Informática UFRGS

UFRGS

INFO1120 – TCP

Técnicas de Construção de Programas

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

Slides – Arquivo 2

Mudança de cultura de desenvolvimento de software

ANTES:	HOJE:
Computador: recurso caro	Computador: recurso barato
Aplicações Convencionais	Aplicações mais complexas
Usuários ingênuos	Usuários mais exigentes
Software tipicamente "batch"	Software tipicamente interativo
Pensar antes de fazer (Ciclo tradicional)	Pensar experimentando (Prototipação)
Definições (e documentação) em papel	Disponibilidade de Ferramentas ("paperless")
Ferramentas isoladas (editor, compilador, etc.)	Ferramentas integradas em ambientes
Métodos indutivos (prática primeiro)	Métodos dedutivos (modelos primeiro)
Desenvolver software é arte	Desenvolver software envolve princípios científicos

INVARIANTE: Atividade intensivamente intelectual e ainda fortemente baseada em pessoas (não automatizada)

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*)

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

Discussão sobre leitura

- Meyer, B. ´Software Quality´ - OOSC - cap. 1

- 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.

Facilidade de Uso (*ease of use*)

- Qualidade historicamente ligada a 'amigabilidade' (*user friendliness*)
- Diferentes pontos de vista de qualidade de interfaces:

  a) Performance humana satisfatória(ISO)
  - Eficácia (%) - coeficientes de erro
  - Eficiência (*t, $*) - velocidade de uso

  b) Tempo de aprendizado e de retenção

Usabilidade

- Adequação entre características (físicas/cognitivas) dos usuários e características da interação (com o sistema) para realização de tarefas
- Não é propriedade intrínseca do sistema mas do trio (usuário, sistema, tarefa)
- Expressa por alguns fatores:
  - **facilidade de aprendizado**, intuitiva e "natural"
  - **flexibilidade de interação**, multiplicidade de formas de uso
  - **robustez de interação**, acompanhamento e recuperação em situações de incidentes

Usabilidade

- Resumo: USABILIDADE É QUALIDADE DE USO
  - facilidade de aprendizado
  - facilidade de uso (operação)
  - taxa de erros minimizada
  - Satisfação dos usuários
  - adequação à tarefa

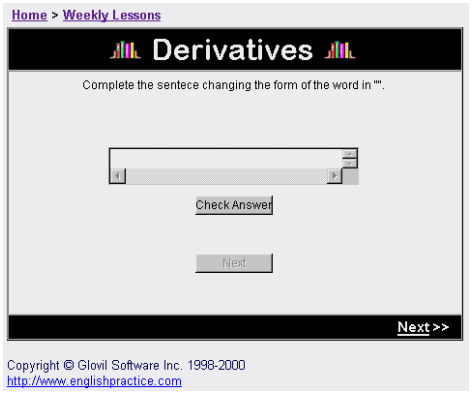
- Usabilidade é obtida por **construção**
  - Clara compreensão dos requisitos de usabilidade **durante** as etapas iniciais da concepção e não somente ao final
    - BUSCAR: Usabilidade como requisito do sistema (´built-in approach´)
    - EVITAR: Usabilidade somente como critério de avaliação (´day-after approach´)

Problema de Usabilidade

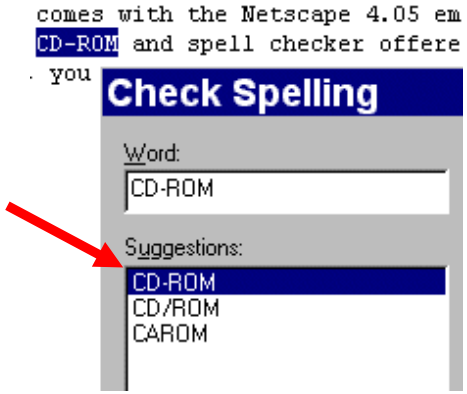
- se há dificuldades (reais ou potenciais) para determinado usuário ou (grupo de usuários) realizar uma tarefa com a interface !!!
- Graus de severidade

Tipo	*Descrição (necessidade de reparo)*
0Sem importância	***Não afeta a operação da interface***
1Cosmético	***Não há necessidade imediata de solução***
2Simples reparado)	***Problema de baixa prioridade (pode ser***
3Grave reparado)	***Problema de alta prioridade (deve ser***
4Catastrófico	***PRIORIDADE MÁXIMA no reparo***

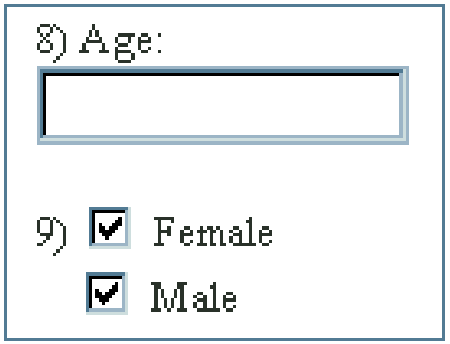
Exemplos de problemas



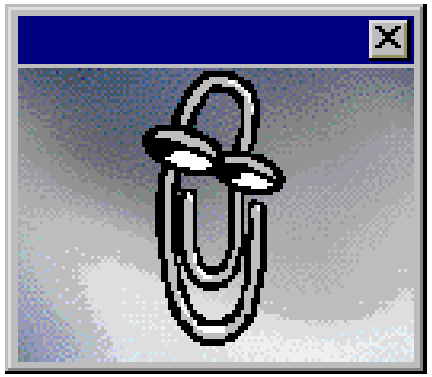
Exemplos de problemas



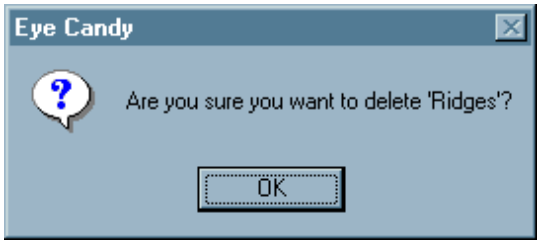
Exemplos de problemas



Exemplos de problemas

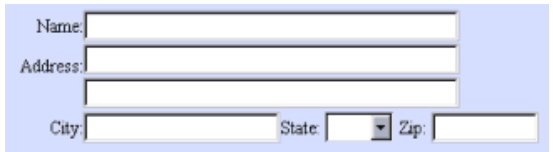


Exemplos de problemas



Exemplos de problemas

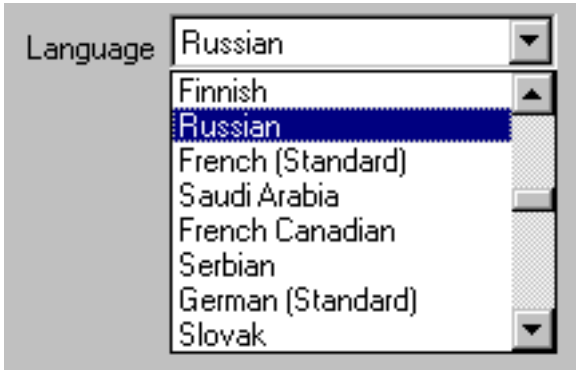
Exemplos de problemas



USA

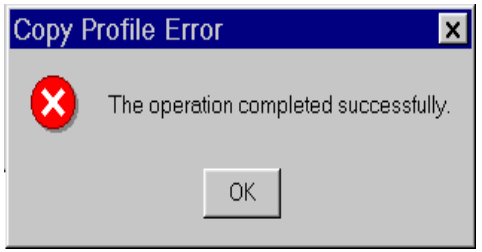
Internacional

Exemplos de problemas

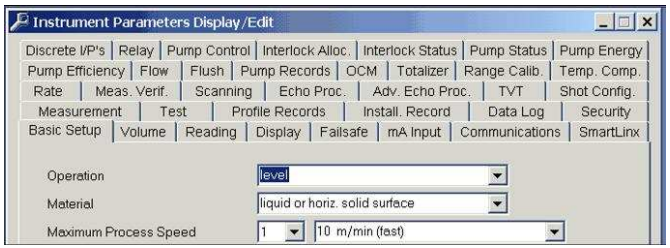


Onde achar “Portuguese”? Acima ou abaixo?

Exemplos de problemas



Exemplos de problemas



Exemplos de problemas

Problemas em Interfaces em geral: NÃO somente em Software !!!

Don Norman, The Design of Everyday Things,

Métricas para medir usabilidade?

- Desempenho durante a realização de tarefas:
  - Conclusão de tarefas (c/ sucesso, parcialmente concluída, não-concluída);
  - Tempo de realização da tarefa;
  - Ocorrência de erros;
- Satisfação subjetiva do usuário e correspondência com os objetivos do usuário;
- Adequação a padrões (normas, recomendações, regras ergonômicas, etc.)
  - Internacionais (ISO)
  - Continentais (Comunidade Européia, MercoSul, etc)
  - Nacionais
  - Institucionais (Style guide)

Interessado em IHC?

- Disciplina INFO1043 – IHC, na **UFRGS** desde **1993**
- Livro:
  - Preece, J. et alli. *Design da Interação*, Bookman, 2005.
- Links interessantes:
  - *Usabilidade*: www.useit.com
  - Sun's guide to Web style
    - http://www.sun.com/980713/webwriting/index.html
  - HCI Bibliography : http://www.hcibib.org/
- **Entre em contato com Prof. Marcelo Soares Pimenta**

Ou simplesmente procure por "Marcelo Pimenta" no Google...

Usabilidade tem futuro !!

Revista INFO-Exame
Setembro, 2007


info EXAME Setembro 2007

TIRAGEM DA EDIÇÃO: 377 236 EXEMPLARES

8 TEM MENSAGEM PRA VOCÊ
10 WWW.INFO.ABRIL.COM.BR
12 CORREIO LIVRE

zap! 15
16 MASHUP
24 360°
26 INTERNET
28 ZONA WIRELESS
30 BLOGS S.A.
32 DATA INFO
34 JOHN C. DVORAK
36 DAGOMIR MARQUEZI
38 SANDRA CARVALHO

TI 59
60 ZOOM
O streaming de vídeo nunca esteve tão em alta nas empresas
66 CIO DO MÊS
Como Germán Quiroga tirou da planta a venda de imóveis pela web na Cyrela
68 TENDÊNCIAS

CAPA 40

12 NEGÓCIOS DE TECNOLOGIA
VEJA ALGUMAS IDÉIAS PROMISSORAS PARA FATURAR O PRIMEIRO MILHÃO EM ATÉ TRÊS ANOS

USABILIDADE DÁ DINHEIRO

CRESCE A PREOCUPAÇÃO DAS EMPRESAS COM SITES AMIGÁVEIS

Quantas vezes você já entrou num site e teve dificuldade para encontrar alguma informação absolutamente básica, como o telefone de contato da empresa? Falhas de usabilidade como essa são comuns, tanto na web como nos aplicativos que rodam nos computadores. Num estudo clássico sobre o tema, a pesquisadora americana Claire Karat demonstrou que cada dólar investido em usabilidade pode trazer um retorno entre 3 e 100 dólares. Essa regra tem sido comprovada em numerosos casos práticos e é o principal argumento de vendas de consultores especializados no tema. Em TI, as oportunidades incluem prestar serviços a sites e a desenvolvedores de aplicativos. Existem várias empresas atuando dessa forma, mas sobra espaço para mais gente. "O momento é ótimo. As pessoas estão deixando de ver a usabilidade como algo pitoresco", diz Amyris Fernandez, coordenadora dos cursos de usabilidade do iGroup.

Um pré-requisito para atuar como consultor nessa área é ter boa familiaridade com as tecnologias de desenvolvimento de aplicativos e sites. Assim, será mais fácil sugerir alterações no produto para melhorá-lo. Também será preciso obter treinamento específico. A PUC-RJ (www.cce.puc-rio.br/artes/espergodesign.htm) e a UFRGS (www.inf.ufrgs.br/esp/cursos/curso2.php) são duas das instituições que oferecem cursos de pós-graduação em usabilidade no Brasil. Além disso, há empresas, como o iGroup, que realizam seminários e cursos rápidos sobre esse tema, alguns via web.

Um profissional capacitado precisa apenas de um notebook para começar a prestar consultoria. Com essa estrutura, é possível fazer a chamada avaliação heurística. Nela, o especialista verifica uma série de itens e determina o nível de usabilidade do aplicativo ou site. Depois, monta um relatório apontando os problemas e as soluções recomendadas. Um serviço mais completo exige um ambiente para testes, do tipo conhecido como sala de focus group. Nela, usuários vão utilizar o aplicativo ou site que está em estudo, seguindo um roteiro. Seus movimentos serão observados e filmados para análise. Como essa sala só é usada durante períodos curtos, é mais barato alugá-la de empresas que realizam pesquisas de mercado.

11

USABILIDADE

O QUE É: CONSULTORIA EM USABILIDADE PARA SITES E DESENVOLVEDORES DE APLICATIVOS

INVESTIMENTO MÍNIMO: 45 MIL REAIS

EQUIPE INICIAL: 1 PESSOA

RETORNO PREVISTO: 1 ANO

RISCO: MÉDIO

WWW.INFO.ABRIL.COM.BR | SETEMBRO 2007 | INFO 51

Módulo (def.)

- Equivalente a uma subrotina (function ou procedure) representando uma etapa de uma tarefa a ser executada pelo software.
  - Unidades autônomas, coerentes e organizadas em arquiteturas robustas.
  - Projeto Modular – permite alcançar boas configurações de módulos
- Formas mais avançadas de módulos:
  - Classes
  - Componentes

Módulos: características(1/2)

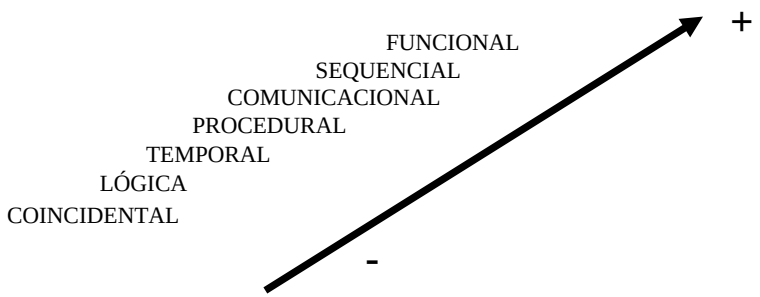
- The choice of a proper module structure is the key to achieving the aims of reusability and extendibility.
- Modules serve for both software decomposition (the top-down view) and software composition (bottom-up).
  - Modular concepts apply to specification and design as well as implementation.
  - A comprehensive definition of modularity must combine several perspectives; the various requirements may sometimes appear at odds with each other, as with decomposability (which encourages top-down methods) and composability (which favors a bottom-up approach).
- Controlling the amount and form of communication between modules is a fundamental step in producing a good modular architecture.

Módulos: características(2/2)

- The long-term integrity of modular system structures requires information hiding, which enforces a rigorous separation of interface and implementation.
- Uniform access frees clients from internal representation choices in their suppliers.
- A closed module is one that may be used, through its interface, by client modules.
- An open module is one that is still subject to extension.
- Effective project management requires support for modules that are both open and closed. But traditional approaches to design and programming do not permit this.
- The principle of Single Choice directs us to limit the dissemination of exhaustive knowledge about variants of a certain notion.

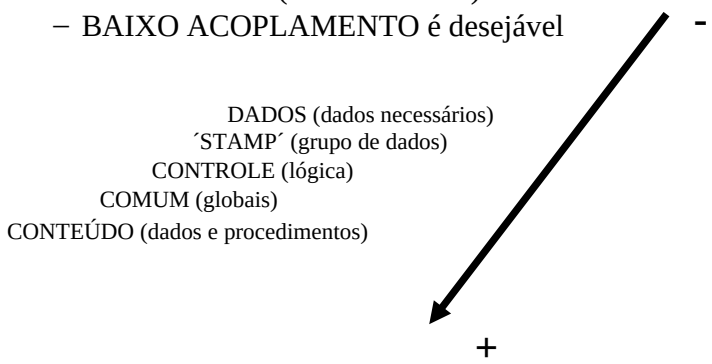
Coesão

- Unidade Funcional (´força modular´)
- ´Elementos do módulo são direcionados , focados e relacionados idealmente para a realização de UM objetivo´
- Categorias de coesão
  - ALTA COESÃO é desejável



Acoplamento

- Interdependência entre os módulos
- Categorias de coesão
  - Não exaustivas (são *Guidelines*)
  - BAIXO ACOPLAMENTO é desejável



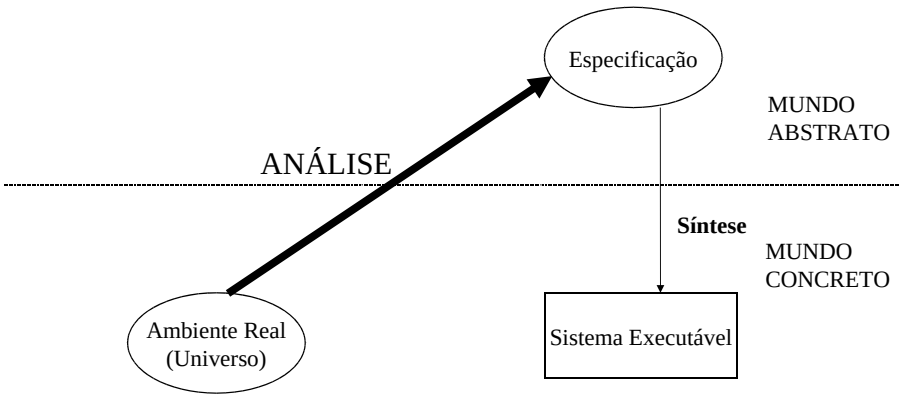
Critérios de modularidade

1. Decomponibilidade Modular
   - Habilita decomposição (problema -> subproblemas)
2. Componibilidade Modular
   - Habilita composição (subproblemas -> problema)
3. Compreensibilidade Modular
   - Pode-se entender cada módulo sem (muita) dependência de outros
4. Continuidade Modular
   - (Pouca) alteração no problema -> alteração de (poucos) módulos
5. Proteção Modular
   - Condição anormal sem muitos efeitos colaterais externos

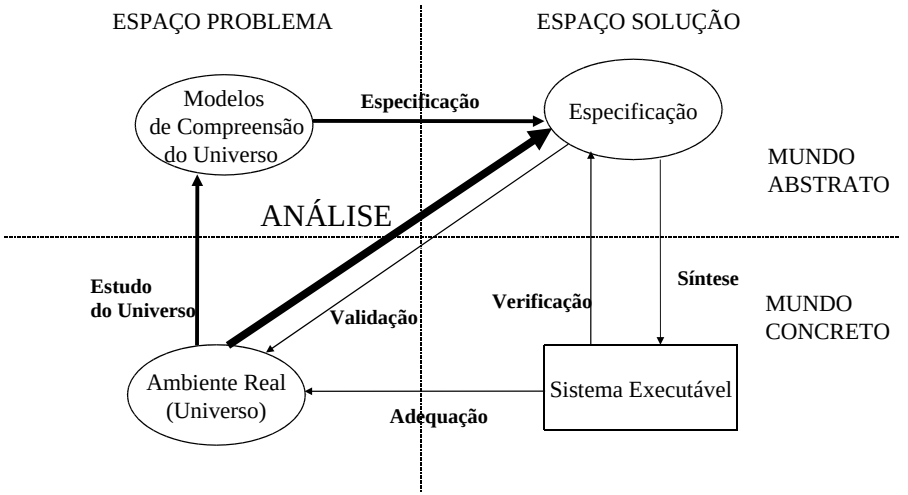
Regras de modularidade

1.Mapeamento direto (Domínio Solução e Domínio do Problema)
   - Módulos devem se compatíveis com elementos do domínio do problema
2. Poucas Interfaces
   - Cada módulo deve se comunicar com tão poucos outros quanto possível.
3. Pequenas Interfaces (Baixo Acoplamento)
   - Se dois módulos se comunicarem, eles devem trocar tão pouca informação quanto possível.
4. Interfaces Explícitas
   - Sempre que dois módulos **A** e **B** se comunicarem, isto deve ser óbvio no texto de **A** ou **B**, ou de ambos.
5. Ocultamento de Informação
   - Toda informação sobre um módulo deve ser privada ao módulo, a menos que seja explicitamente declarada pública.

Abstrato e Concreto



Solução e Problema



Princípios de modularidade

- The Linguistic Modular Units principle.
  - Modules must correspond to syntactic units in the language used.
- The Self-Documentation principle.
  - The module should *contain* its documentation: all information about the module is part of the module itself.
- The Uniform Access principle.
  - All services offered by a module should be available through a uniform notation
- The Open-Closed principle.
  - Modules should be both open (to extensions) and closed (available for use by other modules.
- The Single Choice principle.
  - Whenever a software system must support a set of alternatives, one and only
  - one module in the system should know their exhaustive list.

Leitura Recomendada

- "Modularity"

Cap. 3 do livro

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

PDF disponível no moodle da disciplina