

Técnicas de Avaliação

Interação Usuário-Computador
(SCE-224)

Prof. Rudinei Goularte

Aula de hoje...

- Técnicas de Avaliação
 - Inspeção de Usabilidade
 - Avaliação Heurística
 - Percurso Cognitivo
 - Testes de Usabilidade
 - Think Aloud
 - Medidas de Performance

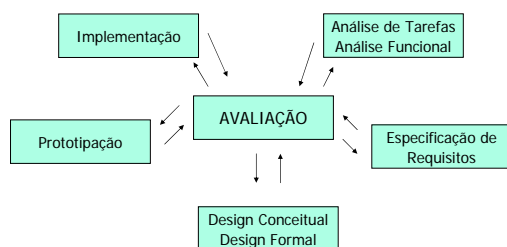
Avaliação: por quê ???

- Avaliar adequação de designs
- Testar a **usabilidade** e a **funcionalidade** do sistema
 - Requisitos dos usuários estão efetivamente sendo atendidos?
 - Sistema comporta-se como esperado?
 - Usuário fica satisfeito e consegue realizar suas tarefas?
- Pode ocorrer...
 - Em laboratório
 - In loco, i.e., no local de uso
- Pode ou não envolver usuários finais

Avaliação do Design

- Avaliação **não** deve ser uma fase única no final do processo de design

Avaliação do Design



Ciclo de Vida Estrela (adaptado de Hix e Hartson, 1993)

Avaliação do Design

- Método de avaliação deve ser cuidadosamente escolhido
 - Classificação em duas dimensões:
 - Se usuários reais estão envolvidos ou não
 - Se a interface está ou não implementada
 - Implementação: qualquer protótipo executável

Avaliação: Grupos de Métodos

- **Inspeção de Usabilidade** (Predictive Evaluation)
 - Pode ou não envolver usuários
 - Aplicável a qualquer fase do desenvolvimento de um sistema (implementado ou não)
- **Testes de Usabilidade**
 - Métodos de avaliação **centrados no usuário**
 - Métodos experimentais ou empíricos
 - Métodos observacionais
 - Métodos de questionamento (consultivos)
 - É necessária uma implementação

Avaliação: Estudos em Laboratório

- Experimental: formal e objetiva
- Com os designers, sem usuários
- Com usuários (Testes de Usabilidade)
 - Facilidades de registro áudio-visual
 - Espelhos de duas faces
 - Computadores instrumentados

Avaliação: Estudos em Laboratório

- Situação artificial
 - Sem interrupção, sem contexto, sem conversas...
- Interessante quando...
 - Situação de uso real é perigosa ou remota
 - Deseja-se avaliar tarefas restritas
 - Deseja-se deliberadamente manipular o contexto
 - Para comparar designs alternativos em um contexto controlado
 - Detectar problemas/ estudar aspectos específicos
 - Observar procedimentos pouco usados

Avaliação: in Loco

- Designer/avaliador vai ao ambiente de trabalho do usuário para observá-lo em ação com o sistema
- Situação de uso real, "quase natural"... porém...
 - Interrupções, ruído, influência da presença do 'observador', humano ou não...
- Certas tarefas são difíceis de observar em laboratório ou em loco
 - Ex.: as muito longas, que consomem dias ou meses...
- Custo é um aspecto relevante
 - Tirar usuário do trabalho x montar uma infra-estrutura de observação no local (e talvez atrapalhar...)

Avaliação do Design

- Crítica às limitações de testar hipóteses científicas em um ambiente fechado
 - Lab não é o mundo real
 - Não pode controlar todas as variáveis
 - Contexto é negligenciado
 - Tarefas testadas são artificiais e curtas

Avaliação do Design

- Pode ocorrer mais cedo no processo
 - Quanto antes detectar problemas, melhor...
- Muitas técnicas que não envolvem necessariamente usuários
 - Dependem do designer, de um especialista em fatores humanos
- Objetivo: identificar se princípios cognitivos ou resultados empíricos aceitos estão sendo violados
 - Em geral, métodos analíticos...



Inspeção de Usabilidade

- Métodos informais de avaliação (empíricos)
 - Usam a habilidade e experiência dos avaliadores
- Alguns métodos (não mutuamente exclusivos)
 - Percurso Cognitivo (Cognitive walkthrough)
 - Avaliação Heurística
 - Revisão de Guidelines
 - Inspeção de Consistência
- Estudos demonstram que são um bom complemento aos testes de usabilidade com usuários



Inspeção de Usabilidade: Objetivos

- Encontrar problemas de **usabilidade** no design de uma interface
- Fazer recomendações no sentido de melhorar a usabilidade do design
- **Usabilidade**
 - Facilidade de aprendizado, eficiência de uso, quão 'agradável' ao usuário, frequência de ocorrência e severidade de erros do usuário



Inspeção de Usabilidade

- Trabalho de inspeção: identificar, classificar e contar o número de **problemas de usabilidade**
- O que é um '**problema de usabilidade**'?



Inspeção de Usabilidade

- Problema de usabilidade
 - Qualquer aspecto do design que pode causar uma usabilidade reduzida ao usuário final do sistema
- Muitas vezes um único problema se manifesta de várias formas
 - Um único elemento da interface pode dificultar o aprendizado, torná-lo lento, causar erros, ou simplesmente ser feio/desagradável
 - Difícil estabelecer distinções, mas...
 - Na maioria dos casos bom senso é suficiente



Inspeção de Usabilidade

- Identificar os problemas é apenas parte do processo...
- A equipe deve fazer um redesign da interface
 - Tentar corrigir a maior quantidade possível de problemas
 - Tipicamente, os relatórios gerados a partir da lista de problemas contém sugestões...
 - Muitos métodos sugerem reuniões entre a equipe de avaliadores e a equipe de desenvolvimento
 - Discussão de soluções de redesign
 - Custo associado ao redesign



Inspeção de Usabilidade

- Problemas devem ser priorizados quanto à sua gravidade
 - Grau de severidade: geralmente derivados do impacto causado pelo problema no usuário e no mercado
 - Compromisso: custo X severidade

Inspeção de Usabilidade

- Métodos de inspeção podem ser aplicados em fases iniciais ou finais do design
- Resultado
 - Relatório dos problemas identificados, com recomendações para mudança
- Ainda...
 - Discussão ou apresentação para os designers e gerentes do projeto

Inspeção de Usabilidade: Métodos

- Avaliação Heurística
 - Inspeção da interface tendo como base uma pequena lista de heurísticas de usabilidade
- Percurso Cognitivo (Cognitive Walkthrough)
 - Avaliador simula um usuário típico 'percorrendo' a interface para executar tarefas típicas
 - Tarefas mais frequentes
 - Tarefas críticas

Inspeção de Usabilidade: Vantagens

- Métodos facilmente **integráveis** a processos convencionais de desenvolvimento de software
- Fornecem **evidências concretas** de quais aspectos do design devem ser aperfeiçoados
- Não exigem muita **experiência** e longo **treinamento** para que possam ser utilizados
- Boa experiência educacional para **designers novatos**

Avaliação Heurística

- Nielsen, 1993: Discount Usability Engineering
 - http://www.useit.com/papers/querrilla_hci.html
- Métodos baratos, rápidos e fáceis de serem usados
- Avaliação heurística é o principal método
 - Fácil: pode ser ensinada em 4 hs
 - Rápida: maioria das avaliações requer cerca de um dia
 - Barata: tanto quanto se deseje
 - Pequeno conjunto de avaliadores examina a interface e julga suas características em face de reconhecidos **princípios de usabilidade** (heurísticas)

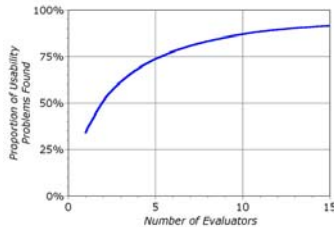
Avaliação Heurística

- Heurísticas
 - Regras gerais que objetivam descrever propriedades comuns de interfaces usáveis
 - Exemplos de Heurísticas
 - Diálogo simples e natural
 - Fale a linguagem do usuário
 - Minimize a memorização do usuário
 - Seja consistente
 - Ofereça feedback
 - Ofereça mensagens positivas
 - Evite armadilhas
 - Ofereça atalhos

Avaliação Heurística

- Deve ser feita por mais de uma pessoa
 - Experiência indica que diferentes pessoas encontram diferentes problemas
 - Resultados melhoram significativamente utilizando múltiplos avaliadores
 - Três a cinco

Quantos avaliadores?



Avaliação Heurística

- Em um primeiro momento...
 - Avaliações individuais
- Sessão de avaliação
 - Cada avaliador percorre a interface pelo menos duas vezes inspecionando os diferentes componentes de diálogo
 - Problemas detectados são associados às **heurísticas violadas**

Avaliação Heurística

- Sessões de avaliação individual
 - Tipicamente 2 hs
 - Mais tempo para interfaces grandes ou complexas (muitos componentes de diálogo)
 - Melhor dividir a avaliação em sessões curtas, cada qual avaliando um cenário específico de interação
 - Além das heurísticas gerais, podem também considerar heurísticas específicas da categoria do produto analisado

Avaliação Heurística

- Sessões de avaliação individual
 - Avaliador deve justificar o que considera um problema com base nas heurísticas que considera violadas
 - Deve ser o mais específico possível
 - Deve listar cada problema separadamente
 - Avaliador pode ser deixado por conta própria...
 - Caso seja um perito no domínio do sistema, ...
 - Caso o sistema seja de domínio geral (voltado para população em geral)
 - Caso contrário, deve ser auxiliado
 - Acompanhamento por pessoa da equipe de desenvolvimento
 - Prover cenários típicos de uso
 - Construído com base na análise de tarefas reais

Avaliação Heurística

- Listas de problemas dos avaliadores consolidadas em uma única lista
- Discussão com equipe de desenvolvimento
- Atribuição de graus de severidade aos problemas
 - A partir da lista de problemas, não é difícil gerar um design revisado
 - Redesign baseado na diretrizes fornecidas pelos princípios de usabilidade violados

Avaliação Heurística

- Exemplos de problemas encontrados com o uso:
 - Rocha e Baranauskas, Cap. 4
 - http://www.sims.berkeley.edu/courses/is213/s01/projects/P1/travelite_HE.htm (alunos aplicando...)
 - http://www.bls.gov/ore/htm_papers/st960160.htm (problema real)
 - Ver também:
 - <http://www.sitepoint.com/article/520> (HE step by step guide)



Avaliação Heurística

- Graus de severidade do problema
 - Combinação de fatores:
 - Frequência de ocorrência
 - comum ou raro
 - Impacto do problema
 - fácil ou difícil
 - Persistência do problema
 - esporádica ou repetidamente
 - Impacto do problema no mercado
 - popularidade do produto



Avaliação Heurística

- Graus de Severidade
 - Não concordo que isso é um problema de usabilidade
 - Problema cosmético - correção pode ser feita se houver tempo
 - Problema menor - correção pode ter baixa prioridade
 - Problema grave - correção deve ter alta prioridade
 - Catástrofe de usabilidade - correção é imperativa
- Valores atribuídos depois da consolidação dos problemas em uma lista única, pelo grupo de avaliadores
- Coerência na atribuição de valores depende da experiência dos avaliadores



Avaliação Heurística

- Adequada tanto para detectar problemas graves como problemas menores
- Adequada para avaliar o design (em estágios iniciais ou avançados)
- Também pode ser usada para avaliar implementações
- Pode ser usada mesmo por quem não tem muita experiência com avaliação



Avaliação Heurística: Procedimento

1. Obter entradas
2. Realizar avaliação independente
3. Discussão/Coleta
4. Atribuição de taxa de severidade



Avaliação Heurística: Procedimento

1. Obter entradas
 - Time de avaliação (treinado no método)
 - Familiaridade com o domínio
 - Storyboards/cenários/protótipos que mostram detalhes suficientes
 - Heurísticas!!!



Avaliação Heurística: Procedimento

2. Realizar avaliação independente
 - Julgar protótipo relativamente ao conjunto de heurísticas
 - Quais heurísticas?
 - Shneidermann: Oito Regras de Ouro
 - Nielsen: 10 heurísticas
 - Outras...

O que é um bug de usabilidade?

- O que quer que o avaliador julgue como tal!
- Durante a avaliação não se deve focar no fato do erro ser relevante ou não.

Avaliação Heurística: Procedimento

- 3. Discussão/Coleta
 - Organizar todos os problemas encontrados
- 4. Atribuição de taxa de severidade
 - Filtrar quais são problemas importantes

Avaliação Heurística

Dez Heurísticas de Usabilidade [Nielsen]:

1. Visibilidade do status do sistema

- O sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de feedback apropriado dentro de um tempo de resposta razoável

Avaliação Heurística

Dez Heurísticas de Usabilidade [Nielsen]:

2. Compatibilidade do sistema com o mundo real

- O sistema deve falar a linguagem do usuário, com palavras, frases e conceitos familiares ao usuário (em vez de termos do sistema)
- O projeto deve seguir convenções do mundo real numa ordem natural e lógica

Avaliação Heurística

Dez Heurísticas de Usabilidade [Nielsen]:

3. Controle do usuário e liberdade

- O sistema deve oferecer saídas claras (saídas de emergência) para situações nas quais usuários se encontram por terem escolhido funções erradas do sistema
- Não deve ser necessário um diálogo extenso
- O sistema deve suportar undo e redo

Avaliação Heurística

Dez Heurísticas de Usabilidade [Nielsen]:

4. Consistência e padrões

- O sistema deve apresentar informações de modo consistente e padronizado (sejam os padrões formais ou não)
- Os usuários não devem ter que imaginar que palavras ou situações ou ações diferentes significam, de fato, a mesma coisa.



Avaliação Heurística

Dez Heurísticas de Usabilidade [Nielsen]:

5. Prevenção de erros

- Tente evitar que o erro aconteça, informando o usuário sobre as consequências de suas ações ou, se possível, impedir ações que levariam a uma situação de erro
 - Melhor que boas mensagens de erro é preparar um projeto que impede os erros de acontecerem



Avaliação Heurística

Dez Heurísticas de Usabilidade [Nielsen]:

6. Reconhecimento ao invés de lembrança (memorização)

- O sistema deve fazer com que objetos, ações e opções estejam claramente visíveis
- O usuário não deve ser obrigado a lembrar informações de uma parte do diálogo para outra
- Instruções para uso do sistema devem estar visíveis ou facilmente acessíveis sempre que apropriado



Avaliação Heurística

Dez Heurísticas de Usabilidade [Nielsen]:

7. Flexibilidade e eficiência de uso

- O sistema deve oferecer aceleradores ao usuário especialista, os quais são invisíveis aos novatos
- O sistema deve permitir que usuários customizem ações frequentes



Avaliação Heurística

Dez Heurísticas de Usabilidade [Nielsen]:

8. Design estético e minimalista

- Os diálogos usuário-sistema não devem conter informações que são irrelevantes ou raramente utilizadas
- Cada unidade de informação no diálogo compete com as unidades que são de fato relevantes e, assim, diminuem a sua visibilidade relativa



Avaliação Heurística

Dez Heurísticas de Usabilidade [Nielsen]:

9. Ajudar os usuários a reconhecer, diagnosticar e corrigir erros

- Mensagens de erro deve ser expressas em linguagem simples, sem códigos, indicando precisamente o problema e sugerir uma solução de modo construtivo



Avaliação Heurística

Dez Heurísticas de Usabilidade [Nielsen]:

10. Ajuda e documentação

- Apesar de ser melhor se o sistema puder ser utilizado sem documentação alguma, há situações em que ajuda e documentação se fazem necessários
- Tal apoio deve ser fácil de ser encontrado, estar focado na tarefa do usuário, listar passos concretos a serem realizados, e não serem grandes

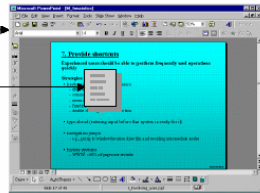
Ofereça atalhos

- Usuários experientes devem ser capazes de realizar tarefas mais rapidamente
- Flexibilidade
- Exemplos
 - Teclas de atalhos, barras de ferramentas
 - Preenchimento automático de campos

Barras de ferramentas

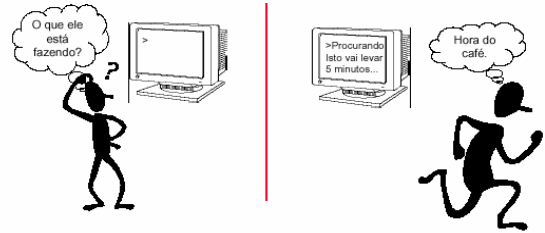
Menu de atalho

Ctrl+B, Ctrl+P, ...



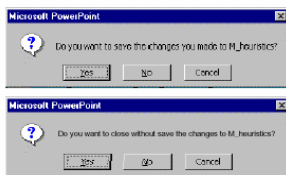
Ofereça feedback

- Mantenha o usuário continuamente informado sobre
 - O que o sistema está fazendo
 - Como o comando foi interpretado
 - Os resultados do comando



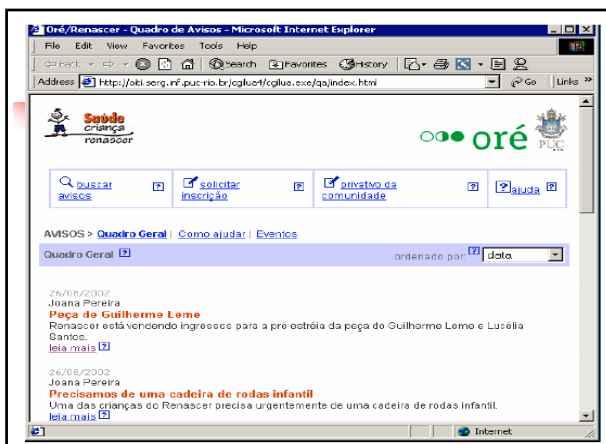
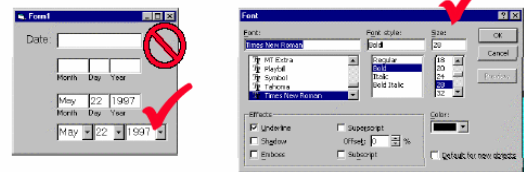
Evite armadilhas

- Erro é humano.
- Comandos devem evitar que pessoas cometam erros.
- Exemplo – Fechando o PowerPoint com mais de uma apresentação aberta...



Minimize a memorização do usuário

- Para pessoas, reconhecer é mais fácil do que lembrar.
- Para computadores, memorizar é mais fácil do que reconhecer.
- Use *menus* e *caixas de diálogo* no lugar de *linhas de comandos*.
- Deixe os objetos visíveis para o usuário, mas evite excessos.



Problema: O usuário não conseguirá entender que o texto "privativo da comunidade" lhe dá acesso a um espaço com mais funcionalidades do que aquele em que ele se encontra.

Heurística violada: correspondência entre o sistema e o mundo real

Explicação: Embora na sede da ASCR tenha alguns espaços que normalmente só são acessíveis por membros da comunidade, o usuário não utiliza a palavra "privativo" no seu cotidiano e não saberá a que ela se refere.

Gravidade: 4 – catastrófico. O usuário não conseguirá acessar as funcionalidades que estão disponíveis apenas para membros, como por exemplo ler avisos específicos ao trabalho em que está envolvido, ou criar um novo aviso.

Problema: O texto "Quadro geral" não transmite a ideia de que está sendo visualizado

Heurística violada: reconhecimento

Explicação: O que está sendo mostrado na seção denominada Quadro Geral são os avisos do Quadro de Avisos que foram colocados em destaque

Gravidade: 3 – grave. Como os usuários na sua maioria têm pouca experiência com informática, pode não ficar claro para eles que os avisos no Quadro geral são aqueles selecionados para estarem em destaque e podem aparecer também em outras seções. Isto pode comprometer o entendimento do usuário sobre como utilizar o Quadro de Avisos.

Avaliação Heurística

"Oito Regras de Ouro" [Shneidermann]:

1. Lute por consistência
2. Possibilite aos usuários experientes o uso de atalhos
3. Ofereça feedback informativo
4. Projete diálogos para produzir fechamento (closure)
5. Ofereça prevenção e fácil resolução de erros
6. Permita reverter ações facilmente
7. Dê o controle ao usuário
8. Reduza sobrecarga de memória

Percurso Cognitivo

- Revisores avaliam a interface proposta no contexto da **execução** de uma ou mais tarefas do usuário
- Origem: Walkthrough para inspeção de código (ES)
 - Percorrer uma sequência de código, passo a passo, detalhadamente, para checar certas características
- Percurso Cognitivo
 - Passos a serem seguidos pelo usuário para executar uma certa tarefa
 - Avaliadores percorrem a sequência de tarefas e ações, passo a passo, para detectar potenciais problemas de usabilidade

Percurso Cognitivo

- Foco: Avaliar um design quanto à sua facilidade de aprendizagem, particularmente **aprendizagem por exploração**
 - Avaliadores verificam se cada passo é ou não adequado a um usuário novato
 - O usuário seria bem sucedido ao tentar executá-lo?
- Processo dividido em duas fases básicas
 - Fase preparatória
 - Fase de análise

Percurso Cognitivo: Fase Preparatória

- É necessário...
 - Uma descrição dos usuários
 - Quem são e que tipo de experiência e conhecimento os avaliadores podem assumir que eles têm
 - Uma descrição da tarefa do usuário
 - Deve ser uma tarefa representativa
 - Uma lista completa das ações
 - Necessárias para completar a tarefa com o protótipo dado
 - Uma descrição do protótipo do sistema
 - Pode ser incompleta, mas razoavelmente detalhada
 - Detalhes como posicionamento e termos usados no menu podem fazer enorme diferença

Percurso Cognitivo: Fase de Análise

- Contar uma 'estória verossímil' sobre como o usuário iria interagir...
- Para cada ação, em cada uma das tarefas, os analistas respondem 4 questões: Os usuários...
 1. Farão a ação correta para atingir o resultado desejado?
 2. Perceberão que a ação correta está disponível?
 3. Irão associar a ação correta ao efeito desejado?
 4. Se a ação correta for executada, perceberão um progresso em relação à tarefa?
- Estória verossímil de sucesso
- Estória verossímil de fracasso
 - Se a resposta à alguma das questões acima é negativa

Percurso Cognitivo

- Pode ser efetuado sobre uma especificação da interface
 - Em papel
 - Protótipo não funcional
 - Protótipo funcional
- Pode ser individual ou em grupo
- Grupo pode envolver...
 - Outros designers, engenheiros de software, representantes de outras unidades organizacionais (publicidade, treinamento, documentação)

Percurso Cognitivo

- Definindo as entradas...
 - Quem são os usuários do sistema?
 - Qual sua experiência e conhecimento técnico?
 - Ex. Usuários de Linux, pessoas que trabalham com o MS Word...
 - Quais tarefas serão analisadas?
 - Todas as que o sistema suporta... ou as mais relevantes... ou as mais problemáticas...
 - Coleção de tarefas deve ser representativa
 - Qual a sequência de ações correta para cada tarefa?
 - Descrever...
 - Granularidade da descrição depende da expertise do usuário-alvo
 - Qual a interface?
 - Como cada tarefa/ação é 'prompted' pelo sistema?
 - Protótipo em papel ou implementação...

Percurso Cognitivo

- Percorrendo as tarefas/ações, respondendo às 4 perguntas...
 1. Usuários farão a ação correta para atingir o resultado desejado?
 - Usuário vai saber como iniciar a tarefa?
 - Ex. Sabe o que precisa ser feito para começar a tarefa?
 2. Usuários perceberão que a ação correta está disponível?
 - Ex. Opção para disparar a tarefa é claramente indicada no menu?
 3. Usuários irão associar a ação correta ao efeito desejado?
 - Ex. Associar um ícone com o que deseja fazer?
 4. Se a ação correta for executada, perceberão um progresso em relação à tarefa?
 - Ex. Há feedback do que ocorreu?

Percurso Cognitivo

- Questões servem de guia para construir as histórias
 - Exemplos de histórias:
 - DFAB: Seção 11.4.1; Rocha e Baranauskas: Cap. 4
- Importante registrar a informação gerada durante o percurso
 - Anotações, vídeo
- Resultados do percurso podem ser usados para corrigir problemas

Usuários típicos: Membros da comunidade ASCR com pouca familiaridade no uso de computadores, e muitas vezes com uma resistência ao seu uso

Tarefa: Incluir um aviso no quadro de avisos

Cenário: O usuário deseja criar um aviso no quadro de avisos para os voluntários da recreação.

Sequência correta de ações:

1. Entrar no espaço privativo do Quadro de Avisos
2. Entrar login e senha
3. Selecionar a opção Criar Avisos
4. Preencher campos (pelo menos os obrigatórios)
5. Confirmar criação do aviso



Percurso Cognitivo

- Limitações
 - Enfoque em um único atributo de usabilidade: facilidade de aprendizagem
 - Não deve ser usado como único método de avaliação
- Vantagens
 - Detecta conflitos entre designer e usuário quanto à concepção das tarefas
 - Detecta escolhas ruins/inconsistentes de nomes, rótulos, terminologia
 - Detecta respostas inadequadas a ações



Testes de Usabilidade

- Teste **com usuário** é fundamental!!!
 - Crescente preocupação com o usuário
 - Restrições de tempo e recursos
- Testes devem ser cuidadosamente planejados e preparados



Testes de Usabilidade

- Qual o objetivo do teste?
 - Melhorar um design em desenvolvimento
 - Quais aspectos estão bons? Quais estão ruins? Como o design pode ser melhorado
 - Forma mais "gradual" de analisar o design
 - **Think Aloud**
 - Avaliar a qualidade global de uma interface em fase final de definição
 - Testes que dêem **Medidas de Performance**



Testes de Usabilidade

- Problemas: Confiabilidade e Validade
- **Confiabilidade**: grau de certeza de que o mesmo resultado será obtido se o teste for repetido
- **Validade**: resultados do teste refletem os aspectos de interface que se deseja testar
 - Resultados obtidos têm significado fora do laboratório?
 - Cuidados: diferenças individuais entre usuários, escolha de usuários, escolha de tarefas, diferença entre equipamentos
 - Validade requer planejamento cuidadoso e tratamento estatístico adequado dos dados coletados



Testes de Usabilidade

- Escolha de **usuários**
 - Representativos de usuários reais do sistema
 - Idade, nível educacional
 - Experiência prévia com uso de computadores, conhecimento do domínio, ...
 - Ideal: usuários reais!!!
 - Nem sempre é possível ...
- **Experimentadores**
 - Preparação: conhecimento sobre a aplicação, sobre a interface
 - Pode até ser os próprios projetistas, mas isso requer um certo cuidado...



Testes de Usabilidade

- **Tarefas**
 - Representativas do uso da interface
 - Dar boa cobertura aos componentes mais significativos
 - Poder ser completadas no tempo razoável para uma sessão de teste (1 a 3 horas)
 - Grau de dificuldade gradativa
 - Planejadas para que possam ser interrompidas a qualquer tempo
 - Descrição de cada tarefa a ser efetuada deve ser fornecida por escrito
 - Realista e inserida em um cenário de uso



Testes de Usabilidade: Etapas

- **Preparação**
 - O local, os equipamentos...
- **Introdução**
 - Explicar aos usuários, colocá-los à vontade, esclarecer objetivos
- **Teste**
 - Evitar interferir, evitar ajudar
- **Sessão Final**
 - Ouvir usuários

Testes de Usabilidade: Etapas

- O que os usuários devem saber?????
 - Propósito do teste é avaliar o sistema, não o usuário
 - Podem expressar suas opiniões livremente
 - Resultados do teste servirão para melhorar a interface
 - O sistema é confidencial
 - Participação no teste é voluntária, e pode ser interrompida por ele
 - Resultados do teste não são públicos, anonimato dos participantes é garantido
 - Explicar caso estejam sendo feitas gravações em vídeo/áudio
 - Explicar que pode perguntar, mas nem sempre o experimentador pode responder
 - Instruções específicas sobre o teste

Testes de Usabilidade

- Gravações em vídeo podem ser um recurso valioso para avaliação posterior
 - Ideal é não identificar as pessoas
- Entretanto, a análise é difícil...
 - Importante complementar registro com anotações, log files...
- Protocolo Think-Aloud
 - Atmosfera informal e agradável

Think Aloud (Pensando em Voz Alta)

- Usuário verbaliza o que está pensando enquanto usa o sistema
- Expectativa é que os pensamentos mostrem como o usuário interpreta cada item da interface
- Inadequada quando o objetivo é obter medidas de desempenho
 - Usuários tendem a ficar mais lentos e cometer mais erros
- Requer experimentador bem-preparado
 - Estimular o usuário a falar
 - Não interferir no uso do sistema

Think Aloud (Pensando em Voz Alta)

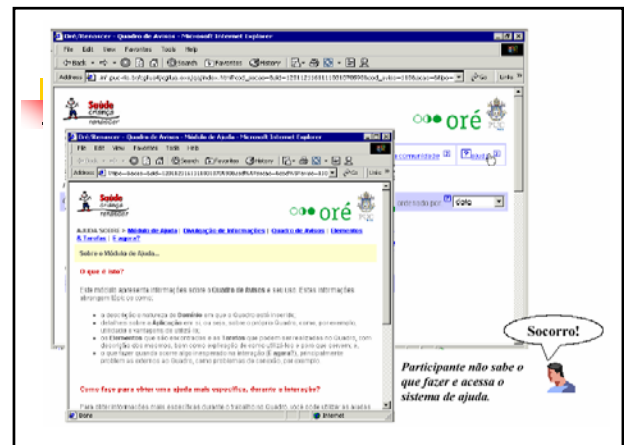
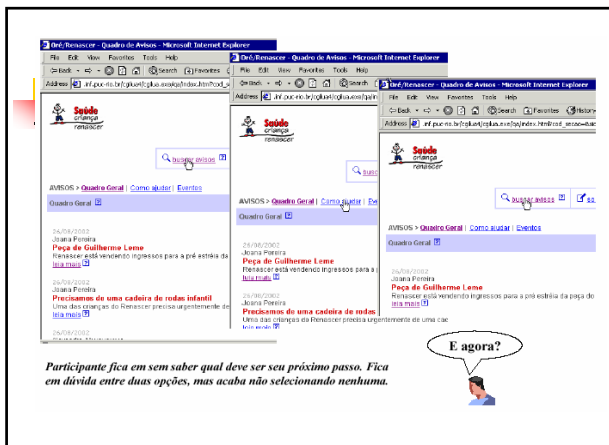
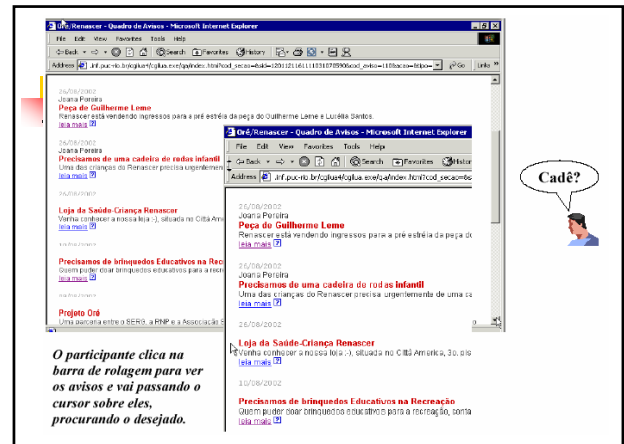
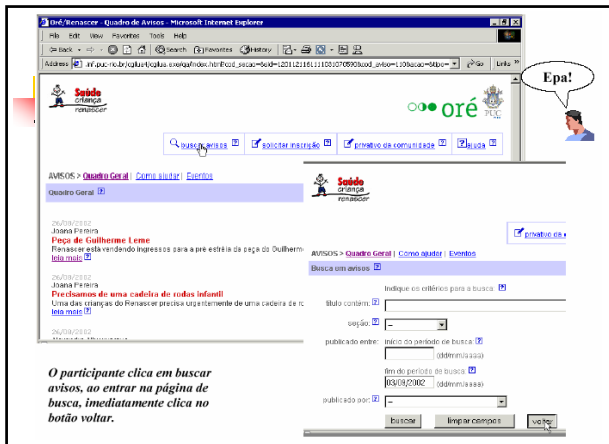
- Formas de questionamento usuais:
 - O que você está pensando agora?
 - O que você acha que essa mensagem significa (depois do usuário notar a mensagem) ?
 - Caso o usuário pergunte se pode fazer alguma coisa: O que você acha que vai acontecer se fizer isso?
 - Caso o usuário se mostre surpreso: Era isso que você esperava que iria acontecer? O que esperava?

Comunicabilidade — Interjeições

- | | |
|-------------------|-------------------------|
| • Cadê? | • Por que não funciona? |
| • E agora? | • Ué, o que houve? |
| • Que isso? | • Para mim está bom. |
| • Objeto ou ação? | • Não dá. |
| • Epa! | • Vai de outro jeito. |
| • Onde estou? | • Não, obrigado. |
| • Assim não dá. | • Socorro! |

Comunicabilidade x problemas na interação

Interjeições	Problemas
Cadê?	navegação
Que é isso? E Agora?	atribuição de significado
Epa! Onde estou?	navegação
Por que não funciona? Ué, o que houve?	atribuição de significado
Para mim está bom...	atribuição de significado
Deixa para lá... Não, obrigado.	affordance baixa/ inexistente affordance recusada
Não dá.	fracasso da tarefa



Think Aloud (Pensando em Voz Alta)

- Vantagem: mostra **o que** o usuário está fazendo e **porque** está fazendo, enquanto está fazendo
- Boa estratégia...
 - Usuários trabalhando aos pares
- Outra alternativa...
 - Pedir que os usuários comentem depois suas ações gravadas em vídeo

Medidas de Performance

- Medidas quantitativas de desempenho
 - Importantes...
 - Para avaliar se os objetivos de usabilidade foram atendidos
 - Certos objetivos podem ser avaliados por medidas quantificáveis
 - Para comparar produtos competitivos
 - Para pesquisas em fatores humanos
 - Ex. Eficiência de uso (uma guideline de usabilidade)
 - Medir tempos para executar tarefas...
- Em geral, um único teste coleta um pequeno sub-conjunto de medidas

Medidas Típicas Quantificáveis

- Tempo que o usuário gasta para executar uma tarefa
- Número de tarefas completadas em um intervalo de tempo
- Razão entre interações de sucesso e de erro
- Número de erros do usuário
- Número de ações errôneas imediatamente subsequentes
- Número de comandos distintos utilizados pelo usuário
- Número de comandos nunca utilizados
- Frequência de uso do help ou manuais, tempo de consulta
- Quantas vezes o manual resolveu o problema do usuário
- Proporção entre comentários do usuário favoráveis e críticos
- Quantidade de 'tempo morto'
- Número de vezes que o usuário desviou do objetivo da tarefa

fator	método de medição	pior caso	nível almejado	melhor caso
Facilidade de uso	Número de erros cometidos	Mais de 10 erros	No máximo 3 erros	Nenhum erro
Facilidade de uso	Porcentagem de vezes que o usuário vai ao sistema de ajuda	Para cada tarefa vai pelo menos 1 vez.	Apenas a 1ª vez que realiza uma tarefa complexa	Nunca
Eficiência para criar aviso	Tempo gasto para criar um aviso	5 min	40 segundos	20 segundos (tempo para digitar campos)
Eficiência para encontrar aviso	Tempo gasto para encontrar um aviso	Não encontrar o aviso	30 segundos	10 segundos (tempo para digitar alguns campos no mecanismo de busca)
Utilidade	Frequência de uso	Uma vez a cada três dias ou menos frequente	Uma vez ao dia	Mais de uma vez ao dia
Eficiência do sistema de ajuda	Porcentagem das vezes que usuário encontrou o que procura no sistema de ajuda	Nunca	Acima de 90% das vezes	100% das vezes
Eficiência do sistema de ajuda	Consegue resolver o problema com base no conteúdo de ajuda	Nunca	Acima de 90% das vezes	100% das vezes
Avaliação inicial	Questionário (subjetivo)	Negativa	Positivo	Muito positivo

Testes de Usabilidade

- A maioria dos testes de usabilidade são feitos em laboratório
 - Usuários são observados diretamente pelos avaliadores
- Localização remota e distribuída dos usuários
 - Observador e usuário separados em tempo e espaço
- Testes de campo
 - Sistemas colocados em ambientes de uso reais
 - Coleta de dados automática pelo sistema
 - Arquivos de log

Testes de Usabilidade

- Laboratórios de Usabilidade
 - Equipe de especialistas em teste e design de interfaces
 - Equipamentos para monitoração
 - BD centralizada para registrar a performance e o log de uso de produtos testados
 - Contato com a equipe de desenvolvimento
 - Plano de teste
 - Participação efetiva no projeto

Testes de Usabilidade

- Laboratórios de Usabilidade
 - Usability Laboratories: A 1994 Survey
 - <http://www.useit.com/papers/uselabs.html>
 - Microsoft
 - <http://www.microsoft.com/usability/default.htm>
 - Sun
 - <http://www.sun.com/usability/>
 - Empresa que vende equipamentos para labs, de usabilidade
 - <http://www.usabilitysystems.com/>

- Técnicas de Avaliação
 - Inspeção de Usabilidade
 - Avaliação Heurística
 - Percurso Cognitivo
 - Testes de Usabilidade
 - Think Aloud
 - Medidas de Performance

