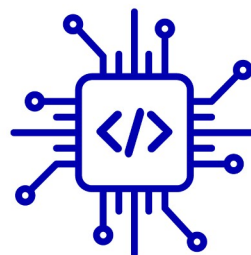


**INSTITUTO
FEDERAL**

Rio Grande
do Sul

Campus
Erechim



Tecnologia em

**ANÁLISE E
DESENVOLVIMENTO
DE SISTEMAS**

Engenharia de Software I

Prof. Dário Lissandro Beutler

E-mail: dario.beutler@erechim.ifrs.edu.br

Objetivo da aula

- Entender o conceito de Engenharia de Software (ES) e mostrar a importância da mesma para o curso de Sistemas de Informação

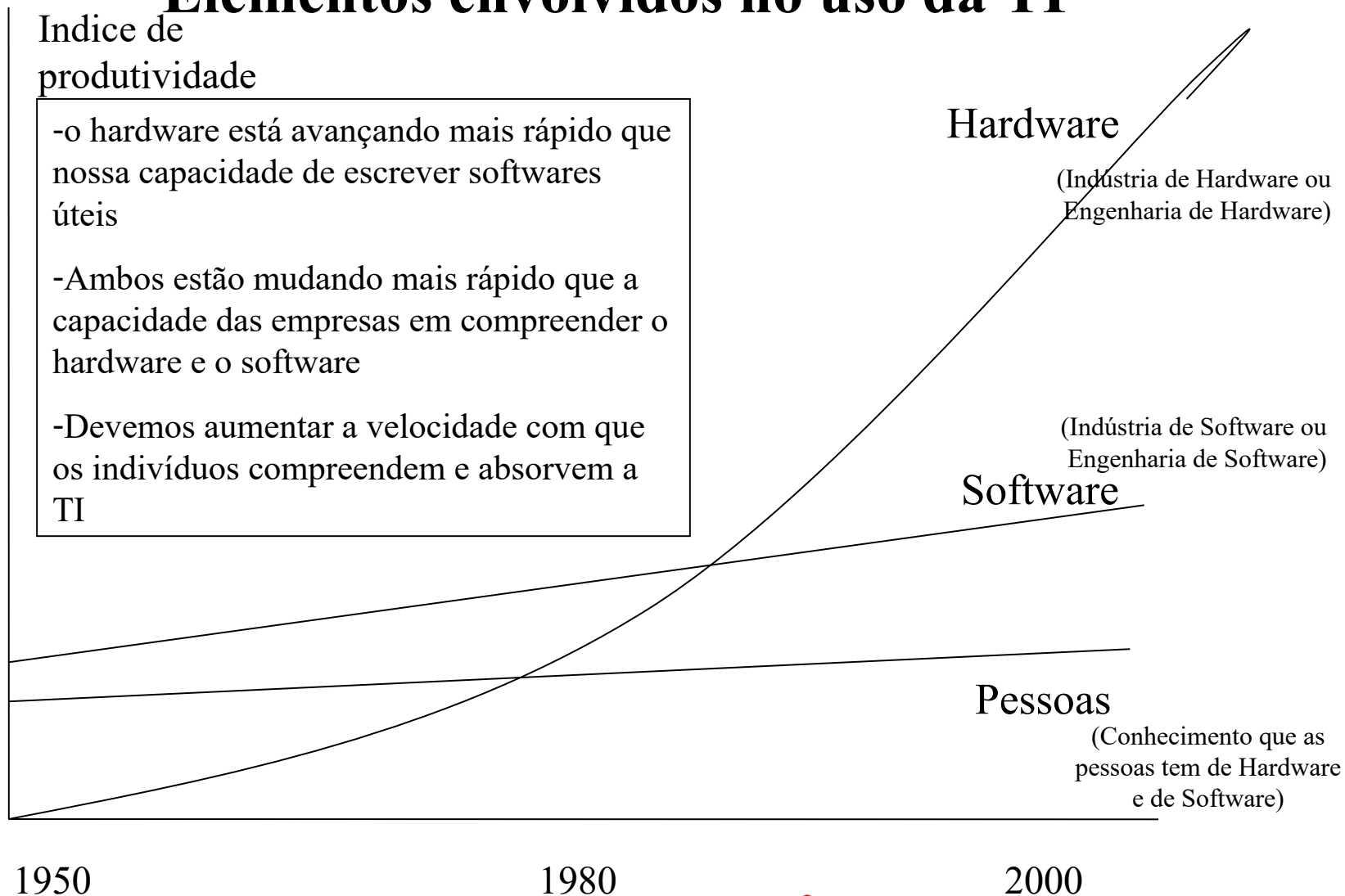
Programa da aula

- Apresentação do professor e dos alunos
- Apresentação do plano de curso
- Questões gerais
- Importância da disciplina de ES no curso
- Software
- Conceito de ES



IMPORTÂNCIA DA ES NO CURSO

Elementos envolvidos no uso da TI



LAUDON, Kenneth C., Jane Price Laudon. **Sistemas de Informação, com Internet.** LTC, 1999.



INSTITUTO FEDERAL
Rio Grande do Sul

Campus
Erechim

IMPORTÂNCIA DA ES NO CURSO

- Desafio para os profissionais de Sist. De Informação
 - Aproveitar mais os recursos de hardware desenvolvendo novos softwares
 - Fazer a indústria de software evoluir e se aproximar mais da indústria de hardware
 - Os profissionais de sistemas de informação (nós, engenheiros de software!) devem aumentar cada vez mais os seus conhecimentos das técnicas e métodos (tecnologia) da Engenharia de Software.

Uma possibilidade é tornar a escrita de aplicações de software tão simples, como, digamos, usar caneta e tinta para escrever um ensaio.

SOFTWARE

■ O que é ?

É o produto que os engenheiros de software projetam e constroem.

■ Porque é importante ?

Porque afeta praticamente todos os aspectos de nossa vida.

■ Quais são os passos ?

Assim como se constrói qualquer produto bem-sucedido.
Aplicando um processo que leva a um resultado de alta qualidade,
que satisfaz as necessidades das pessoas que vão usar o produto.
Você aplica uma abordagem de ES.

SOFTWARE

■ Qual é o produto ?

Do ponto de vista do engenheiro de software é um conjunto de programas, documentos e dados que compõem um software de computador. Do ponto de vista do usuário o produto é a informação resultante, que torna melhor o seu mundo.

■ Como garanto que fiz corretamente ?

Aplicando as idéias da ES.



SOFTWARE

Perguntas ainda sem respostas:

- 1) Por que leva tanto tempo para concluir um software?
- 2) Porque os custos de desenvolvimento são tão altos ?
- 3) Porque não podemos achar todos os erros antes de entregar o software aos clientes?
- 4) Por que continuamos a ter dificuldade em avaliar o progresso enquanto o software é desenvolvido?

Essas preocupações que levam à adoção da prática da engenharia de software.



SOFTWARE

Diferenças entre o software e outros produtos

- o hardware por exemplo é construído por um processo que evolui para um produto físico já o software gera um sistema lógico e não físico. Logo temos que ter claro que produzir software é diferente do que produzir hardware, vejamos algumas características:

1) O software é elaborado(desenvolvido) e não manufaturado.

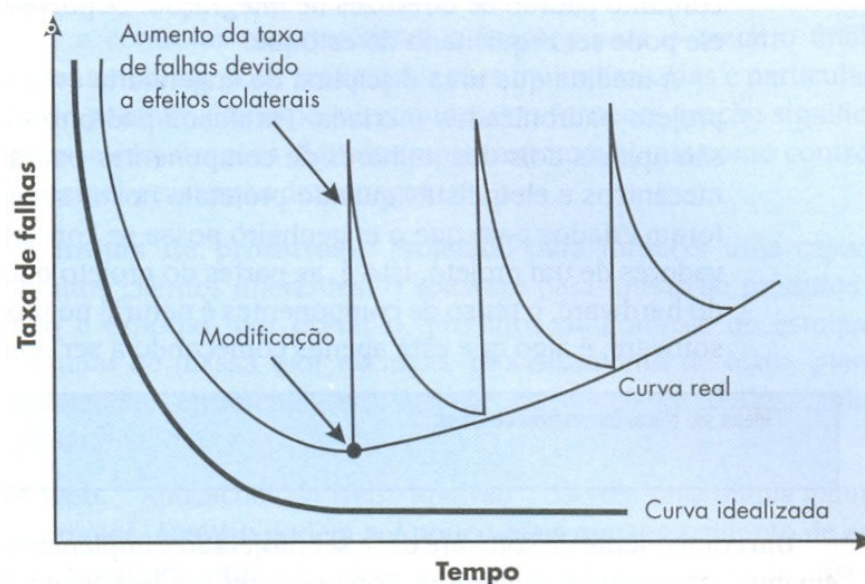
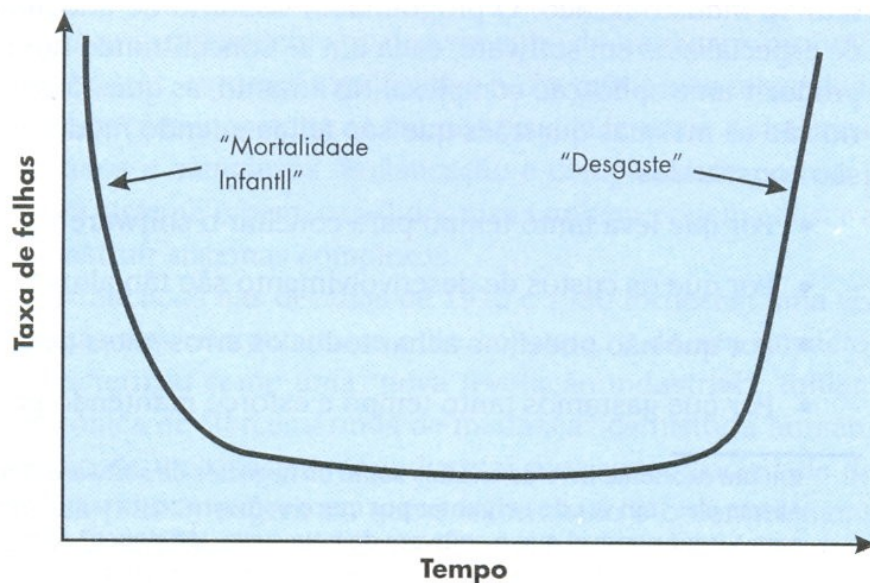
- Embora se tenha algumas semelhanças, as duas atividades são fundamentalmente diferentes.
- Em ambas a qualidade é obtida por um bom projeto, mas a fase de fabricação do hardware pode gerar problemas de qualidade, que são inexistentes para o software
- Ambas atividades dependem de pessoas, mas a relação entre essas pessoas é totalmente diferente



SOFTWARE

Diferenças entre o software e outros produtos

2) O software não se desgasta, mas se deteriora



- outro aspecto diferente é que quando o hardware se desgasta se substitui por uma peça sobressalente e no software não há peça sobressalente. Isso implica que a **manutenção de software é mais complexa que a manutenção de hardware**

SOFTWARE

Diferenças entre o software e outros produtos

3) Apesar da indústria se mover para o desenvolvimento baseado em componentes, ainda a maior parte do software é construído sob encomenda (ad-hoc).

- Na engenharia mecânica e elétrica já existe uma coleção de componentes padronizados, parafusos e circuitos integrados fazem parte de um catálogo que tem um código, uma função bem definida, e um padrão de integração
- os componentes reutilizáveis existem para que o engenheiro se concentre nos apenas nos elementos inovadores de um projeto
- no mundo do software é algo que está apenas começando em ampla escala



SOFTWARE x PROGRAMA

Vocabulário

■ Programa

- Expressão para designar o artefato executável inserido em um computador.
- Arquivo executável ou conjunto de arquivos
 - executáveis
 - de dados
 - de configurações

É uma parte do software
(é o executável)

■ Software

- O conjunto de elementos elaborados no processo de produção de software
 - código
 - documentação (das várias etapas)
 - arquivos diversos

Visão mais ampla



ARTEFATO DE SOFTWARE

Vocabulário

- Um “tipo” de software
- Parte de um software
- Tipos
 - Aplicação (completo, executável)
 - Componente (para produzir aplicações)
 - Framework (para produzir aplicações)



CONCEITO DETALHADO DE ES

- “É a criação e utilização de sólidos princípios de engenharia a fim de obter software de maneira econômica, que seja confiável e que trabalhe eficientemente em máquinas reais” (PRESSMAN, 02).
- 1) Quais são os sólidos princípios de engenharia ?
 - 2) Como construímos software de forma econômica e confiável ?
 - 3) Como criamos programa que trabalhem eficientemente em não apenas uma mas em várias máquinas reais?

CONCEITO DETALHADO DE ES

- “É a aplicação de uma abordagem sistemática, disciplinada e quantificável, para o desenvolvimento, operação e manutenção de software; isto é, a aplicação de engenharia ao software (IEEE, 93).



CONCEITO GERAL DE ES

- Sub-área de Ciência da Computação
- Busca meios sistemáticos de desenvolvimento de software
- Visa a maximização da qualidade e da produtividade na atividade de desenvolvimento de software

Qualidade – que um software apresente um conjunto de características desejáveis (QUAIS ???)

Produtividade – minimização do tempo de desenvolvimento de um software



CONCEITO GERAL DE ES

- “A ES é voltada para o desenvolvimento de **aplicações de médio e grande porte**, abrange **não apenas os aspectos técnicos** da construção de sistemas de software, mas também **aspectos gerenciais**, tais como coordenação de times de programadores, cronograma e financiamento.”

Fonte: www.webopedia.com



ENGENHARIA E PROGRAMAÇÃO

■ Programação

- ☐ Projeto pequeno
- ☐ Você
- ☐ Construir o que você mesmo quer
- ☐ Um produto
- ☐ Poucas modificações, feitas em sequência
- ☐ Tempo de vida curto
- ☐ Barato
- ☐ Pouco impacto

■ Engenharia

- ☐ Projeto Grande
- ☐ Times
- ☐ Construir o que os clientes querem
- ☐ Família de produtos
- ☐ Várias modificações, feitas em paralelo
- ☐ Tempo de vida longo
- ☐ Caro
- ☐ Grande Impacto



WINDOWS VISTA(Longhorn)

- 5000 desenvolvedores (sem incluir pessoal não-técnico);
- 40 milhões de linhas de código;
- 16 milhões de linhas de código somente nos últimos 3 anos;
- Versões compiláveis todos os dias;
- Testes de regressão;
- Intervalo de 3 dias para uma mudança submetida aparecer no executável;
- ~1.7 pessoas testando para cada programador;
- Precisa ter compatibilidade com versões anteriores;
- Precisa ser instalado em milhares de configurações diferentes;



WINDOWS VISTA(Longhorn)

- Como organizar as atividades tal que 5000 pessoas possam trabalhar juntas ao mesmo tempo;
 - **Processo de software;**
 - **Ferramentas de Gerência de Configuração;**
- Como testar tanto código? Para tantas plataformas diferentes?
 - **Testes de regressão?**
- Como projetar um sistema com 40 milhões de linhas de código? Como garantir a integridade deste projeto?
 - **Arquitetura / Projeto de software;**

ATIVIDADES DA ES

- Desenvolvimento de uma nova aplicação
- Modificação de uma aplicação
- Compreensão de artefatos de software
- Produção de subsídios a serem usados no desenvolvimento de aplicações

