

1. Avaliação heurística

A avaliação heurística é um método tradicional de avaliação de usabilidade. O método foi utilizado pela primeira vez em um interface Web em 1994, num estudo para o web site da Sun Microsystems (Nielsen e Sano, 1995). Este método foi desenvolvido por Nielsen e Molich (Nielsen, 1993) e consiste da inspeção sistemática da interface do usuário com relação à sua usabilidade. Basicamente um avaliador interage com a interface e julga a sua adequação comparando-a com princípios de usabilidade reconhecidos, as heurísticas. Nielsen sugere um conjunto com apenas 10 regras heurísticas para guiar a avaliação:

Diálogos Simples e Naturais: as interfaces de usuários devem ser o mais simples possível. Interfaces devem combinar as tarefas do usuário de tal forma que o mapeamento entre os conceitos computacionais e os do usuário seja simples. Deve-se apresentar exatamente a informação que o usuário precisa - nem mais nem menos - na hora e lugar exatos onde é necessária. Informação que será usada em conjunto deve ser exibida em conjunto, ao menos na mesma tela. Tanto os objetos de informação quanto as operações devem ser acessados em uma seqüência que casa com o modo como os usuários irão realizar suas tarefas efetiva e produtivamente. Muitas vezes tais seqüências são forçadas pela interface, mas normalmente é melhor permitir que o usuário controle o diálogo o máximo possível, de tal forma que a seqüência possa se ajustar às preferências do usuário individual.

Falar a Linguagem do Usuário: a terminologia da interface deve ser baseada na linguagem do usuário, e não orientada ao sistema. Para tanto, deve-se verificar quais termos são utilizados com maior freqüência pelos usuários. As informações também devem ser organizadas conforme o modelo mental que o usuário possui do domínio.

Minimizar a Sobrecarga de Memória do Usuário: o software deve exibir elementos de diálogo para o usuário e permitir que o mesmo faça suas escolhas, sem a necessidade de lembrar deste ou daquele comando específico. Para facilitar a utilização da interface, deve ser apresentado ao usuário um pequeno número de regras que se aplicam por toda a interface. Se o número de regras é grande o usuário terá de aprender/lembrar todas as regras, o que pode não ser tão simples. Por outro lado, se o software não tiver regra alguma, então o usuário deverá lembrar de cada elemento de diálogo. O uso de comandos genéricos é uma maneira de se ter um pequeno conjunto de regras. Comandos genéricos fazem com que coisas similares ocorram em diferentes circunstâncias, sendo suficiente ao usuário aprender poucos comandos para trabalhar com vários tipos de dados.

Consistência: consistência é um dos princípios básicos de usabilidade. Se os usuários souberem que um mesmo comando ou uma mesma ação terá sempre o mesmo efeito, eles ficarão mais confiantes no uso do software, e serão encorajados a fazerem novas descobertas. A mesma operação deverá ser apresentada na mesma localização em todas as telas e deverá ser formatada da mesma maneira para facilitar o reconhecimento.

Feedback: O sistema deverá informar continuamente ao usuário sobre o que ele está fazendo. O tempo de resposta influi no tipo de feedback que deve ser dado ao usuário. Um décimo de segundo (0, 1s) é o limite para o usuário pensar que o sistema está reagindo instantaneamente, o que significa que nenhum feedback especial é necessário; um segundo (1,0s) é o limite para que o fluxo de pensamento do usuário não seja interrompido, mesmo que o usuário perceba uma certa demora; e dez segundos (10s) é o limite para manter a atenção do usuário focalizada no diálogo. As vezes feedbacks especiais são necessários para contextualizar uma navegação mais demorada do usuário.

Saídas Claramente Marcadas: De modo a fazer com que o usuário sinta que pode

controlar o software, deverá ser fácil sair das situações mais variadas possíveis. Por exemplo, todas as caixas de diálogo devem possuir um botão de cancelar para trazer o usuário para a situação anterior. Muitas vezes, as saídas podem ser fornecidas por meio de uma facilidade de desfazer (undo) a última operação para retornar ao estado anterior. Os usuários rapidamente aprendem a confiar neste mecanismo, logo, deve estar disponível como um comando genérico por todo o software. Neste caso, o usuário poderá confiar no aprendizado por exploração, pois saberá desfazer eventuais erros.

Atalhos: embora deva ser possível operar a interface conhecendo-se apenas algumas regras gerais, deveria também ser possível para o usuário experiente executar mais rapidamente operações freqüentemente utilizadas, através de atalhos. Aceleradores típicos incluem abreviações, teclas de função, dique duplo do mouse, ou botões especiais para funções freqüentes. Também podem ser apresentados através da exibição dos últimos comandos executados, ou da função de volta (backtrack) em sistemas de hipertexto. Atalhos são também necessários quando por uma política de uma empresa ou organização a informação que se encontra em uma maior profundidade da árvore navegacional tenha que ser recuperada diretamente pela interface principal. Por exemplo, na página do Instituto de Informática da UFRGS (<http://www.inf.ufrgs.br>) na época de publicar o resultado do processo de seleção do Pós-Graduação em Computação deveria ter um elo na primeira página em vez de obrigar ao usuário a navegação pelo caminho cursos - pós-graduação - resultado seleção, muitas vezes não muito fácil de deduzir.

Boas mensagens de erro: as mensagens de erro devem seguir algumas regras: linguagem clara e sem códigos. Devem ser precisas. Devem ajudar o usuário a resolver o problema. Não devem intimidar ou culpar o usuário.

Prevenir Erros: melhor do que possuir boas mensagens, é evitar situações de erro. Conhecendo-se as situações que mais provocam erro, sempre é possível modificar a interface a tornar muito difícil que este erro ocorra.

Ajuda e Documentação: o melhor é se ter um software que seja tão fácil de usar que não necessite de ajuda ou documentação para complementar a interface do usuário. Além disto, sabidamente usuários raramente lêem a documentação

Como certamente um só avaliador não irá encontrar todos os problemas de uma interface, idealmente são utilizados vários. Nielsen sugere que a melhor relação custo/benefício é alcançada quando se utilizam entre 3 e 5 avaliadores. Cada avaliador deve realizar a sua inspeção individualmente e somente depois de todas avaliações terem sido concluídas, os avaliadores podem se comunicar. Este cuidado é importante para garantir avaliações independentes e sem influências. Os resultados tanto podem ser registrados por cada avaliador como por um observador presente durante as sessões, onde os avaliadores verbalizam seus comentários. Ao final de todas as sessões o observador deverá reunir todas as avaliações feitas em um único documento. Além disto, o observador poderá auxiliar o avaliador em caso de problemas com o protótipo, se este for o caso.

Tipicamente, uma sessão de avaliação heurística dura entre uma e duas horas. O resultado da avaliação é uma lista de problemas de usabilidade, indicando qual ou quais princípios foram violados e a gravidade do problema. Na tabela 4.1 são apresentados alguns dos problemas encontrados durante a avaliação do CD-ROM Literatura Gaúcha (veja figura 4.1), assim como comentários que ilustram como os avaliadores utilizaram as heurísticas para identificar problemas.

Descrição do Problema	Severidade [1-3]	Heurística violada e Comentários
-----------------------	------------------	----------------------------------

<p>Na pagina "Eriço Veríssimo" há a opção de visualizar um vídeo sobre o autor, que é disponível pelo ícone de uma câmera filmadora (figura autor) . Ao iniciar o vídeo não há como interrompê-lo.</p>	<p>3</p>	<p><u>Fornecer saídas claramente marcadas e prevenir erros:</u></p> <p>os avaliadores consideraram o problema muito grave (5) porque ele obriga o usuário a ver todo o vídeo mesmo que a sua ativação tenha sido ocasionada por um 'click' acidental. Esta é uma situação de erro que poderia ser minimizada com a opção para interromper o vídeo. Observa-se aqui um problema que foi classificado por mais de uma heurística, o que é aceitável na identificação de problemas complexos como este.</p>
<p>Na obra "O tempo e o Vento" o botão [crítica] (base da figura Obra) é disponível mas, uma vez selecionado, não mostra o texto correspondente à crítica do livro, como esperado.</p>	<p>2</p>	<p><u>Fornecer Mensagens de Retomo Adequada:</u> Sabendo que mensagens adequadas são necessárias para a eficiente utilização da interface, o avaliador inspeciona a interface em busca de todas as situações onde boas mensagens de retorno não são fornecidas. Assim, o avaliador identificou este problema e atribuiu severidade moderada (3) supondo que os usuários ficariam desorientado sobre a função de crítica.</p>
<p>Não existem opções de atalho.</p>	<p>1</p>	<p><u>Fornecer atalhos:</u> o avaliador usou esta heurística para procurar situações que necessitassem de atalhos. Contudo, foi considerado que atalhos não são necessários nesta aplicação e por isto uma severidade baixa (1) foi atribuída. Esta heurística lembra o avaliador de considerar as situações onde atalhos são necessários.</p>

TABELA 4.1 Exemplos de Problemas de Usabilidade.

O método de avaliação heurística pode ser empregado sem modificações em projetos Web. Contudo, recentemente Nielsen e outros autores (Bailey, 1999) tem sugerido outros conjuntos de regras que podem ser utilizados em substituição das heurísticas clássicas de Nielsen. A avaliação heurística é

um método simples, cuja eficiência reside na capacidade dos avaliadores de reconhecer problemas de usabilidade. Em principio, qualquer pessoa pode ser treinada para a aplicação deste método, embora melhores resultados sejam obtidos com avaliadores experientes. Outra vantagem deste método é que ele pode ser aplicado em qualquer etapa do desenvolvimento, mesmo sobre protótipos em papel.