

Avaliação de Sistemas



Métodos de Avaliação

Métodos de Avaliação

Avaliação através de Inspeção

- Avaliação Heurística
- Percuso Cognitivo
- Método de Inspeção Semiótica

Avaliação através de Investigação

Questionários

Entrevistas

Focus Group, Brainstorming

Avaliação através de Observação

- Teste de Usabilidade
- Método de Avaliação de Comunicabilidade
- Protótipo em papel

Métodos de Avaliação

Avaliação através de Inspeção

- Permite ao avaliador (inspetor) examinar uma solução para tentar antever as possíveis consequências de certas decisões de design;
- Não envolvem diretamente o usuário;
- Os avaliadores tentam se colocar no lugar do usuário com determinado perfil;

Avaliação Heurística



Avaliação Heurística (Nielsen, 1994)

- # **É um tipo de avaliação por inspeção**
- # **Alternativa rápida e de baixo custo para encontrar problemas de usabilidade;**
- # **Poucos avaliadores (3 a 5) examinam a interface e julgam sua conformidade com um determinado conjunto de princípios (“heurísticas”) de usabilidade.**

Avaliação Heurística: o método

1. Avaliadores selecionam as tarefas (e telas correspondentes) a serem avaliadas

2. Avaliador examina uma tela buscando identificar violações das heurísticas

3. Para esta tela, avaliador examina os elementos buscando identificar violações da heurísticas

4. Avaliador lista os problemas, indicando local e gravidade, e recomenda soluções

5. Avaliador distribui problemas encontrados para outros avaliadores

6. Cada avaliador examina a lista de problemas identificado pelos outros, julgando cada problema/solução

7. Avaliadores geram juntos um relatório unificado, incluindo-se as propostas de solução para cada problema

Avaliação Heurística: as heurísticas

1. Visibilidade do estado do sistema

O sistema deve sempre manter os usuários informados sobre o que está acontecendo através de feedback adequado e no tempo certo.

Avaliação Heurística: as heurísticas

2. Correspondência entre o sistema e o mundo real.

O sistema deve falar a língua do usuário, com palavras, expressões e conceitos que lhe são familiares, em vez de utilizar termos orientados ao sistema. O projetista deve seguir as convenções do mundo real, fazendo com que a informação apareça em uma ordem natural e lógica.

Avaliação Heurística: as heurísticas

3. Reconhecimento em vez de memorização

O projetista deve tornar os objetos, ações e opções visíveis. O usuário não deve ter que se lembrar para que serve um elemento de interface cujo símbolo não é reconhecido diretamente; da mesma maneira, o usuário não deve ter que se lembrar de informação de uma parte da aplicação quando tiver passado para uma outra parte da aplicação. As instruções de uso do sistema devem estar visíveis ou facilmente acessíveis sempre que necessário.

Avaliação Heurística: as heurísticas

4. Controle e liberdade do usuário

Os usuários freqüentemente escolhem funções do sistema por engano e precisam de uma “saída de emergência” claramente marcada para sair do estado indesejado sem ter que percorrer um diálogo extenso. A interface deve permitir que o usuário desfça ou refaça suas ações.

Avaliação Heurística: as heurísticas

5. Consistência e padronização

Os usuários não devem ter que se perguntar se palavras, situações ou ações diferentes significam a mesma coisa. O projetista deve seguir as convenções da plataforma ou ambiente.

Avaliação Heurística: as heurísticas

6. Flexibilidade e eficiência de uso

Aceleradores podem tornar a interação do usuário mais rápida e eficiente, permitindo que o sistema consiga servir igualmente bem os usuários experientes e inexperientes. Aceleradores comuns são itens em barras ou paletas de ferramentas, bem como teclas de atalho para itens de menu e acionamento de botões. Além disto, o projetista pode prover mecanismos a serem utilizados pelos usuários para customizar ações freqüentes.

Avaliação Heurística: as heurísticas

7. Prevenção de erros

Melhor do que uma boa mensagem de erro é um projeto cuidadoso que evite que um problema ocorra, caso isto seja possível.

Avaliação Heurística: as heurísticas

8. Ajude os usuários a reconhecerem, diagnosticarem e se recuperarem de erros

As mensagens de erro devem ser expressas em linguagem simples (sem códigos), indicar precisamente o problema e sugerir uma solução de forma construtiva.

Avaliação Heurística: as heurísticas

9. Ajuda e documentação

Informações documentais devem ser facilmente encontradas, focadas na tarefa do usuário, enumerar passos concretos a serem realizados, e não serem muito extensas.

Avaliação Heurística: as regras

10. Projeto estético e minimalista

Os diálogos não devem conter informação que seja irrelevante ou raramente necessária. Cada unidade extra de informação em um diálogo compete com as unidades relevantes de informação e reduz sua visibilidade relativa.

Avaliação Heurística

- # Recomenda-se que o avaliador percorra a interface pelo menos duas vezes: uma para ganhar uma visão de conjunto e outra para examinar cuidadosamente cada elemento.
- # O produto da avaliação: lista dos problemas de usabilidade encontrados.
- # Para cada elemento de interface: problema deve ser detalhado e justificado, fazendo alusão às heurísticas que foram violadas, bem como ter sua severidade avaliada.

Avaliação Heurística

- # A gravidade de um problema de usabilidade é uma combinação de três fatores (Nielsen 1994):
 - Frequência: É um problema comum ou raro?
 - Impacto do problema: Será fácil ou difícil para os usuários superarem o problema?
 - Persistência do problema: os usuários serão incomodados pelo problema repetidas vezes?

Avaliação Heurística

- ✦ Nielsen sugere a seguinte escala de severidade:
 - 0 = não concordo que isto seja um problema
 - 1 = problema cosmético – não precisa ser consertado a menos que haja tempo no cronograma do projeto
 - 2 = problema pequeno – o conserto deste problema deve receber baixa prioridade
 - 3 = problema grande – importante de ser consertado; deve receber alta prioridade
 - 4 = problema catastrófico – é imperativo consertar antes de se lançar o produto

Avaliação Heurística

- # **Relatório final: pode conter, por exemplo, os seguintes itens (Prates e Barbosa 2003):**
 - Problemas esperados (e possíveis consertos)
 - O quão bem o sistema apóia as tarefas dos usuários
 - Caminhos de interação primários (importantes e/ou freqüentes)
 - Caminhos de interação alternativos ou pouco utilizados
 - Consistência
 - Elementos de estilo
 - Recomendações de projeto

Exemplo – (Barbosa e da Silva, 2010)

Considere o seguinte fragmento de tela de login de um Web site de livraria:¹

Login

Para fazer o login digite seu email ou seu CPF /CNPJ (caso tenha se cadastrado como Pessoa Física ou Jurídica, respectivamente) depois digite sua Senha e clique em Confirmar.

Email:

ou CPF/CNPJ:

Senha:

Exemplo

Visibilidade do estado do sistema, prevenção de erros. O elemento secundário *Cadastre-se* tem mais destaque do que o elemento *Confirmar*. Isso pode levar o usuário a acionar o botão errado ou se perguntar se entrou corretamente na tela de login, e até mesmo voltar para a página anterior e repetir a operação de acesso a essa página.

- Local: abaixo do formulário, apenas nessa tela.
- Severidade: 3 (problema grande), pois o usuário pode acreditar que precisa se cadastrar a cada compra, ou que o sistema está com defeito, e com isso pode desistir de efetuar a compra através desse site.
- Recomendação: destacar o botão primário (*Confirmar*) e reduzir a ênfase dos botões secundários (*Cadastre-se* e *Esqueci senha*). Considere modificar os botões secundários para links, mais afastados do botão primário do formulário.

Exemplo

Controle e liberdade do usuário. Os usuários não têm a opção, através do Web site, de voltar à página anterior. Para isso, precisam utilizar o botão de volta do próprio navegador.

- Local: ausência de um botão de volta em todos os formulários do site.
- Severidade: 2 (problema pequeno). O usuário está acostumado a utilizar o botão de volta do navegador em outros sites, e perceberá que pode fazer isso sem perder o que tenha feito no site (e.g., itens colocados no carrinho de compras).
- Recomendação: incluir um botão *Voltar* como botão secundário do formulário.

Exemplo

Consistência e padronização, prevenção de erros. Os campos de preenchimento alternativo ("Email:" e "ou CPF/CNPJ:") não estão claramente marcados, como de costume, por botões de opção (*radio buttons*). Como os usuários costumam seguir dicas visuais melhor do que instruções textuais, muitos preencherão os dois campos.

- Local: formulário de login, campos "Email:" e "ou CPF/CNPJ:".
- Severidade: 2 (problema pequeno). Apesar de ineficiente, o preenchimento dos dois campos não impede o usuário de efetuar o login.
- Recomendação: identificar os campos alternativos por botões de opção, que devem ser automaticamente selecionados quando o usuário inicia a digitação no campo correspondente.

Exemplo

Flexibilidade e eficiência de uso, consistência e padronização. O usuário não tem a opção de pedir para o sistema se lembrar do seu e-mail ou mesmo manter seu login ativo, como ocorre em boa parte dos sites de comércio eletrônico.

- Local: formulário de login, ausência de botões de seleção (*checkboxes*).
- Severidade: 2 (problema pequeno) para usuários ocasionais; 3 (problema grande) para usuários frequentes, que provavelmente darão preferência a Web sites que se lembrem “deles”.
- Recomendação: oferecer um checkbox *Lembrar dos meus dados* e/ou um checkbox *Manter meu login ativo por 15 dias*.

Ferramenta UX Check

The image shows a web browser window with the 'UX Check' tool overlay. The tool's sidebar on the left lists several heuristics: 'Visibility of system status', 'Match between system and the real world', 'User control and freedom', 'Consistency and standards', 'Error prevention', 'Recognition rather than recall', and 'Flexibility and efficiency of use'. Each heuristic has a brief description. The main browser window displays the Google homepage with search results for 'SIPPA - Sistema de Pr...'. A modal window titled 'Add a heuristic' is open, showing a dropdown menu with 'Visibility of system status' selected, a text area for 'Notes', a text area for 'Recommendation', and a dropdown for 'Severity' set to '0'. The modal also has 'Cancel' and 'Save' buttons.

UX CHECK

- Pause selection
- View progress
- Settings
- Stop evaluation

Visibility of system status
The system should keep users informed through appropriate feedback within reasonable time

Match between system and the real world
The system should speak the users' language rather than system-oriented terms. Follow real-world conventions

User control and freedom
Users often make mistakes and need 'emergency exits' to leave the unwanted state. Support undo and redo

Consistency and standards
Users shouldn't have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions

Error prevention
Prevent problems from occurring in the first place, or check for them and present users with a confirmation option before they commit to the action

Recognition rather than recall
Minimize memory load by making objects, actions, and options visible. Instructions should be visible or easily retrievable

Flexibility and efficiency of use
Accelerators - unseen by the novice user - may often speed up the interaction for the expert user. Allow users to tailor frequent actions

Pesquisar no Google ou digitar URL

Facebook SIPPA - Sistema de Pr... Portal da UF... Diário do Nordeste - lrio O POVO ONLINE | Po Index of /-cris

Add a heuristic
Visibility of system status

Notes

Recommendation

Severity 0

Cancel Save

Exercício



Fazer avaliação heurística de um serviço do governo.