

Avaliação de Interfaces: Avaliação por check-list

A PARTIR DE AGORA COMEÇAREMOS A FAZER UMA COISA MUITO IMPORTANTE COM NOSSAS (E A DOS OUTROS) INTERFACES: AVALIAR. MAS AVALIAR O QUE? QUANDO? COMO? POR QUE? PERGUNTAS COMO ESSAS SERÃO RESPONDIDAS AQUI E FALAREMOS, NESTE TÓPICO, ESPECIALMENTE DO PROCESSO DE AVALIAÇÃO POR CHECK-LIST.

AUTOR(A): PROF. THIAGO GRAZIANI TRAUE

O que é avaliar?

Bem, vamos comparar as provas que você faz, regularmente na universidade. As provas são avaliações, certo? Além das provas, seus professores podem (e devem) aplicar diversas outras formas de avaliação: atividades, chamadas orais, apresentações de seminários, acompanhamento direto de sua evolução etc. Os professores e tutores avaliam o quanto você está evoluindo em determinado assunto, baseado na comparação dos seus resultados com os resultados esperados. E isso gera uma nota.

Bem, avaliar interfaces não é tão diferente. Precisamos "colocar na mesa" tudo que queremos que a interface faça (ou deveria fazer) e aplicar algum método de avaliação, seja ele qual for, para verificar se ela [a interface] sabe (permite) fazer o que deveria fazer.

Então avaliar uma interface é, nada mais nada menos, do que verificar se ela atende aos requisitos funcionais e de usabilidade da aplicação.

A avaliação não se limita ao software, mas pode envolver, também, o hardware, especialmente quando este é desenvolvido em seu projeto.

Lembre-se, também, que a avaliação não pode e não deve ser tratada como uma fase única do projeto, onde você avalia, faz o redesign necessário e pronto. Ela precisa acontecer em mais fases. Lembre-se, também, que a avaliação da interface não pode ocorrer apenas "se der tempo", ela precisa ocorrer.

Mas quando, exatamente? Bem, quando mais se avaliar melhor, ou seja, o tempo inteiro, em cada fase do processo, mas isso é uma utopia. Então Hix e Hartson recomendam que a avaliação ocorra em (independente da ordem):

- Na fase de especificação de requisitos;

- Na fase de prototipação;
- Na fase de implementação;
- Na fase de Design (e em todas as subfases envolvidas);
- Na fase de análise (e em todas as subfases envolvidas).

Para cada uma das fases, claro, temos de aplicar um método diferenciado. Não podemos testar código sem o código, por exemplo. Por isso, cada fase exige seu próprio processo de avaliação.

De qualquer forma, não podemos deixar de considerar que a avaliação é um processo importante e sempre possui um certo grau de incerteza, mesmo após avaliações formais exaustivas. De qualquer forma é melhor do que não avaliar. Temos de ficar bastante atentos a cada detalhe, pois são eles que farão toda a diferença na vida do usuário.



Legenda: TEMOS DE FICAR DE OLHO EM CADA DETALHE. OLHOS ABERTOS SEMPRE!

Objetivos da avaliação de interfaces

Fazemos avaliação para garantir que o que os usuários esperam seja atendido, ou seja, para garantir o alcance das metas e necessidades (MNs) dos usuários. Quando avaliamos tardiamente, ou seja, com o sistema implementado (ou quando avaliamos um sistema pronto que não necessariamente foi desenvolvido por nós), estamos conhecendo os problemas experimentados pelos utilizadores.

Portanto, podemos definir três grandes objetivos da avaliação de interfaces:

- **Avaliar as funcionalidades:** É importante avaliarmos as funcionalidades das interfaces para saber se estão de acordo com os requisitos do sistema e das tarefas do usuário, ou seja, é preciso avaliar se as interfaces permitem o usuário a fazer o que se espera que ele faça, de modo fácil, eficiente, seguro, agradável etc. Essa é uma das avaliações mais importantes, pois inclui quase todas as fases do processo.
- **Avaliar o impacto do design junto aos usuários:** É preciso avaliar a usabilidade do sistema, ou seja, aspectos como facilidade de aprendizagem, atitudes do usuário em relação às dificuldades ou facilidades da aplicação, sobrecargas mentais ou qualquer outro tipo de sobrecarga aos usuários, etc. Essa avaliação é muito importante pois visa garantir o principal princípio de usabilidade de Steve Krug: Não fazer o usuário pensar.
- **Identificação dos problemas específicos do design:** Precisamos identificar quais aspectos de nosso design causam confusão aos usuários ou não estão suficientemente claros. Claro, este objetivo é praticamente o resultado dos objetivos anteriores, mas precisamos conhecer cada detalhe e saber separar as dificuldades dos sucessos dos usuários, causados pela interface.

Baseado nisso, apresentaremos neste tópico um dos primeiros, mais simples (de preparar, utilizar etc.) métodos de avaliação. Avaliação por *check-list*.

O que é um check-list?

Você deve ter em seu celular (especialmente se for um *smartphone*) ou em seu computador algum aplicativo que lhe permite criar listas de tarefas.

Por exemplo, ao chegar em casa você se depara com a geladeira quase vazia. Você pode criar uma lista de itens que precisam ser comprados, como leite, pão, queijo, salada, tofu (espécie de queijo feito com leite de soja), abóbora etc. Você anota cada um destes itens. Vai até o supermercado e riscada cada item que encontrar da lista. Caso não encontre algum, o mantém na lista, certo? É simples!

Bem um *check-list* é exatamente isso: Uma lista de afazeres.

Contudo, aplicativos mais avançados permitem colocar prioridades num item de lista, por exemplo, a compra de leite pode ter prioridade sobre a compra do tofu, ou seja, você não pode voltar para casa sem o leite, mas não tem problema se não encontrar o tofu. Neste caso, o item "leite" tem um peso maior do que o item "tofu".

E não apenas prioridades para cada item, mas também podem ter um grau de solução, ou seja, um *ranking* de quanto aquele item foi ou não resolvido, para termos uma precisão ainda maior.

Por exemplo, usando o mesmo cenário do supermercado, imagine que você achou o leite para comprar, mas o vencimento está próximo. Você tinha de comprar leite e compra, mas não tantos quanto gostaria, ou seja, terá de procurar em outro mercado em outra oportunidade mais leite. Isso quer dizer que, se

considerarmos uma escala de zero a dez sobre a solução do problema, podemos dizer que o item leite foi "cinco", pois não foi totalmente atendido na lista, apenas parcialmente.

Resumindo, cada item da lista pode ter uma prioridade e um grau de solução, que você mesmo pode definir.



Legenda: ILUSTRAÇÃO DE UM CHECK-LIST

Avaliação por *check-lists*

Bem, uma vez sabendo o que é um *check-list*, podemos utilizá-lo para fazer a avaliação de nosso sistema ou site etc.

A avaliação por *check-list* é a mais simples de todas, pois quem define o que será avaliado é a equipe de avaliação, ou seja, o(s) avaliador(es) definem cada item da lista, com suas prioridades e, em seguida, procuram no sistema uma forma de resolver o determinado item e, ao conseguirem, marcam o item se foi resolvido totalmente, parcialmente ou com uma escala qualquer para tal.

O importante é não deixar passar nada, por isso, é muito importante que quem faça a avaliação não seja alguém que esteja envolvido com a programação ou com o design, para garantir que os vícios do desenvolvimento (quando estamos desenvolvendo vamos testando cada funcionalidade) afetem ou influenciem o teste.

Mas o que colocar na lista, ou seja, o que verificar? Bem, isso depende muito do tipo de sistema ou do público alvo. Um jogo possui itens de checagem diferentes de um sistema para um avião supersônico ou de uma torradeira. Cada projeto possui suas particularidades e, portanto, possui seu próprio conjunto de itens a serem checados.

Um jogo, por exemplo, possui objetivo de "entretenimento" e "diversão", então itens como "o jogo causa a sensação de bem-estar" e "o jogo prende a atenção do jogador" devem constar na lista, assim como quando estamos avaliando por *check-list* um microondas, temos itens como "o aparelho permite ajuste do relógio" e "o *timer* é preciso", ou ainda, "a comida aquece corretamente" etc. São sistemas diferentes, portanto listas diferentes.

De qualquer forma, quando estamos avaliando sistemas online como sites, por exemplo, podemos ter alguns itens padrões para checagem, como:

- Segurança (quando aplicável)
 - O sistema permite autenticação criptografada
 - O sistema permite que se faça logoff (sair do modo autenticado) facilmente
 - O sistema não permite que fiquem vestígios na realização do logoff
 - O sistema armazena cookies com tempo de vida
 - O sistema perde a sessão de autenticação ao ser fechado
 - A autenticação possui tempo de vida
 - O usuário sente-se seguro o tempo inteiro
 - Utiliza-se certificação digital entre as transações mais importantes
 - etc.
- Usabilidade
 - Todos os ícones são intuitivos
 - O posicionamento dos menus é consistente o tempo todo
 - O posicionamento dos elementos de cada interface é consistente o tempo inteiro
 - Há campos de busca
 - Há campos de busca avançada
 - O usuário consegue encontrar rapidamente o que está procurando
 - O usuário sabe o que fazer rapidamente em cada interface
 - O usuário sabe qual o real objetivo do sistema logo na primeira interface
 - Os erros permitem que os usuários os entendam
 - Os erros de uso são facilmente corrigíveis pelos próprios usuários
 - Existem mecanismos de contato
 - O sistema pode ser utilizado por crianças
 - O sistema pode ser utilizado por pessoas que não possuem conhecimento em Tecnologia
 - O tempo de resposta de cada funcionalidade é aceitável
 - etc.
- Interface
 - As interfaces são limpas
 - As interfaces são claras

- etc.
- Acessibilidade
 - O sistema permite acessibilidade
 - O sistema permite aumento do tamanho da fonte
 - O sistema permite diminuição do tamanho da fonte
 - O sistema possui algum mecanismo para alto contraste em todas as interfaces
 - É permitida regionalização (troca da linguagem) do sistema
 - Os textos são claros
 - Os botões parecem botões (que podem ser clicados)
 - Existem menus e submenus consistentes
 - Itens de navegação local e social aparecem no local correto
 - A publicidade não atrapalha a navegabilidade
 - A publicidade não atrapalha a leitura
 - O sistema pode ser utilizado por pessoas com deficiência motora
 - O sistema pode ser utilizado por pessoas com deficiência visual
 - O sistema pode ser utilizado por pessoas com deficiência auditiva
 - O sistema pode ser utilizado por pessoas com outro tipo de necessidade especial (especificar)
 - etc.
- Requisitos funcionais
 - {aqui depende de cada cenário, ou seja, cada sistema possui uma lista diferente de requisitos}
 - O usuário consegue atingir todas as metas
 - O usuário consegue realizar todas as atividades
 - etc.

Itens como os descritos acima podem ser facilmente aplicados em formato de *check-list*, mas não são o suficiente. É preciso realizar testes exaustivos com públicos diferenciados para garantir que todos os itens estão sendo atendidos.

Avaliando sistemas web

Além de utilizar os itens descritos acima, para avaliar a usabilidade de sites web você pode utilizar uma lista como a descrita abaixo, usando a mesma ideia descrita até então: Cada item tem uma prioridade e cada resultado pode ter sua própria escala de aceitação, como proposto abaixo como exemplo.

O que fazer com os resultados?

Antes de saber o que fazer com o resultado, é preciso sabermos quais resultados esperar. Bem, no caso de *check-lists*, os resultados são simples: Teremos uma relação de itens não marcados (itens que não estão sendo atendidos) e uma relação de itens marcados parcialmente.

Em outras palavras, podemos pegar o *check-list* e excluir todos os itens que estão sendo atendidos integralmente, ou seja, excluir todos os itens que não apenas estão checados, mas estão com o grau de atendimento máximo.

SEMPRE Há MAIS DE UM AVALIADOR NESTE MÉTODO

Lembre-se que a avaliação por *check-list* é, normalmente, realizada por mais de um avaliador e os resultados podem ser diferentes. Precisamos considerar cada resultado individualmente.

Feito isso, você terá em mãos uma poderosa lista para o redesign, ou seja, uma poderosa lista que lhe permitirá ver o que não está, de fato, sendo atendido. É preciso checar, então, item por item.

Por exemplo, digamos que um dos itens descritos acima não esteja sendo atendido integralmente como, por exemplo, "O tempo de resposta de cada funcionalidade é aceitável". É preciso, então, checar se o servidor de banco está bem configurado, o servidor web está respondendo corretamente, se os algoritmos utilizados não são complexos demais e precisam ser revistos ou, até mesmo, se o processo utilizado está atrapalhando o usuário mais do que ajudando. Lembre-se do que Steve Krug diz: "não me faça pensar".

Uma vez corrigidos cada um dos itens não marcados ou marcados como parciais, precisamos submeter o sistema para uma nova avaliação, considerando-se os mesmos itens. Neste novo processo, espera-se que os itens que não foram atendidos anteriormente passam a ser e que o redesign não tenha afetado nenhum item previamente marcado como concluído integralmente.

Este pode parecer um ciclo infinito, mas devemos submeter à reavaliação nossas interfaces quantas vezes forem necessárias, para garantir o sucesso de nossa aplicação. Esta é a essência do ciclo da avaliação de interfaces.



Legenda: CICLO DE DESIGN, AVALIAÇÃO E REDESIGN

Chegamos ao final de mais um importante tópico. Neste vimos a importância de se fazer a avaliação das interfaces e, especificamente neste caso, utilizando o método de *check-lists*, que são, essencialmente, listas de checagem com graus diferentes de urgência para cada item e que, ao serem submetidas à uma avaliação, nos promovem um relatório completo de item a item do que está ou não sendo atendido no sistema. Uma vez com os resultados em mãos, podemos rever o design e as funcionalidades.

ATIVIDADE FINAL

O que pode ter um item de uma lista de avaliação de interfaces?

- A. Pode ter uma prioridade (o quanto ele é necessário) e o grau de atendimento dele no sistema.
- B. Pode ter uma prioridade (o quanto ele é necessário), apenas.
- C. Pode ter o grau de atendimento dele no sistema, apenas.
- D. Apenas o item é necessário para se realizar uma avaliação, ou seja, não é preciso ter mais nada associado a ele.

O que fazer depois de ter o sistema checado por um check-list?

- A. Nada, apenas disponibilizá-lo aos usuários com a lista do que é ou não atendido.
- B. Corrigir cada item não atendido, ou seja, fazer o redesign necessários, quantas vezes forem precisas.
- C. Fazer um único ciclo de redesign e entregar o sistema
- D. Melhorar a documentação do sistema

Uma check-list genérica pode ser usada para se avaliar qualquer sistema.

Essa afirmação é verdadeira ou falsa e por que?

- A. Verdadeira, pois todo sistema permite interação.
- B. Falsa, pois todo sistema é diferente.
- C. Verdadeira, pois todo sistema é igual.
- D. Falsa, pois cada sistema é único e possui seu próprio conjunto de funcionalidades, público alvo, requisitos etc.

REFERÊNCIA

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

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

Ferreira, S. M., Nunes, R. R, 2008, e-Usabilidade, 1ª ed., Editora LTC

Benyon, D., Turner, P. & Turner, S., 2005, Designing Interactive Systems: People, Activities, Contexts, Technologies. Ed. Pearson

