

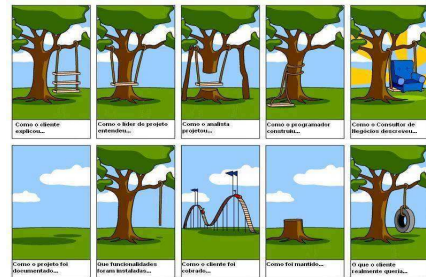
Técnicas de Construção de Programas

Prof. Marcelo Soares Pimenta
mpimenta@inf.ufrgs.br

Porto Alegre, agosto a dezembro de 2009

1
©Pimenta 2009

Qual o problema do Software?



©Pimenta 2009

Qual o problema das organizações?



3
©Pimenta 2009

Programa vs Software

Programa

Uso Pessoal
Doc pequena
Usuário é o autor
Erro é 'irrelevante'
Sem manutenção

- Programa é artefato
- Desenvolvimento é 'arte'
- Atividade Pessoal
(programming-in-the-small)

Software

Uso Comercial
Doc rica
Usuários diferenciados
Erro é grave
Muita manutenção

- Software é **produto**
- Desenvolvimento necessita de **'engenharia'**
- Construção em equipe de SW com múltiplas versões
(programming-in-the-large)

4
©Pimenta 2009

Introdução

Projeto Ariane 5



- Projeto da Agência Espacial Europeia que custou:
 - 10 anos.
 - US\$ 8 Bilhões.
- Lançamento em 4/junho/1996
- Garante supremacia europeia no espaço.

Ver em <http://www.esrin.esa.it/htdocs/esa/ariane/>

5
©Pimenta 2009

Resultado



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

CAUSA: FALHA DE SOFTWARE!

- Shut-down! Ocorrerá um *run time error* (out of range, overflow, ou outro) e ambos computadores (principal e back-up) se desligaram.
- Diagnóstico: 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.

6
©Pimenta 2009

Quais são os problemas?

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

Problema: Falta de Qualidade !!

- “Crise de Software”
 - Software :
 - Produto de baixa qualidade
 - Desenvolvimento de Software:
 - Processo com Baixa Qualidade
 - Baixos graus de satisfação (stress, falta de tempo, etc)
 - Baixa produtividade dos projetistas de software

Mudança de cultura de desenvolvimento de software

- Barateamento do HW
 - Tecnologias de integração mais avançadas (*chips*)
- Fazer Software ainda é atividade intelectual:
 - Mão de obra intensiva (\$)
 - Baseada em experiência
 - Baixos níveis de assistência e automação
- Usuários mais exigentes: qualidade e prazo
- Software mais complexo:
 - metodologias tradicionais já não servem

	Manual	Inf
CONV	V	V
Não CONV	F	V

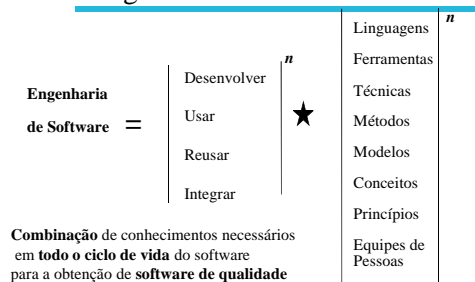
Engenharia de Software

- Dificuldade:
 - NÃO há teoria subjacente (motor elétrico: equações de tensão, potência, corrente, etc): Quais equações a seguir para SW?
- NA PRÁTICA, enfoque de Engenharia para sistematizar as atividades de:
 - Entender claramente o problema que se quer resolver
 - Desenvolver ferramentas e técnicas para resolvê-lo
 - Gerenciar equipe para resolvê-lo
- Aspectos Tecnológicos e Gerenciais

Engenharia de Software

- SW é um produto ‘diferente’
 - **Virtual**: falta de leis e propriedades físicas para SW - visibilidade, massa, volume, cor, odor, etc. - e não degrada
 - **Maleável**: pode ser modificado após pronto
- Engenharia de Software <> Programação
 - Sistematização das atividades de analisar, especificar, projetar, programar (implementar), verificar, validar, manter e gerenciar um projeto de software.
 - Software engloba programas e todos os documentos associados:
 - especificações, projetos e planos de teste
 - documentação técnica e para os usuários

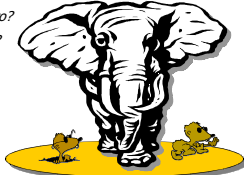
Engenharia de Software



O que é ter qualidade?

- ✓ *Satisfazer as necessidades e expectativas dos clientes?*
- ✓ *Estar conforme com as especificações de produto?*
- ✓ *Ter a melhor relação custo x benefício?*
- ✓ *Fornecer produtos adequados ao uso?*
- ✓ *Fazer mais, melhor e mais barato?*
- ✓ *Ser o mais atrativo do mercado?*
- ✓ *Etc...*

Subjetiva demais ...
Centrada no Produto ...
Centrada no Valor ...
Centrada no Processo ...
Centrada no Cliente ...



13

©Pimenta 2009

What is quality?

- Quality, simplistically, means that a product should meet its specification.
- This is problematical for software systems
 - There is a tension between customer quality requirements (efficiency, reliability, etc.) and developer quality requirements (maintainability, reusability, etc.);
 - Some quality requirements are difficult to specify in an unambiguous way;
 - Software specifications are usually incomplete and often inconsistent.

– Source:
Chapter: Quality Management, Sommerville, 1. Software Engineering, 7a ed

14

©Pimenta 2009

Algumas definições ...

ISO 9000:2000

PROCESSO: Um sistema de atividades que usa recursos para transformar entradas em saídas.

PRODUTO: O resultado de um processo.

SISTEMA: Conjunto de elementos inter-relacionados ou interconexos.

CARACTERÍSTICA: Propriedade (coisa) distinguível.

REQUISITO: Necessidade ou expectativa que é declarada, usualmente implícita ou obrigatória.

QUALIDADE: Habilidade de um conjunto de características inerentes de um produto, sistema ou processo para atender plenamente os requisitos dos clientes ou outras partes interessadas.

15

©Pimenta 2009

Características do software

CARACTERÍSTICA	SUB-CARACTERÍSTICA	PERGUNTA CHAVE
Funcionalidade Satisfaz as necessidades?	Adequação	Propõe-se a fazer o que é apropriado?
	Corretude	Faz o que foi proposto de forma correta?
	Interoperabilidade	Interage com os sistemas especificados?
	Conformidade	Está de acordo com as normas, leis, etc.?
Confiabilidade É imune a falhas?	Segurança de acesso	Evita acesso não autorizado aos dados?
	Maturidade	Com que frequência apresenta falhas?
	Tolerância a falhas	Ocorrendo falhas, como ele reage?
	Recuperabilidade	É capaz de recuperar dados em caso de falha?
Usabilidade É fácil de usar?	Compreensibilidade	É fácil entender o conceito e a aplicação?
	Aprensibilidade	É fácil aprender a usar?
	Operacionalidade	É fácil de operar e controlar?

* Características do Produto de Software (ISO/IEC 9126)

16

©Pimenta 2009

Características do software

CARACTERÍSTICA	SUB-CARACTERÍSTICA	PERGUNTA CHAVE
Eficiência É rápido e "enxuto"?	Tempo	Qual é o tempo de resposta, a velocidade de execução?
	Recursos	Quanto recurso usa? Durante quanto tempo?
Manutenibilidade É fácil de modificar?	Analísabilidade	É fácil de encontrar uma falha, quando ocorre?
	Modificabilidade	É fácil modificar e adaptar?
	Estabilidade	Há grande risco quando se faz alterações?
	Testabilidade	É fácil testar quando se faz alterações?
Portabilidade É fácil de usar em outro ambiente?	Adaptabilidade	É fácil adaptar a outros ambientes?
	Cap. para ser instalado	É fácil instalar em outros ambientes?
	Conformidade	Está de acordo com padrões de portabilidade?
	Cap. para substituir	É fácil usar para substituir outro?

* Características do Produto de Software (ISO/IEC 9126)

17

©Pimenta 2009

O que é gestão da qualidade?

ORGANIZAÇÃO: Grupo de pessoas e instalações com um arranjo ordenado de responsabilidades, autoridades e relações.

GESTÃO: Atividades coordenadas para dirigir e controlar uma organização.

GESTÃO DA QUALIDADE: Atividades coordenadas para dirigir e controlar uma organização no que diz respeito à qualidade.

OBJETIVOS DA QUALIDADE: Alguma coisa pensada, ou alvos, relacionados à qualidade.

POLÍTICA DA QUALIDADE: Totalidade das intenções e direção de uma organização relativas à qualidade, formalmente expressas pela alta direção.

SISTEMA DE GESTÃO DA QUALIDADE: Sistema para estabelecer uma política e objetivos da qualidade da qualidade, bem como os métodos para alcançar esses objetivos.

18

©Pimenta 2009

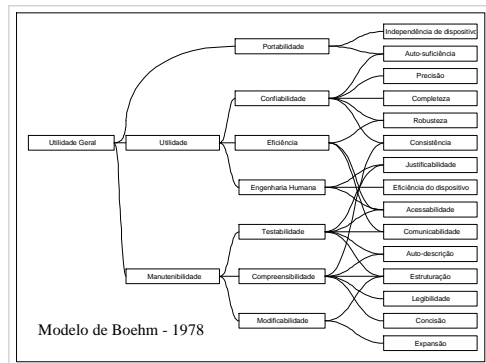
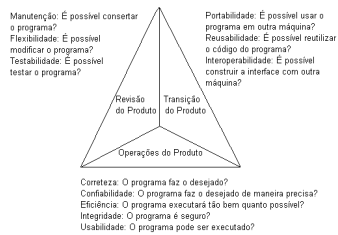
Definições de Qualidade de Software

- Qualidade relacionada à conformidade do software com os requisitos:
 - Problema: raramente os requisitos estão completos.
- Qualidade relacionada à satisfação do usuário:
 - Problema: usuários diferentes.

Modelo de McCall - 1977

- Identifica três áreas de trabalho:
 - operação;
 - revisão;
 - transição.
- Identificação de critérios em cada área de trabalho.

Modelo de McCall



CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE NBR ISO / IEC 9126

- ➡ FUNCIONALIDADE
- ➡ CONFIABILIDADE
- ➡ USABILIDADE
- ➡ EFICIÊNCIA
- ➡ MANUTENIBILIDADE
- ➡ PORTABILIDADE

CARACTERÍSTICAS DE QUALIDADE DE SOFTWARE NBR ISO / IEC 9126

1

FUNCIONALIDADE

Adequação

PARA AS TAREFAS ESPECIFICADAS

Precisão

RESULTADOS / EFEITOS CORRETOS

Interoperabilidade

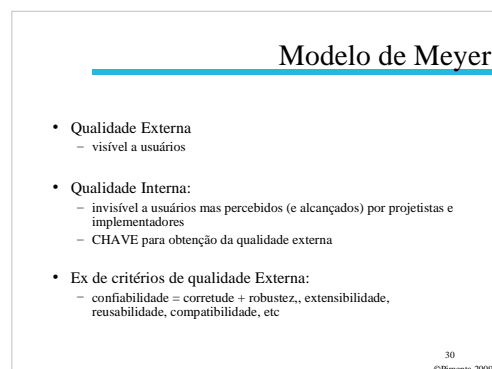
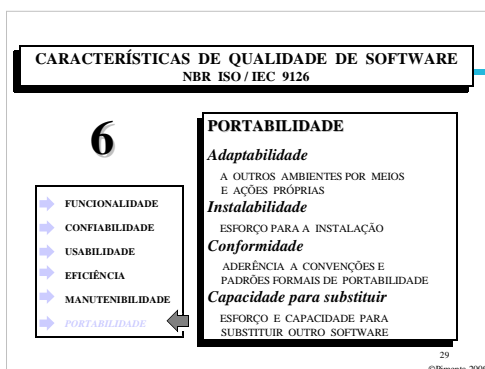
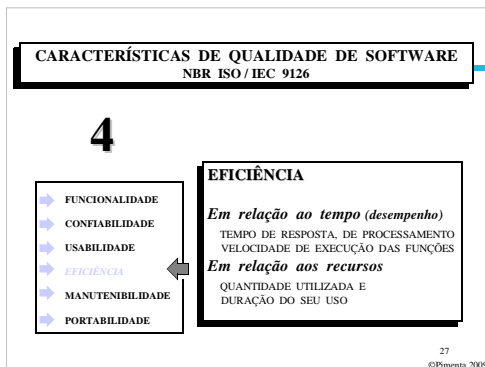
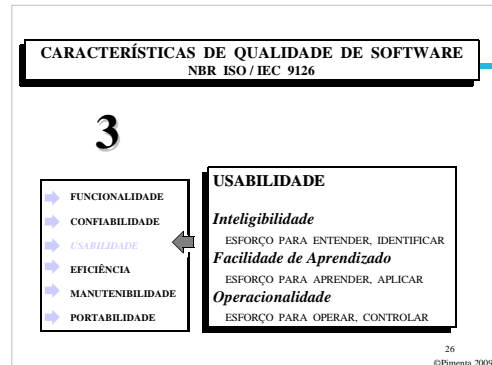
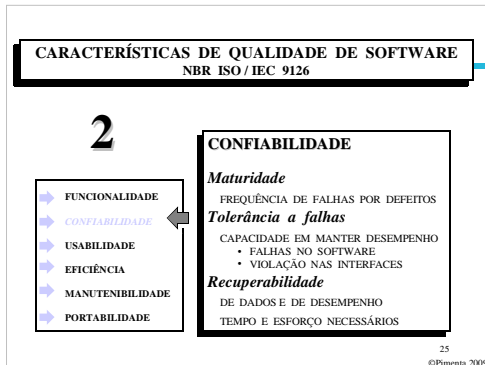
INTERAGIR COM SISTEMAS ESPECIFICADOS

Conformidade

A NORMAS, CONVENÇÕES, LEIS, DESCRIÇÕES

Segurança de acesso

EVITAR ACESSO ACIDENTAL OU DELIBERADO



Leitura Recomendada

- “Software Quality”

Cap. 1 do livro

Meyer, B. *Object-Oriented Software Construction*,
2ª edição, 1997.

PDF disponível no moodle da disciplina e/ou na HP
do professor (conferir)

Discussão sobre as leituras

Objetivo das leituras: complementar o conteúdo visto em aula; criar e/ou resolver dúvidas e sugerir discussão sobre alternativas e/ou consequências!!

Por isto, para cada leitura recomendada, sugere-se que o aluno faça
POR ESCRITO (no máx uma página por cada texto):

- f) * Um resumo de 4 ou 5 parágrafos com as idéias principais de cada texto lido;
- g) * Uma lista de dúvidas para discussão na aula seguinte;

Na aula seguinte, o professor pode solicitar sua participação oralmente;

O envio dos escritos por email é aceito até as 24h da véspera da aula

Discussão

- Discuta a relação entre corretude e reusabilidade.
- Discuta a relação entre robustez e usabilidade.
- Procure exemplos de critérios conflituosos e discuta como resolver estes conflitos no desenvolvimento do software.