Engenharia de Software Introdução

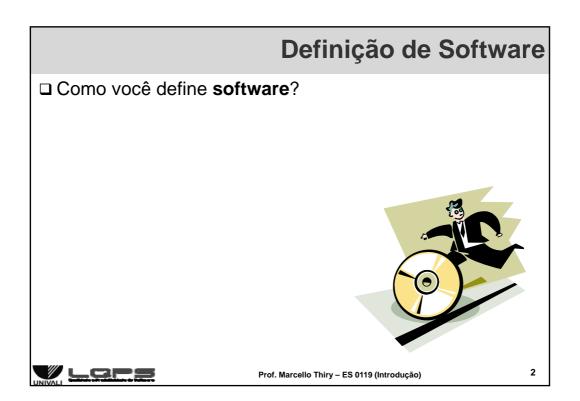
Marcello Thiry

thiry@univali.br marcello.thiry@gmail.com



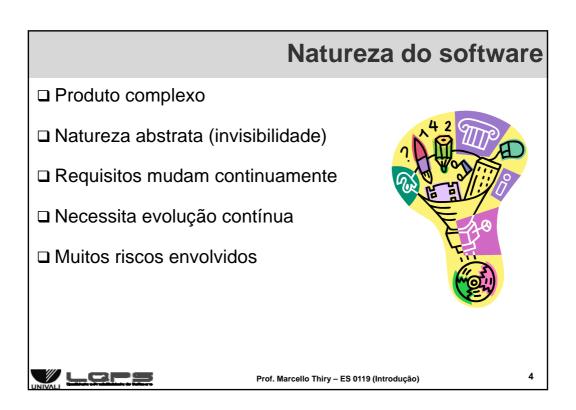
LQPS

http://www.univali.br/lqps



Definição de Software Como você define software? Um conjunto de programas, constituído por blocos de código e pela documentação de apoio que é produzida Aplicação de princípios por meio de métodos e ferramentas

Prof. Marcello Thiry - ES 0119 (Introdução)



Crise do software: 70's Crise do software: 70's Disseminação do uso de computadores Programação individual para programação em grupo Problemas com a identificação de requisitos Difícil interação entre engenheiros e clientes Falta de sistemáticas conhecidas para desenvolvimento Baixa produtividade das equipes





Alguns dados importantes

- ☐ Apenas 34%* dos projetos são completados no prazo, orçamento e atendem todas as funcionalidades definidas (Standish Group, 2004)
- □ 15% dos projetos acabam sendo cancelados antes do término (Standish Group, 2004)
- Mais de 50% dos projetos estão fora de prazo, custo ou não atendem funcionalidades críticas (Standish Group, 2004)
- Mais de 50% dos projetos custam 43% em relação a estimativa inicial (Standish Group, 2004)
 - * Mesmo assim, houve uma melhora de mais de 100% em 10 anos



Prof. Marcello Thiry – ES 0119 (Introdução)

Engenharia de Software

- □ Termo adotado no final da década de 60
- □ Objetivo foi buscar sistemáticas já utilizadas para:
 - □ Levar os **custos** e **prazos** a níveis aceitáveis
 - □ Gerenciar o processo de desenvolvimento
 - □ Permitir o trabalho em grupo
 - □ Aumentar **qualidade** e **produtividade**





Prof. Marcello Thiry - ES 0119 (Introdução)

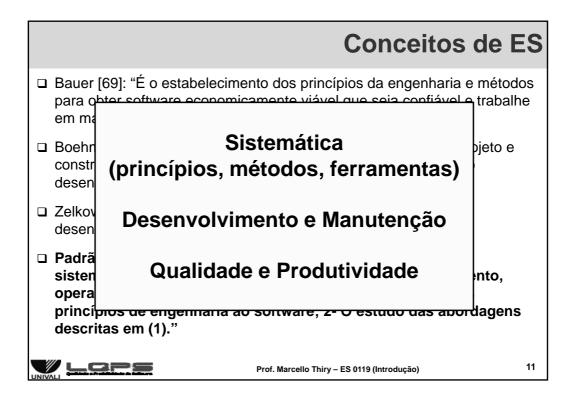
9

Conceitos de ES

- ☐ Bauer [69]: "É o estabelecimento dos princípios da engenharia e métodos para obter software economicamente viável que seja confiável e trabalhe em máquinas reais"
- □ Boehm [76]: "Aplicação prática do conhecimento científico ao projeto e construção de programas e da documentação necessária para o desenvolvimento, operação e manutenção dos mesmos."
- ☐ Zelkowitz [83]: "é o estudo dos princípios e metodologias para o desenvolvimento e manutenção de sistemas."
- □ Padrão IEEE 610.12 [90]: "1- Aplicação de uma abordagem sistemática, disciplinada e quantitativa para o desenvolvimento, operação e manutenção do software; isto é, a aplicação dos princípios de engenharia ao software; 2- O estudo das abordagens descritas em (1)."



Prof. Marcello Thiry - ES 0119 (Introdução)



Terminologia adotada

- Princípios: fatos aceitos que são utilizados como premissa e justificativa para ações (abstração, modularidade, ...)
- ☐ <u>Técnicas e Métodos</u>: regras para a execução de uma atividade específica (técnicas de teste, técnicas para elicitar requisitos, ...)
- □ <u>Metodologia</u>: conjunto de técnicas e métodos que são aplicados de forma sistemática para atingir objetivos definidos
- ☐ <u>Ferramentas</u>: oferecem suporte computacional (pode ser parcial) para a implantação de metodologias



Prof. Marcello Thiry - ES 0119 (Introdução)

Qualidade □ Produto de software: entidade de software disponível para liberação do usuário (ISO 13596) □ Qualidade: totalidade das características de uma entidade, que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas (ISO 13596) □ Qualidade de software: totalidade das características de um produto de software que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas* (ISO 13596) □ Visão simplificada → Capacidade de atender aos requisitos estabelecidos *Necessidades implícitas são as expectativas dos stakeholders

Prof. Marcello Thiry - ES 0119 (Introdução)

13

LGPS

Qualidade de Software □ Qualidades de Processo ☐ A qualidade de um produto de software é fortemente dependente da qualidade do processo pelo qual ele é construído e mantido □ O processo de software pode ser definido, gerenciado, medido e melhorado □ Qualidades de Produto □ Qualidades internas: atributos que são geralmente acrescentados pelo desenvolvedor (base para atender as qualidades externas) □ Qualidades externas: visíveis aos usuários, devem estar explicitamente definidas na especificação de requisitos do projeto LOPS 14 Prof. Marcello Thiry - ES 0119 (Introdução)

Algumas qualidades - 1

- □ <u>Eficiência</u> (processo): o software é produzido dentro do prazo e orçamento determinado
- ☐ <u>Eficiência</u> (produto): capacidade do software apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob condições especificadas
- □ <u>Produtividade</u> (processo): capacidade da equipe em produzir o software, atendendo as qualidades especificadas
- □ <u>Confiabilidade</u> (produto): o software funciona como esperado (atende as necessidades do usuário)
- □ <u>Corretude</u> (produto): o software produz a saída correta (faz aquilo que deveria fazer de acordo com o que foi especificado)



Prof. Marcello Thiry - ES 0119 (Introdução)

15

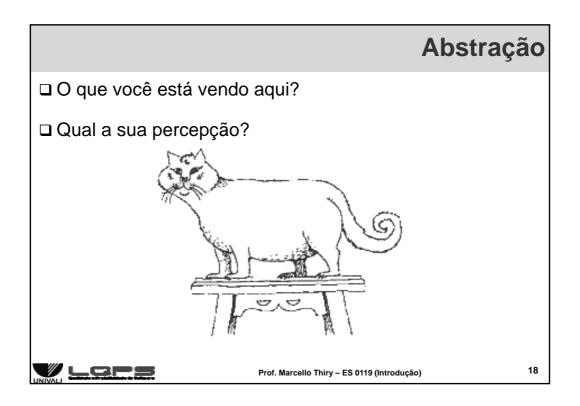
Algumas qualidades - 2

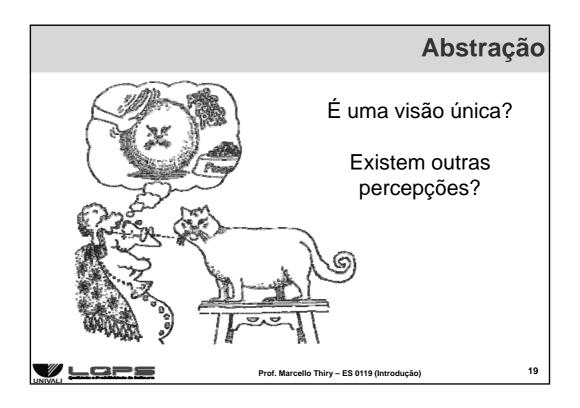
- □ Robustez (produto): quando o software se comporta "razoavelmente" sob circunstâncias não previstas na especificação dos requisitos
- □ <u>Usabilidade</u> (produto): facilidade de uso do ponto de vista do usuário
- □ <u>Portabilidade</u> (produto): o software pode ser transferido para outros computadores ou sistemas sem maiores modificações
- Reusabilidade (produto): o software (ou parte dele) pode ser usado novamente em outros projetos. Isto significa que o software é modular, que cada módulo tem uma interface bem definida e que cada módulo tem um resultado claro a partir de sua execução
- □ <u>Interoperabilidade</u> (produto): o software pode interagir com outros sistemas
- Modificabilidade (produto): o software deve ser facilmente modificado se os requisitos mudarem

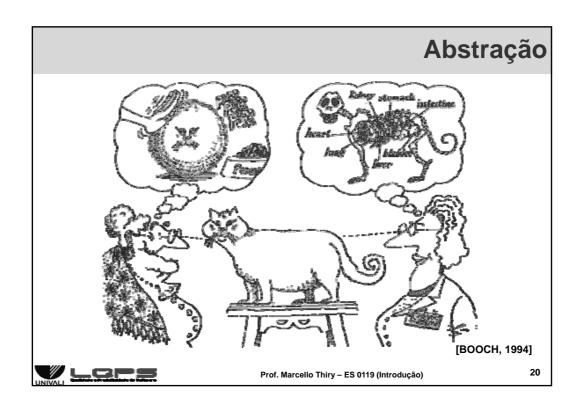


Prof. Marcello Thiry – ES 0119 (Introdução)

Rigor e formalidade Atividade criativa (inspiração) precisa de formalismo? Rigor Complemento necessário para criatividade Rigor não restringe a criatividade, mas a melhora quando incrementa a crença nos resultados criativos Formalidade: Alto grau de rigor (leis matemáticas) Permite a mecanização do processo

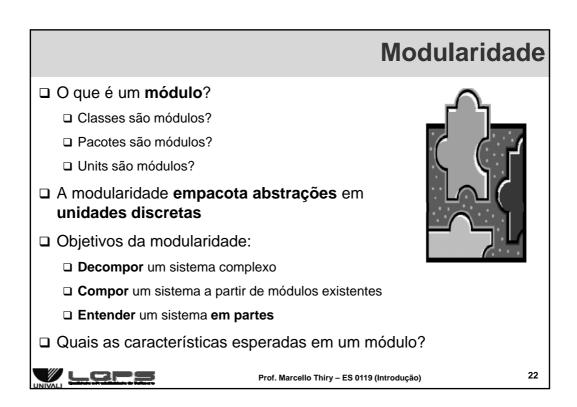






Prof. Marcello Thiry - ES 0119 (Introdução) - Página 10

Abstração Identificar aspectos importantes de um fenômeno e ignorar detalhes Técnica para lidar com a complexidade A seleção de quais aspectos são importantes depende do observador e do problema observado Criação de modelos: diagramas (módulos, classes, ...), protótipos, equações, etc

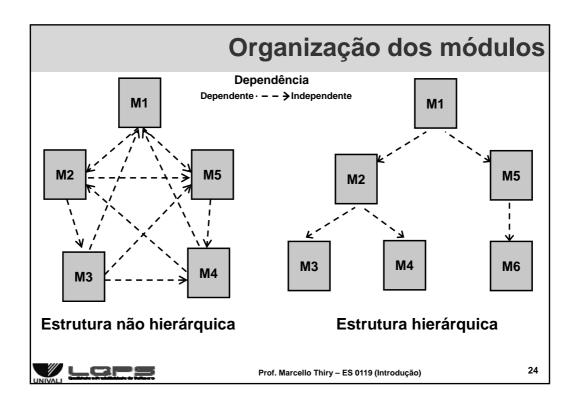


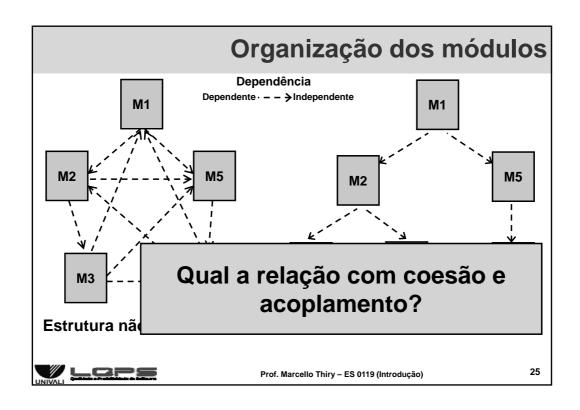
Coesão e Acoplamento

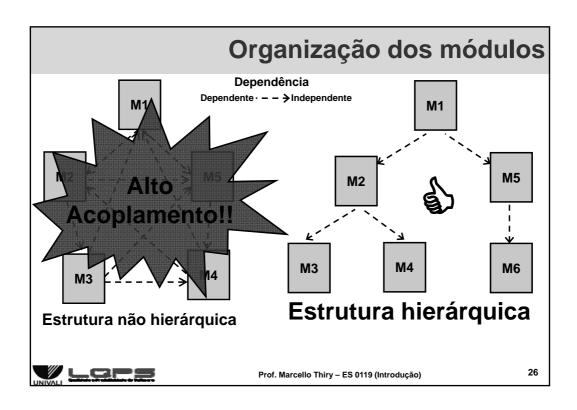
- Medidas de qualidade típicas aplicadas a qualquer tipo de módulo: sub-sistemas, classes, rotinas, etc
- □ Coesão: quando um módulo possui um único objetivo, onde todas as partes do módulo estão alinhadas com este objetivo (medida interna)
- □ Acoplamento: grau de dependência entre os módulos (medida externa)
- □ É desejável que os módulos tenham alta coesão com baixo acoplamento



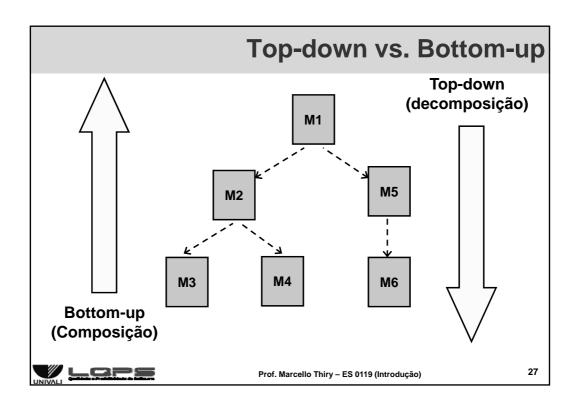
Prof. Marcello Thiry - ES 0119 (Introdução)

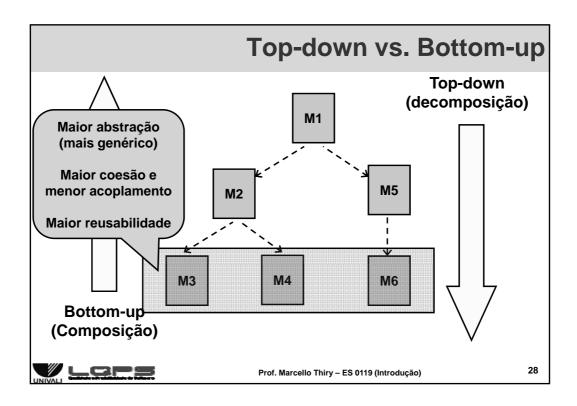




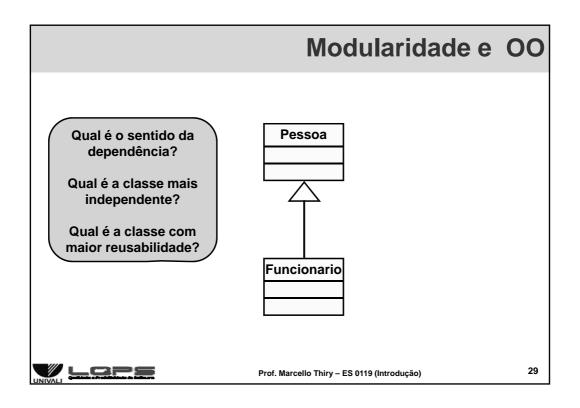


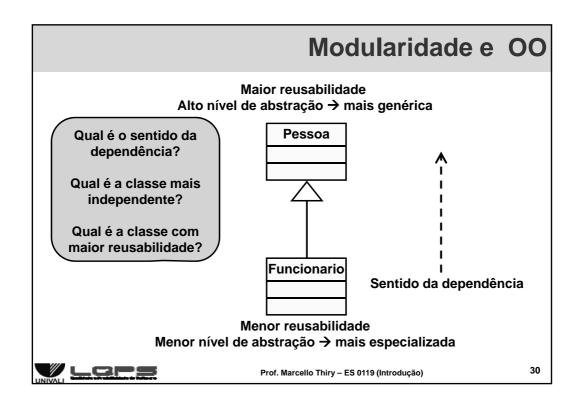
Prof. Marcello Thiry - ES 0119 (Introdução) - Página 13





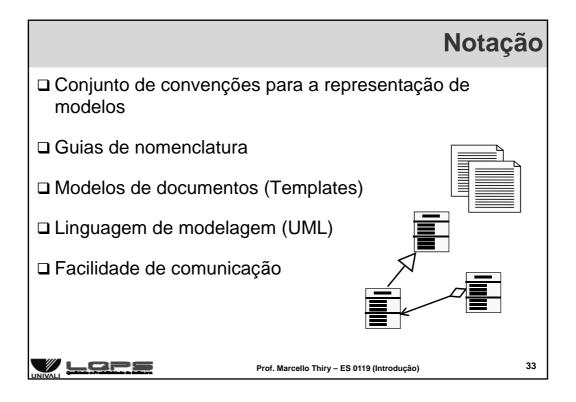
Prof. Marcello Thiry - ES 0119 (Introdução) - Página 14

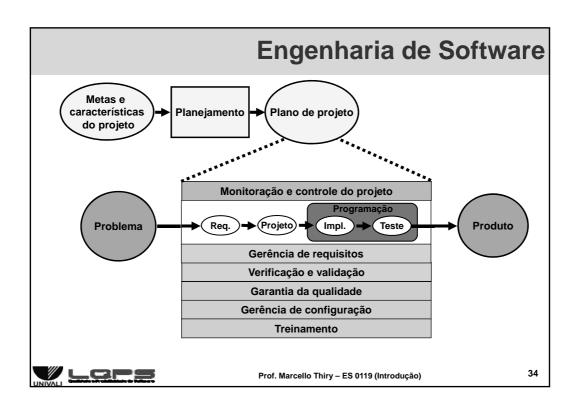




Antecipação a mudanças Software sofre modificações continuamente Importante: prever possíveis modificações (como?) Capacidade de manutenção: maior qualidade Deve fazer parte do projeto Princípio que permite o sistema evoluir Relacionamento com a reusabilidade Ferramentas para gerenciar versões e revisões

Encapsulamento □ Um módulo pode ser pensado como um "iceberg", onde apenas uma pequena parte está visível (interface) □ Separação entre interface e implementação □ Interface: o que o módulo está exportando (módulo visto como caixa preta); assinaturas das operações □ Implementação: métodos de construção da interface; uma interface pode ter diferentes implementações □ Independência: facilita a manutenção do módulo; a dependência é da interface, não da implementação





Prof. Marcello Thiry - ES 0119 (Introdução) - Página 17

Tarefas da Engenharia de Software □ Análise do problema e □ Documentação identificação de requisitos ■ Manutenção □ Projeto (design) □ Controle de qualidade □ Codificação □ Treinamento ☐ Testes e integração do código □ Estimativas de recursos ☐ Instalação e entrega do □ Gerência de projetos software 35 Prof. Marcello Thiry - ES 0119 (Introdução)

