

Usabilidade em interfaces

A USABILIDADE É UM CRITÉRIO DE QUALIDADE DE SOFTWARE E, PORTANTO, TAMBÉM É CONSIDERADA UM CRITÉRIO DE QUALIDADE DE INTERFACES, AFINAL INTERFACES FAZEM PARTE DE SOFTWARES. NA VERDADE, A USABILIDADE DAS INTERFACES PODE INFLUENCIAR MUITO NO PROCESSO DE AVALIAÇÃO DE UM SISTEMA. VEREMOS NESTE TÓPICO QUAIS SÃO OS PRINCIPAIS ASPECTOS DE USABILIDADE EM SISTEMAS COMPUTACIONAIS, MAS VOLTADOS PARA O DESIGN DE INTERFACES, OU SEJA, A RELAÇÃO ENTRE USABILIDADE E DESIGN DE INTERFACES.

AUTOR(A): PROF. THIAGO GRAZIANI TRAUE

O que é usabilidade, de fato?

A usabilidade é uma característica de qualidade de software, definida por milhares de autores, normas internacionais, manuais etc., de formas diferentes; contudo todos querem dizer a mesma coisa, ao final. Usabilidade é a característica que diz se o sistema pode ou não ser usado por qualquer um e quão fácil é encontrar as coisas dentro do sistema, portanto, permitir o usuário a atingir suas metas e necessidades. É muito importante não confundir usabilidade com utilidade. A utilidade é a característica que nos permite verificar se o sistema é útil para seu usuário, ou seja, se permite o usuário de divertir com um jogo ou pagar suas contas de forma efetiva, quando estamos falando de sistemas de internet banking (Rocha, H. V.).

Um ponto importante a ser considerado aqui é que o processo de criação de boas interfaces e com excelente usabilidade é conseguida com o trabalho de vários profissionais, trabalhando em equipe e com um planejamento preciso e bem gerenciado.



Legenda: O PROCESSO DE CRIAÇÃO E GARANTIA DE USABILIDADE é UM TRABALHO DE VÁRIOS
PROFISSIONAIS.

Usabilidade de Nielsen

Em 1993 Jakob Nielsen, o pai da usabilidade em interfaces de sistemas computacionais, definiu sete "slogans" de usabilidade, descritos por Rocha H. V.

Mas 1993 é muito distante, não há nada mais novo? Calma. Como dito, Nielsen é o pai da usabilidade em interfaces e tudo que foi publicado em seguida é baseado em suas ideias, ou seja, vamos direto à fonte e usar referências consagradas.

Estes slogans referem-se a como devemos tratar a desenvolvimento de interfaces a fim de garantir a usabilidade e a aceitabilidade. Não se assuste com os slogans, por mais "grosseiros" que possam ser, eles servem para instigar os desenvolvedores de interfaces a criar, cada vez mais, um trabalho melhor.



Legenda: ESTE é JAKOB NIELSEN, PAI DA USABILIDADE EM INTERFACES E ATUAL DIRETOR NA NORMAN NIELSEN GROUP.

Sua melhor tentativa não é boa o suficiente

Para Nielsen, é impossível criar boas interfaces a partir de uma única ideia e excluindo o usuário do processo. Os usuários são seres humanos (como você) e têm uma criatividade enorme para interpretar elementos da interface de formas diferentes. Isso quer dizer que podemos ter ideias maravilhosas para

interfaces, mas sempre haverá um usuário que não entenderá suas telas. Uma boa alternativa a isso é segundo Nielsen, utilizar métodos rigorosos de avaliação de interfaces, mas isso é assunto para outro momento.

O usuário está sempre certo

Se um usuário possui dificuldades quaisquer em acessar um determinado conteúdo ou realizar alguma tarefa no sistema, ou seja, algum problema de interação, não podemos jogar a culpa no usuário e achar que uma hora ele irá conseguir ou aprender. Em outras palavras não devemos ser prepotentes em relação a nossas interfaces. Devemos ser humildes no desenvolvimento de interfaces e aceitar que uma dificuldade do usuário pode ser, na verdade, uma falha em nossa interface.

Por exemplo, se identificamos que os usuários possuem dificuldade numa determinada área do sistema, não adianta colocarmos avisos em vermelho piscante; devemos refazer as interfaces dessa funcionalidade a fim de direcionar o usuário em seu objetivo.

O usuário não está sempre certo

Pois é, embora contradizente ao slogan anterior, temos que tomar cuidado ao sempre atender as necessidades dos usuários, através de suas dificuldades, ou a partir de sugestões dos usuários. Muitas vezes os usuários sabem desenvolver seu trabalho, mas não em um computador, ou seja, "os usuários nem sempre sabem o que é bom para eles" (Rocha, H. V.), pois tendemos a estranhar inovações em nosso processo mental de solução de problemas.

Usuários não são designers

Seria ótimo se pudéssemos colocar o usuário para trabalhar para nós e fazer todo o trabalho difícil, como o design, por exemplo. Ou, ainda, criarmos interfaces que podem ser altamente personalizadas onde cada usuário coloca os componentes onde quer, certo? Isso não é tão simples de resolver. É importante lembrarmos sempre que os usuários não são designer, ou seja, podem não tomar as melhores decisões de design e acabar se perdendo no próprio trabalho.

Para Rocha, H. V., *et.al.*, usuários novatos não costumam personalizar suas interfaces e preferem trabalhar como elas são originalmente, mesmo se essa opção está disponível. É importante sabermos que a customização por parte do usuário só é útil se produzir um design coerente com o modelo do processo de trabalho do usuário e que seja tão eficiente quanto o design original.

Designers não são usuários

Nós designers entendemos de design e não do processo de trabalho de nossos usuários. Quem entender do processo de resolver um problema é o usuário e não o designer.

Isso quer dizer que designer não são usuários, simplesmente porque não são! Desenvolver sistemas é uma arte que possui sua própria complexidade, por isso requer-se profissionais preparados para isso, como os designers. Por isso, não devemos excluir completamente os usuários do processo de desenho mas, para não colocarmos o usuário para trabalhar para nós (o que pode ser ruim como visto no slogan anterior), não podemos dar todo poder de decisão aos usuários.

Menos é mais

Esse talvez seja um dos principais slogans de usabilidade de Nielsen e quem tem se tornado tendência mundial no desenvolvimento de sistemas. Menos é sempre mais!

Isso quer dizer que quanto mais elementos colocamos em nossas interfaces mais complexas as tornamos.

Mas se nosso software é cheio de funcionalidades, não devemos colocá-las disposição dos usuários? Sim, devemos é claro, mas não tudo na mesma tela. Precisamos dividir as coisas, criar quantas interfaces forem precisas para sempre ajudar os usuários, sem sobrecarregá-lo com informações, ícones, cores, mensagens piscando etc., em uma única tela. Cada tela deve ter sua própria responsabilidade. Lembre-se: Em design de interfaces menos é sempre mais!

Ajuda que não ajuda

Quantas vezes você já usou o *help* (ajuda) de algum software ou sistema e não conseguiu encontrar o que estava procurando ou, pior, encontrou e não entendeu e ficou frustrado? Bem isso ocorre com mais frequência do que você imagina.

O material de ajuda dos sistemas precisa ser claro, objetivo e conciso. Não podemos o ignorar e o fazer de qualquer jeito. É importante lembrar que se o usuário está recorrendo ao sistema de ajuda de sua aplicação, então pode ser que algo esteja errado com o design. Sistemas realmente bem feitos possuem o material de ajuda que raramente são acessados, pois as interfaces não apresentam qualquer dificuldade de uso pelo usuário, seja ele leigo ou não.

Atributos de usabilidade de Nielsen

Baseados em seus slogans de usabilidade, Nielsen publicou em 1993 cinco atributos essenciais que explicam a usabilidade de interfaces de sistemas computacionais e podem (devem) servir de guia para o design.

Learnability (habilidade de aprendizagem)

Não existe tradução para o português de "*learnability*" (o que seria mais ou menos algo parecido com "aprendabilidade"), por isso usamos o termo habilidade de aprendizagem.

Para Nielsen, esse é o atributo de usabilidade de interfaces mais importante, pois definirá como o usuário vê seu sistema até o final de sua vida.

O sistema precisa promover facilidade de aprendizagem aos usuários, ou seja, logo na primeira vez que o usuário (seja ele avançado ou não) tem contato com o sistema, ele deve conseguir a interagir, rapidamente.

É claro que o usuário não aprende a utilização de um sistema sem antes vê-lo ou interagir com ele. Isso ocorre com o tempo de uso. A aprendizagem de um sistema através de suas interfaces é algo temporal. Quanto menor o tempo de aprendizagem, melhor são as interfaces.

Eficiência

Os usuários precisam desempenhar suas tarefas de maneira eficiente, ou seja, rápida e com segurança, especialmente depois de aprender a utilizar o sistema.

Isso quer dizer não podemos tornar complexas as atividades dos usuários experientes, ou seja, uma vez que o usuário tenha domínio do sistema, ele não precisará entrar em diversas telas para resolver suas metas e necessidades.

Memorability - Facilidade de lembrança

Este é um outro termo que não possui tradução para o português; seria o equivalente a "memoriabilidade", mas este termo não existe, portanto, usaremos a expressão "facilidade de lembrança". Depois de aprendido, o sistema não pode ser esquecido, ou seja, depois do usuário atingir um certo nível de conhecimento do sistema, ele não poderá esquecer de como operá-lo, mesmo se ficar um bom tempo sem utilizá-lo.

Este atributo está fortemente ligado a habilidade de aprendizagem. Se o sistema é fácil de aprender, então provavelmente ele será fácil de lembrar.

Prevenção de erros

Um erro é quando a ação solicitada desvia da ação pretendida, ou seja o usuário não consegue fazer o que queria. É quase óbvio este atributo de usabilidade, mas nem sempre atendido. Temos de minimizar ao máximo os erros e evitar a perda de trabalho dos usuários e isso é uma tarefa que se consegue com uma boa relação entre programação e interfaces eficientes.

Satisfação subjetiva

Ao utilizar o sistema, seus usuários precisam ficar satisfeitos ou, em outras palavras os usuários precisam gostar do sistema.

Este atributo é o mais importante quando estamos desenvolvendo jogos, pois o objetivo principal da aplicação é o entretenimento.

Contudo, trabalhar com aplicações corporativas, por exemplo, não precisa ser tedioso. Claro, não vamos transformar aplicações corporativas em jogos, mas as interfaces precisam promover a satisfação aos usuários.

Este atributo de usabilidade também é importante para sistemas cujos usuários passam boa parte do seu dia operando como, por exemplo, pontos de venda, sistemas gerenciadores de clínicas médicas etc.

A Usabilidade de Steve Krug

Embora importante, Nielsen não é o único autor que fala sobre usabilidade para design de interfaces. Há um outro autor, também consagrado, chamado Steve Krug que também desenvolveu seu próprio modelo de usabilidade e tem sido bastante aceito.

Para Krug, a usabilidade em sistemas interativos precisa seguir algumas leis, bastantes simples de entender e interpretar.

A primeira lei, e mais importante delas, intitula um de seus livros mais importantes: “Não me faça pensar”.

É isso mesmo, para Krug, se seu sistema faz com que o usuário precise pensar, então há algo errado em seu sistema, ou seja, devemos falar a mesma língua do usuário.

Krug diz ainda que “nada importante deve estar a mais de dois cliques de distância”, ou seja, o usuário que não precisa pensar, rapidamente encontra o que procura em seu sistema ou site.

Os usuários recorrem aos computadores e sites pois possuem em sua mente duas coisas: metas e necessidades (MNs). Se o seu usuário atinge rapidamente suas MNs, então seu sistema possui as características de usabilidade esperadas.

Como visto aqui a usabilidade é um item fundamental para o desenvolvimento de boas aplicações e sua relação com as interfaces é intrínseca.

O usuário é e deve sempre ser o foco de nosso trabalho quando estamos atendendo a usabilidade que é o principal fator de aceitação de nossa aplicação. Sem usabilidade não há aceitação. Sem aceitação o usuário não utilizará a aplicação.

ATIVIDADE FINAL

Quem é Jakob Nielsen?

- A. Pai da engenharia de software
- B. Pai da usabilidade em interfaces do design de interfaces

- C. Pai do desing de interfaces, apenas
- D. Pai dos princípios de design

Os usuários precisam ficar satisfeitos e com uma ótima sensação ao utilizar o sistema. Estamos falando de...?

- A. Usabilidade
- B. Tratamento de erros
- C. Satisfação subjetiva
- D. Eficiência

Colocar tudo da interface poderia resolver o problema, mas pode sobrecarregar as telas. Precisamos colocar apenas o que é necessários. Estamos falando de qual princípio de usabilidade?

- A. Eficiência
- B. Usuário está sempre certo
- C. Designer não é usuário
- D. Menos é mais

REFERÊNCIA

Rocha, H. V. e Baranauskas, M. C. C., 2003, Design e Avaliação de Interfaces Humano-Computador, Instituto de Computação, Universidade Estadual de Campinas, UNICAMP

Krug, S., 2008, Não me faça pensar - Uma abordagem de bom senso à usabilidade na Web, 2ª Ed., Alta Books

Nielsen, J. (1993) Usability Engineering. Academic Press, Cambridge, M

Norman, D. A. (1988) The Psychology of Everyday Things. Basic Books, New York

