

INF1403 – Introdução a Interação Humano-Computador (IHC)

Turma 3WA

Professor: Alberto Barbosa Raposo

Avaliação Heurística

19/03/2012

O que é Avaliação Heurística?

- “Heurística” = baseada em um conhecimento prático (sem comprovação científica), que vem da experiência cotidiana continuada.
- TRATA-SE DE UM MÉTODO DE INSPEÇÃO
 - **O que é um método de inspeção?**
 - **Não envolve usuários.**
 - É uma **análise realizada por especialistas que advogam pelo usuário** – ou seja: sabendo os anseios e necessidades dos usuários, e conhecendo as técnicas possíveis de IHC, avaliam se determinado artefato computacional proporciona uma boa experiência para o usuário.

Jakob Nielsen: maior promotor do conhecimento heurístico em IHC

useit.com: usable information technology

useit.com: Jakob Nielsen's Website



Permanent Content

Alertbox

Jakob's column on Web usability

[Middle-Aged Users' Declining Web Performance Gap](#)

(March 31)

Between the ages of 25 and 60, people's ability to use websites declines by 0.8% per year — mostly because they spend more time per page, but also because of navigation difficulties.

[Designer-User Gap](#) (March 17)

[Company Name in Microcontent](#) (March 3)

[Application Design](#) (Feb. 19)

News

[Usability Week 2008](#) Conference

- > London, May 19-24
- > San Francisco, June 16-21
- > Melbourne, July 21-26

Due to increased demand and the raving feedback from the New York conference, we have added a session for [B2B user experience](#) to the San Francisco conference. (Also presented in London.)

BusinessWeek [Apple Goes Corporate](#)

O MÉTODO DE AVALIAÇÃO HEURÍSTICA

- <http://www.useit.com/papers/heuristic/>
 - O Método de Avaliação Heurística (mAH) foi **proposto por Jakob Nielsen** em 1994.
 - Trata-se de uma **inspeção guiada por heurísticas** – princípios gerais de bom design de interface, voltado para maximizar a usabilidade do artefato.
 - Tradicionalmente, utilizam-se 10 Heurísticas.
 - Elas têm sido alteradas e expandidas desde a sua proposta original, para cobrir novas tecnologias e ambientes computacionais.

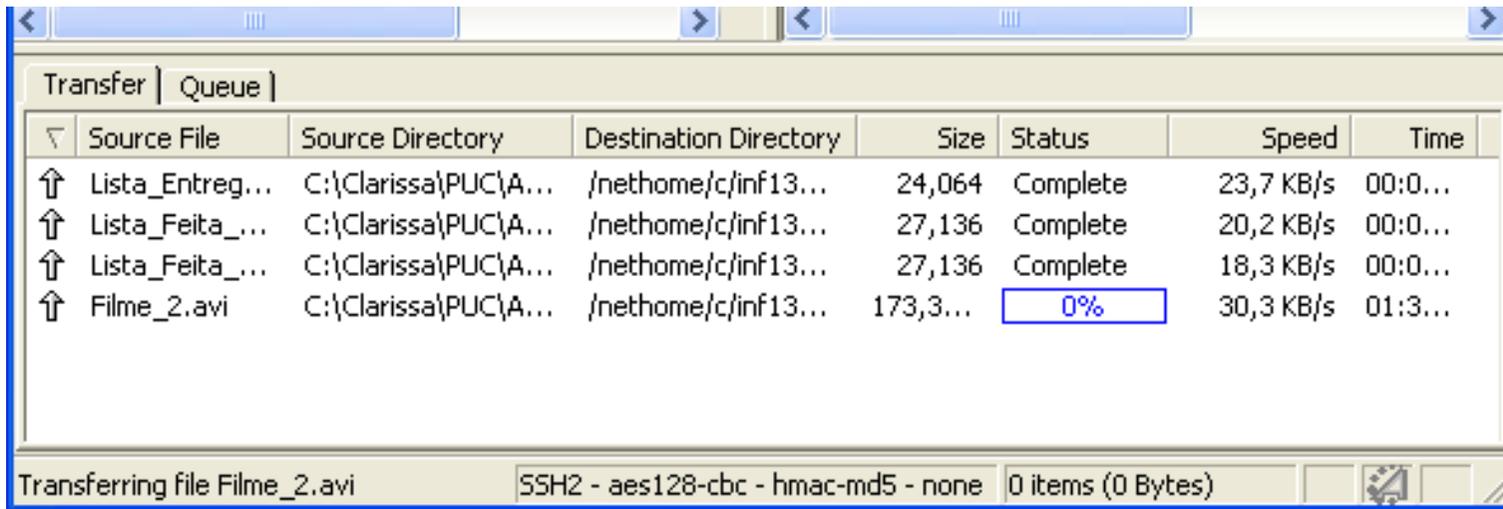
As 10 Heurísticas de Nielsen (1ª)

1. Visibilidade do Estado do Sistema

- Os usuários devem ser constantemente – *e sem demora* – informados sobre o estado em que o sistema está.
 - Informações claras, constantes e imediatas sobre o *estado do sistema*

Exemplo

- visibilidade do estado do sistema



| Source File | Source Directory | Destination Directory | Size | Status | Speed | Time |
|-------------------|----------------------|-----------------------|----------|----------|-----------|---------|
| ↑ Lista_Entreg... | C:\Clarissa\PUC\A... | /nethome/c/inf13... | 24,064 | Complete | 23,7 KB/s | 00:0... |
| ↑ Lista_Feita_... | C:\Clarissa\PUC\A... | /nethome/c/inf13... | 27,136 | Complete | 20,2 KB/s | 00:0... |
| ↑ Lista_Feita_... | C:\Clarissa\PUC\A... | /nethome/c/inf13... | 27,136 | Complete | 18,3 KB/s | 00:0... |
| ↑ Filme_2.avi | C:\Clarissa\PUC\A... | /nethome/c/inf13... | 173,3... | 0% | 30,3 KB/s | 01:3... |

Transferring file Filme_2.avi SSH2 - aes128-cbc - hmac-md5 - none 0 items (0 Bytes)

Heurística é respeitada.

Na avaliação heurística, identificamos apenas situações nas quais as heurísticas são violadas.

As 10 Heurísticas de Nielsen (2ª)

2. Correspondência entre Sistema e Mundo Real

- Os conceitos, termos, vocabulário, tarefas e procedimentos adotados na interface do sistema devem ser *os tão próximos quanto possível da realidade do usuário* no domínio de atividade a que o sistema se refere.

Exemplo

- correspondência entre o sistema e o mundo real

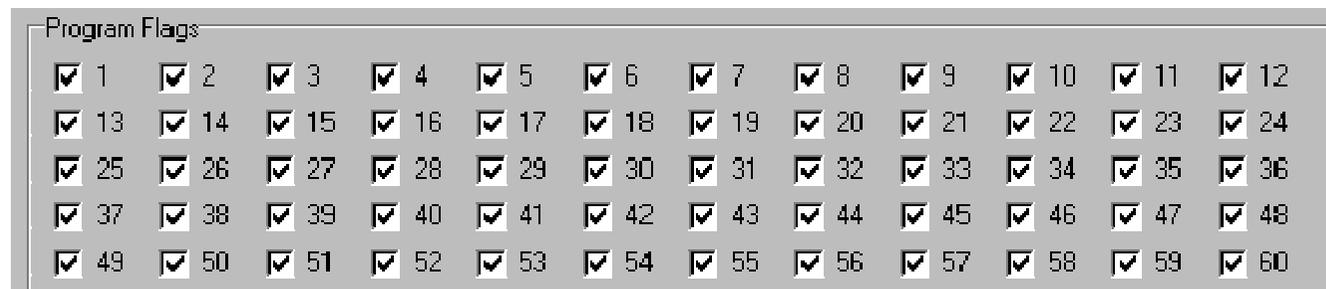
CSE HTML Validator, 4.0

-Verifica se há erros de sintaxe em documentos HTML.

-Cada flag está associado a um conjunto de tags HTML. Explicação disponível no *help*.

Heurística é **violada**.

Projetista preocupado em adaptar rapidamente o sistema a mudanças na especificação de tags.

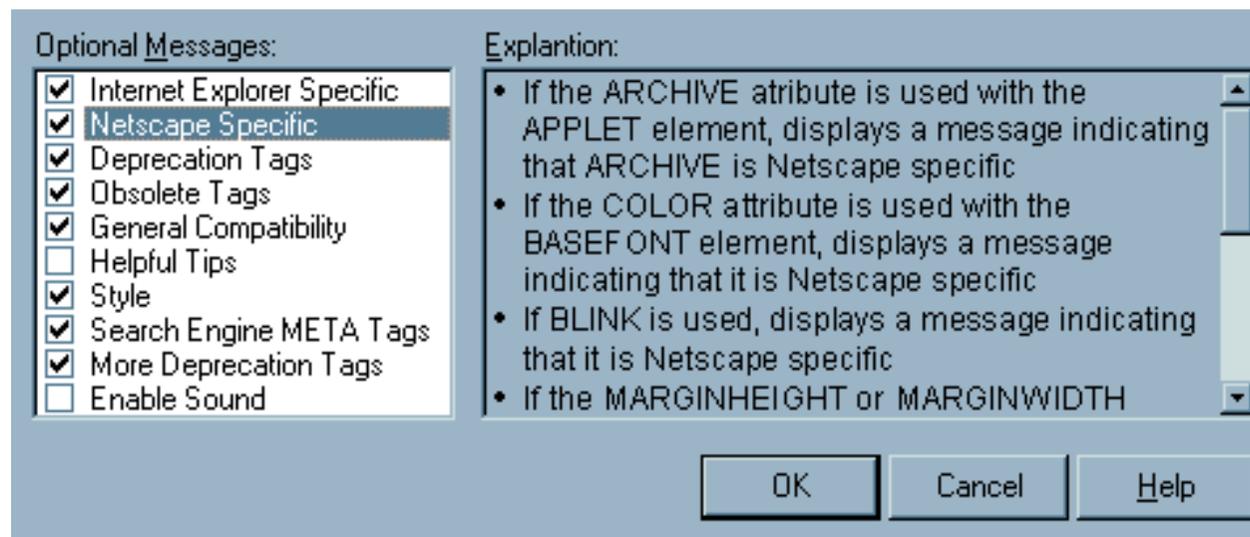


<http://homepage.mac.com/bradster/iarchitect/>

Exemplo

- correspondência entre o sistema e o mundo real

Uma solução muito melhor! Beneficia tanto o usuário quanto o projetista.



<http://homepage.mac.com/bradster/iarchitect/>

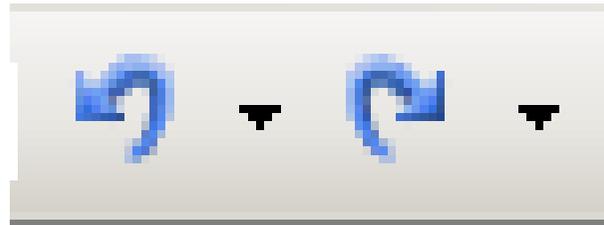
As 10 Heurísticas de Nielsen (3ª)

3. Controle e liberdade para o usuário

- A interface deve ser mais *reativa* do que *ativa* (deixar o usuário controlar o sistema). E como ao exercer este controle, o usuário pode ocasionalmente incorrer em erro, ela deve oferecer a possibilidade de *desfazer* o(s) último(s) comando(s) para o usuário retornar rápido ao (a um) estado anterior.

Exemplo

- controle e liberdade do usuário



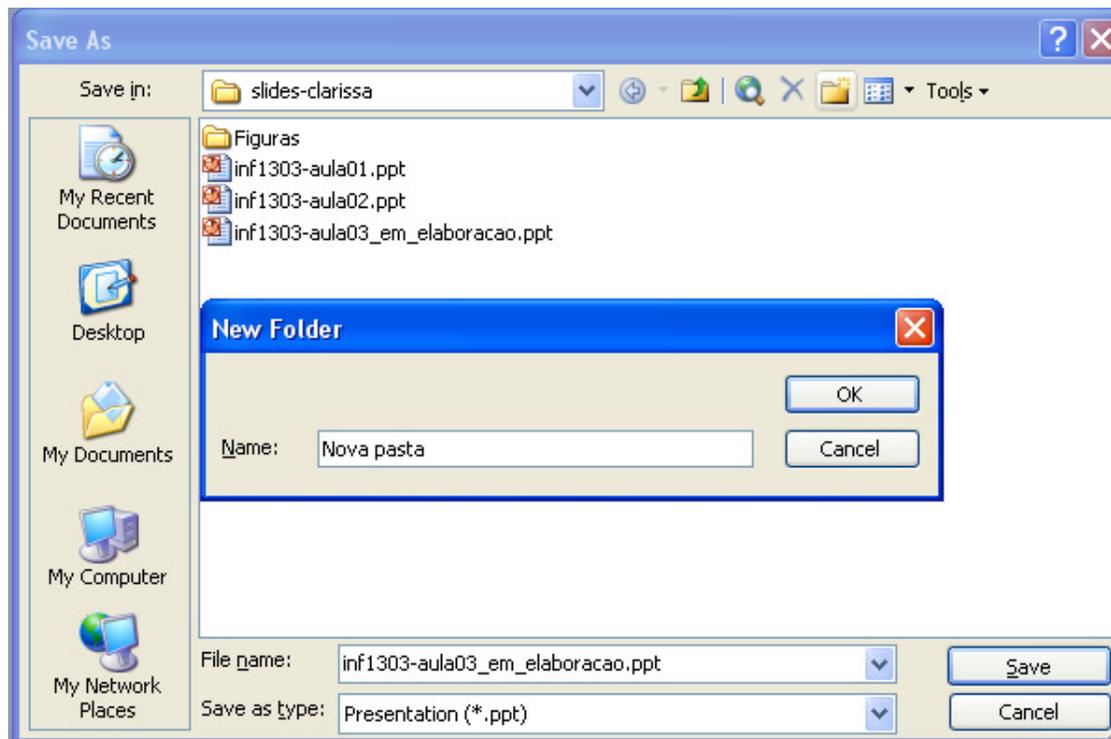
As 10 Heurísticas de Nielsen (4ª)

4. Consistência e Padronização

- Palavras, signos, interações *semelhantes ou relacionados* devem ter em comum significados *semelhantes ou relacionados* (e vice-versa). Caso a plataforma em que o sistema está rodando tenha padrões estabelecidos, a interface deve adotá-los.

Exemplo

- Consistência e padronização



Heurística é violada.

Save As permite que o usuário crie uma pasta.

Cancel destacado não cancela toda a operação realizada. A pasta a recém criada não é removida.

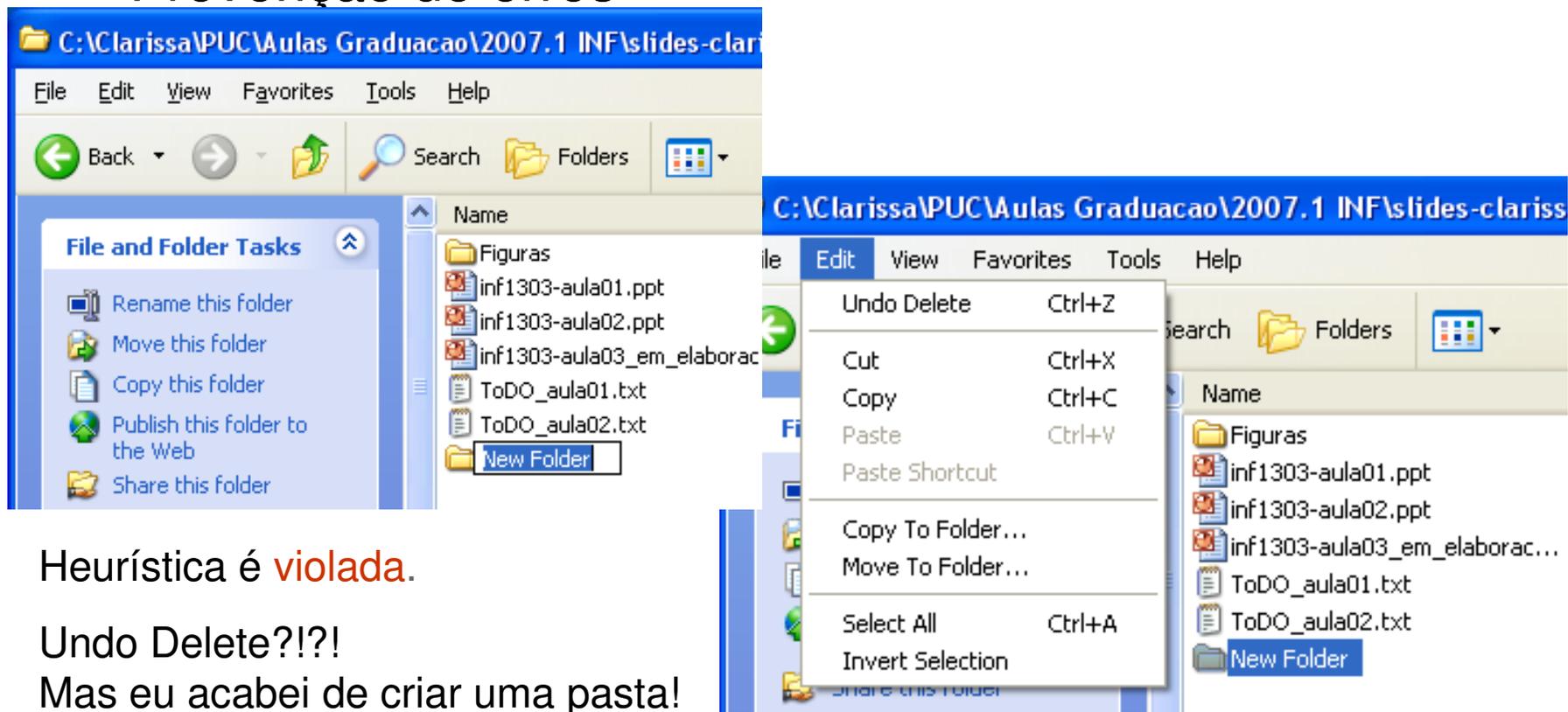
As 10 Heurísticas de Nielsen (5ª)

5. Prevenção de Erros

- A interface do sistema deve informar/sinalizar claramente ao usuário os efeitos e consequências de suas ações, para evitar enganos. Sempre que possível, deve evitar erros se puder detectar que as pré-condições para uma ação não estão satisfeitas, ou que a ação não é cabível no contexto corrente.

Exemplo

- Prevenção de erros



Heurística é **violada**.

Undo Delete?!?!

Mas eu acabei de criar uma pasta!

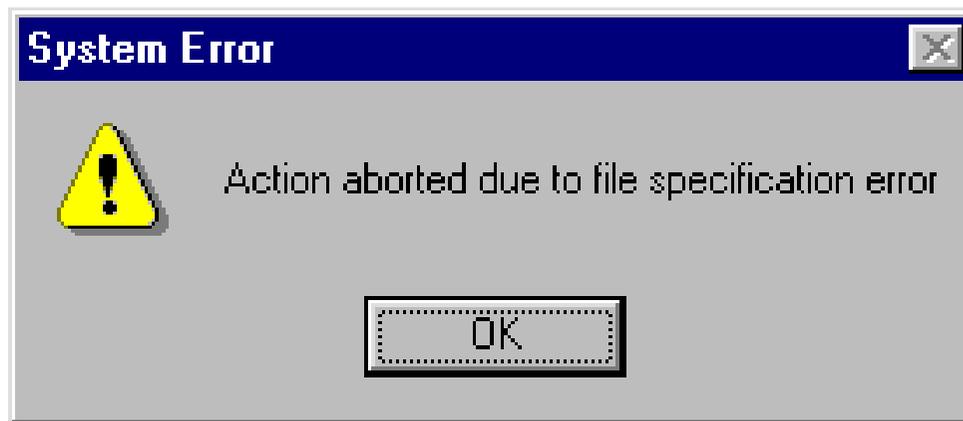
As 10 Heurísticas de Nielsen (6ª)

6. Ajuda para reconhecer, diagnosticar e remediar erros

- O sistema deve ter mensagens de erro claras e informativas, que ajudem o usuário a entender o que houve e reparar o erro.

Exemplo

- ajuda aos usuários para reconhecerem, diagnosticarem e se recuperarem de erros



Heurística é **violada**.

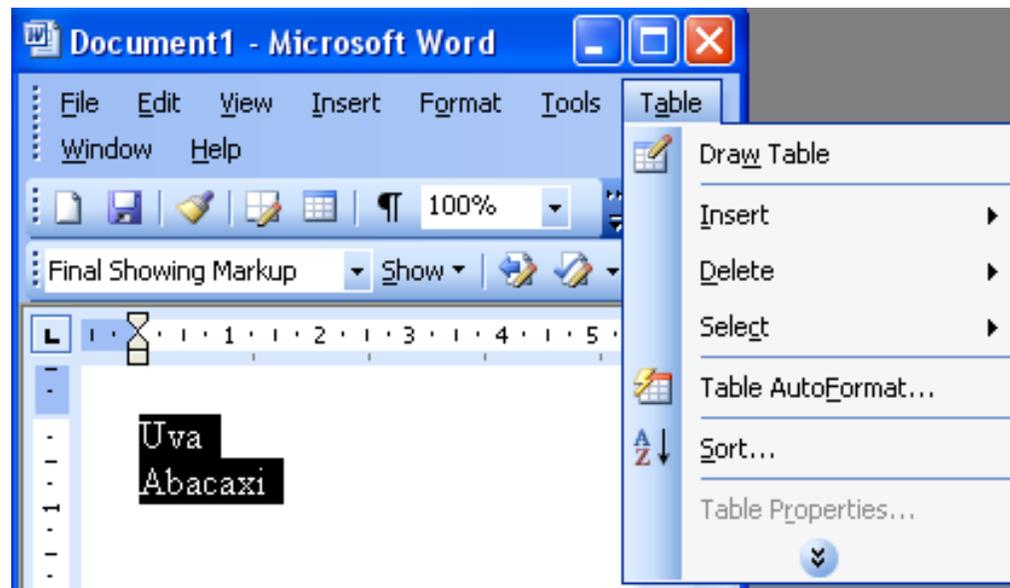
As 10 Heurísticas de Nielsen (7ª)

7. Reconhecimento ao invés de memorização

- A interface não deve exigir que o usuário *decore* a forma de acionar o sistema. Ao contrário, deve apresentar claramente as alternativas de ação, de modo que baste o usuário 'bater o olho' e reconhecer a ação a executar.

Exemplo

- reconhecimento em vez de memorização



Heurística é **violada**.

Ordenar uma lista no menu Table?!?!

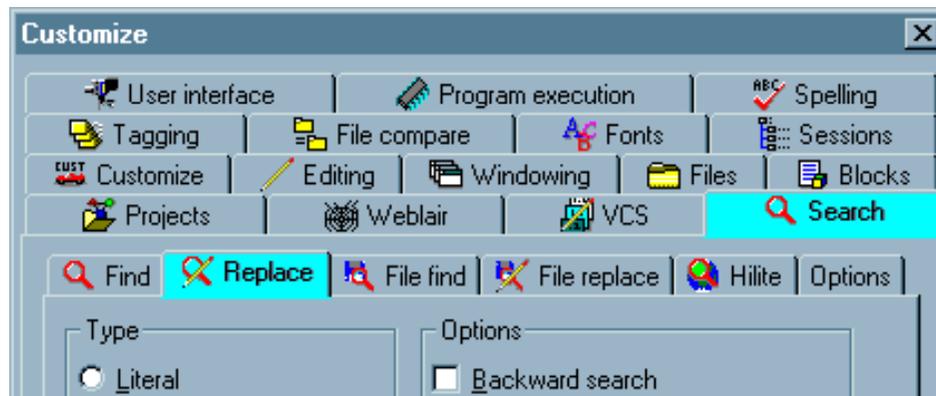
As 10 Heurísticas de Nielsen (8ª)

8. Flexibilidade e eficiência no uso

- As ações de interface devem ter diferentes formas de ser acionadas, dispor de teclas aceleradoras associadas a elas e também deve ser possível customizar as interfaces para acionar ações frequentes.

Exemplo

- flexibilidade e eficiência de uso



Heurística é **violada**.

Único acesso às
categorias é com o *mouse*.

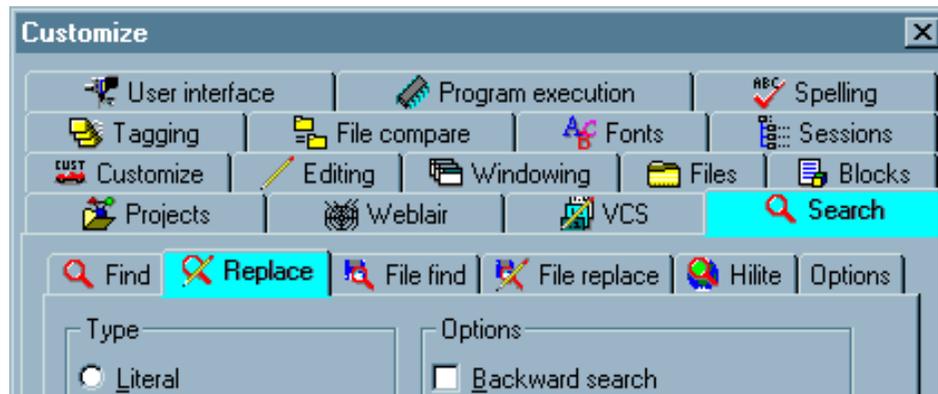
As 10 Heurísticas de Nielsen (9ª)

9. Design estético e minimalista

- A interface deve ter a quantidade de informação necessária – só o relevante, com ponto de acesso para mais, se o usuário quiser. Além disto o layout da interface deve ser agradável, bonito e leve.

Exemplo

- estética e design minimalista



Heurística é **violada**.

Imagens e cores distrativas.

Uma mesma situação pode violar mais de uma heurística.

As 10 Heurísticas de Nielsen (10ª)

10. Ajuda e Documentação

- O sistema deve oferecer ajuda para o usuário em todas as ações e atividades. O acesso deve ser claro e rápido, o conteúdo informativo e contextualizado, contemplando (organizadamente) os diferentes perfis de usuários a quem o sistema se destina. Hoje em dia também é muito importante oferecer suporte online eficiente e eficaz.

Procedimento

- **Determinação da Proposta de Design**
 - Apresentação: papel, protótipo ou produto acabado?
 - Verificação das condições gerais da inspeção: material completo e inspecionável a contento?
- **Navegação Geral pelo Sistema (ou sua representação)**
 - Qual o sentido geral que o avaliador dá ao sistema que vai analisar em detalhe?
- **Determinação do Perfil dos Usuários**
 - Quem são os usuários (suas características e contextos individuais, sociais, culturais)?
 - O que almejam realizar com o produto (principais metas)?
- **Determinação de Cenários de Uso**
 - Em que situações hipotéticas mas plenamente plausíveis os usuários (em que os avaliadores estão pensando quando fazem sua inspeção) poderiam encontrar-se?

Observação: Por vezes os avaliadores fazem inspeções de caráter mais geral, sem instanciar usuários específicos ou cenários de uso (é o caso por exemplo de produtos que têm problemas de usabilidade óbvios e demasiado graves).

Procedimento

- Cada avaliador
 - é guiado por um conjunto de heurísticas (princípios e regras básicas para o design);
 - julga a **conformidade** do produto aos princípios e regras selecionados;
 - anota que princípios e regras foram infringidos e onde;
 - julga a gravidade dos problemas encontrados; e
 - gera um relatório *individual* com suas conclusões e comentários.

A sessões individuais têm duração de 1 a 2 horas. A interface é examinada mais de uma vez com vistas a descobrir infrações dos princípios e regras heurísticas adotadas.

Resultado da Avaliação Heurística

- Um relatório CONSOLIDADO contendo o consenso dos especialistas sobre:
 - Problemas que acham que os usuários vão encontrar, e por quê;
 - Grau de *severidade* de cada problema (a escala normalmente usada é de 1 (menos severo) a 5 (mais severo));
 - Recomendações sobre:
 - Como resolver os problemas prioritários (ou seja, mais severos);
 - Como resolver problemas fáceis e rápidos de eliminar; e
 - Alternativas de encaminhamento de solução de outros problemas.



Qualificação dos Problemas – Localização

- **Localização**

- em um único local na interface
- em dois ou mais locais na interface, casualmente
- na estrutura geral da interface, de forma sistemática
- pode ser algo que “não está lá”, ou seja, precisa ser incluído na interface



Qualificação dos Problemas – Gravidade

- **Gravidade (ou Severidade)**
 - **freqüência** com que o problema ocorre
 - É um problema comum ou raro?
 - **impacto** do problema
 - Será fácil ou difícil para os usuários superarem o problema?
 - **persistência** do problema
 - É um problema que ocorre apenas uma vez e que os usuários conseguem superar facilmente, ou os usuários sofrerão com o problema repetidas vezes?



Escala de Gravidade

4: Catastrófico

- é imperativo consertar este problema antes do lançamento do produto
- muitos usuários não conseguirão atingir seus objetivos (para eles, o produto não funciona)

3: Problema grande

- importante de ser consertado; deve receber alta prioridade
- muitos usuários ficarão muito insatisfeitos

2: Problema pequeno

- o conserto deste problema é desejável, mas deve receber baixa prioridade
- alguns usuários ficarão insatisfeitos em momentos específicos

1: Problema cosmético



Para um exemplo detalhado, em português, consultar:

- **Avaliação de Interfaces de Usuário – Conceitos e Métodos.**
 - Jornadas de Atualização em Informática. Congresso da Sociedade Brasileira de Computação. 2003. Autoras: Raquel Oliveira Prates e Simone Diniz Junqueira Barbosa.
 - Online no website do curso
(consulte weblinks → bibliografia)
- **Material sobre avaliação heurística em inglês disponível em:** <http://www.useit.com/papers/heuristic/>

Exercício

- Para cada heurística apresentada, procure exemplos de violação em sistemas que você está acostumado a usar.
- Para os exemplos de violação, indique o grau de severidade que você entende que a violação representa, e procure sugerir pelo menos uma solução que você acredita que pode resolver o problema.