

Avaliação Heurística

Nielsen (www.useit.com)



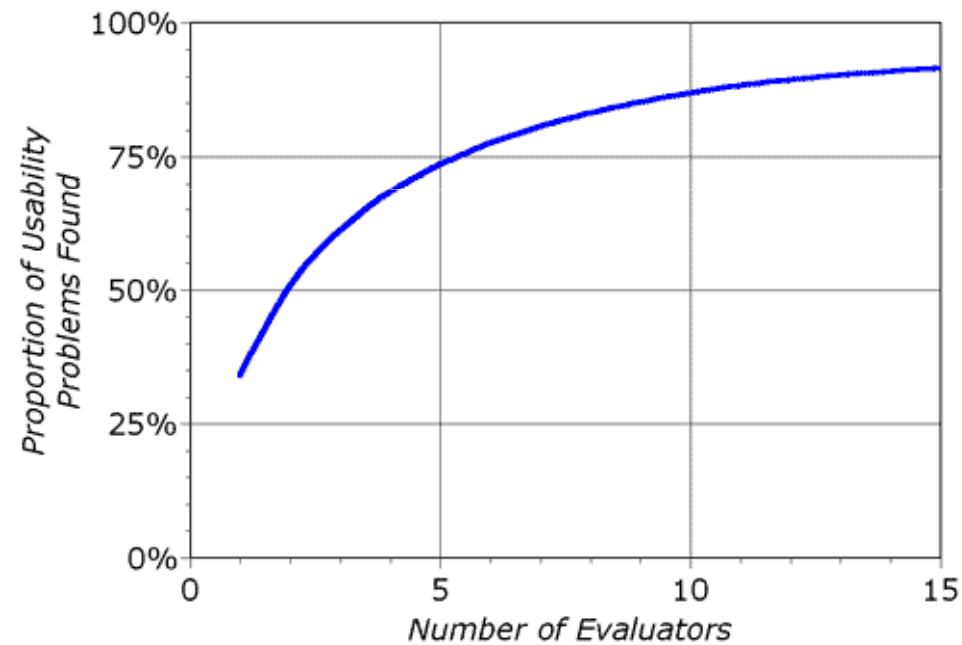
Procedimento

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

1. Obter entradas

- Obter Time de avaliação (treinado no método)
- Familiarizar os avaliadores com o domínio
- Fornecer Storyboards/scenários que mostram detalhes suficientes
- Selecionar Heurísticas a serem utilizadas!

Quantos avaliadores?



2. Realizar avaliação independente



- Julgar protótipo relativamente ao conjunto de heurísticas
- Quais heurísticas?
 - 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

3. Discussão/Coleta



- Organizar todos os problemas encontrados
- Filtrar quais são problemas importantes

Graduação da severidade

- Baseada na
 - Frequência
 - Impacto
 - Persistência
 - Impacto no mercado
- Escala de 0 a 4

Custo benefício

- Estudo de 1994 mostra economia de 48 vezes
- Outros dados variados

Jakob Nielsen

10 heurísticas de usabilidade

10 heurísticas genéricas

- 1 Visibilidade do status do sistema
- 2 Casamento entre o sistema e o mundo real
- 3 Controle e liberdade do usuário
- 4 Consistência e padrões
- 5 Prevenção de erros
- 6 Reconhecimento em vez de lembrança
- 7 Flexibilidade e eficiência de uso
- 8 Projeto estético e minimalista
- 9 Ajudar usuários a reconhecer, diagnosticar e recuperar de erros
- 10 Ajuda e documentação

Visibilidade, Modelo Conceitual/Mapeamento, Iniciativa do diálogo, Consistência...



- **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
- **2 Casamento entre o sistema e 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.
- **3 Controle e liberdade do usuário**
 - O sistema deve oferecer saídas claras para situações nas quais usuários se encontram por terem escolhido funções do sistema por: não deve ser necessário um diálogo extenso. O sistema deve suportar *undo* e *redo*.
- **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.

Permitir errar, reconhecer, escolher...



- **5 Prevenção de erros**
 - Melhor que boas mensagens de erro é preparar um projeto que impede os erros aconteçam
- **6 Reconhecimento em vez de lembrança**
 - 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.
- **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 programem ações frequentes.

Estética, minimalista, ajuda...



- **8 Projeto 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 unidade que são de fato relevantes e, assim, diminuem a sua visibilidade relativa.

- **9 Ajudar usuários a reconhecer, diagnosticar e recuperar de erros**

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

- **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 ser realizados, e não serem grandes.

Avaliação Heurística

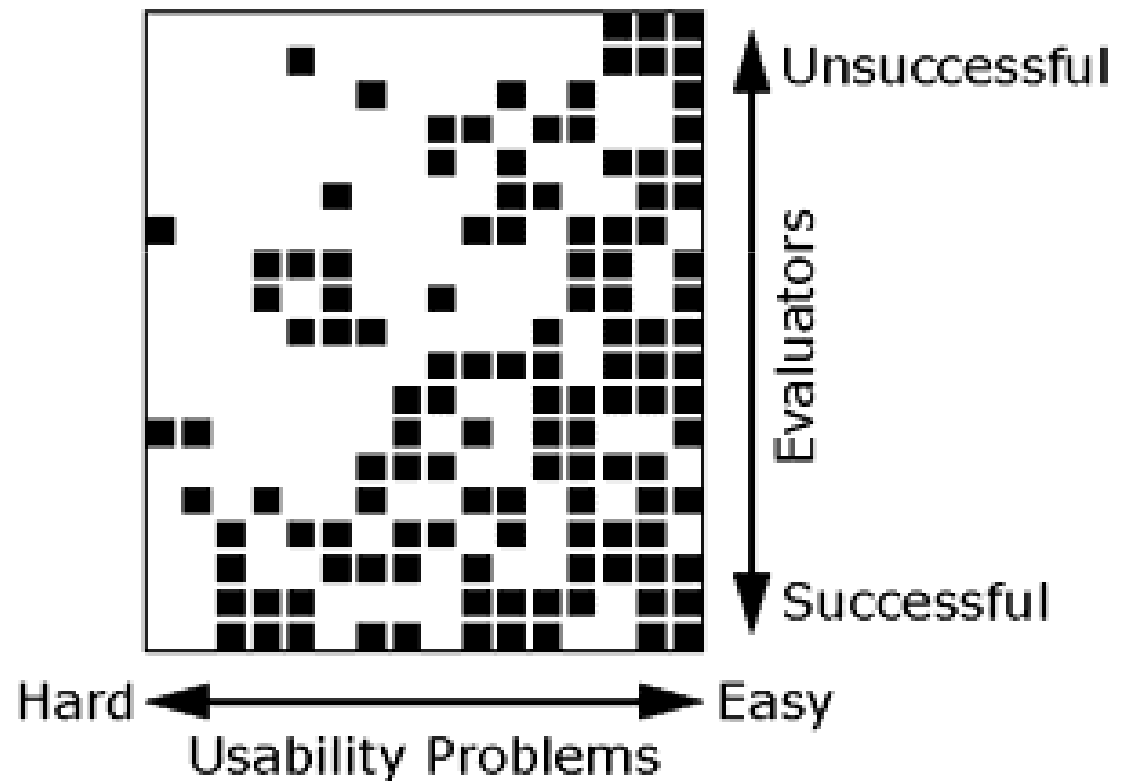
- Nielsen, 1993: *discount usability engineering*
 - http://www.useit.com/papers/querrilla_hci.html
- Métodos baratos, rápidos e fáceis de serem usados
- Avaliação heurística é o principal método
 - Fácil: pode ser ensinada em 4 hs.
 - Rápida: maioria das avaliações requer cerca de um dia
 - Barata: 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

- 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
- 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
- Heurísticas
 - Regras gerais que objetivam descrever propriedades comuns de interfaces usáveis

Avaliação Heurística

- Porque múltiplos avaliadores?
- Um avaliador não detecta todos os problemas
- Bons avaliadores acham problemas simples e problemas complexos



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
 - 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

Avaliação Heurística

- Sessões de avaliação individual
 - Avaliador pode ser deixado por conta própria...
 - Caso seja um perito no domínio do sistema, ...
 - ou 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
- Listas de problemas dos avaliadores consolidadas em uma única
- Atribuição de graus de severidade aos problemas
- Discussão com equipe de desenvolvimento
 - 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, Design e Avaliação de Interfaces Humano Computador, Cap. 4
 - Resumo dos resultados:
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)

Avaliação Heurística

- Graus de severidade do problema
 - Combinação de três 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 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)
- Pode ser usada mesmo por quem não tem muita experiência com avaliação
- Também pode ser usada para avaliar implementações