → Transformar os modelos físicos em códigos-fonte
- Tratar da segurança de dados
- Todos procedimentos devem ser documentados (Manual do sistema)
- Criação da versão de testes – Alfa – Beta
- Considerar a operação simultânea com o sistema antigo



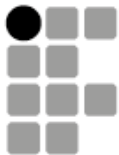
INSTITUTO FEDERAL

São Paulo

Campus São João da Boa Vista

Implantação

- Marco fundamental do ciclo de vida de um sistema
- Deve ser planejada e articulada;
- Deve-se realizar:
 - Teste geral do sistema com dados reais – Trabalho paralelo
 - Treinamento de usuários
 - Acompanhamento dos procedimentos
 - Controle e análise dos resultados processados



- Correção de possíveis erros
- Melhorias e novas implementações
- Manter o sistema atualizado de acordo com as necessidades da empresa

Esforço de Desenvolvimento por Fases (Custos % = Recursos Humanos e Financeiros)

