

INTERACÇÃO PESSOA MÁQUINA

# AULA 10

## AVALIAÇÃO

©2012-...LÍGIA FERREIRA

BASEADO EM MATERIAL ©ALAN DIX ©SALVADOR ABREU

@INTRODUÇÃO DESIGN DE INTERFACES



# AVALIAÇÃO

- Três objectivos principais:
  - ➡ As funcionalidades do sistema:
    - Funcionalidades disponíveis de acordo com a análise
  - ➡ Impacto da interface no utilizador
    - Capacidade de aprendizagem, capacidade de utilização, atitude do utilizador
  - ➡ Identificação dos problemas específicos do sistema

# AVALIAÇÃO

- **Desenho**: possivelmente sem envolver directamente o utilizador.
- **Implementação**: estuda a utilização real do sistema.
- Algumas técnicas são aplicadas em **ambos** os casos

# AValiação Heurística

- Criada por J. Nielsen, que fez vários estudos para avaliar a sua eficiência (taxa favorável de custo/benefício).
- Executada por um perito
- Com base em análise e julgamento
- Processo:
  - ➡ O avaliador inspecciona exhaustivamente a interface
  - ➡ Compara a interface com a heurística
  - ➡ Elaborar uma lista de problemas de usabilidade
  - ➡ Explica e justifica cada problema encontrado de acordo com a heurística

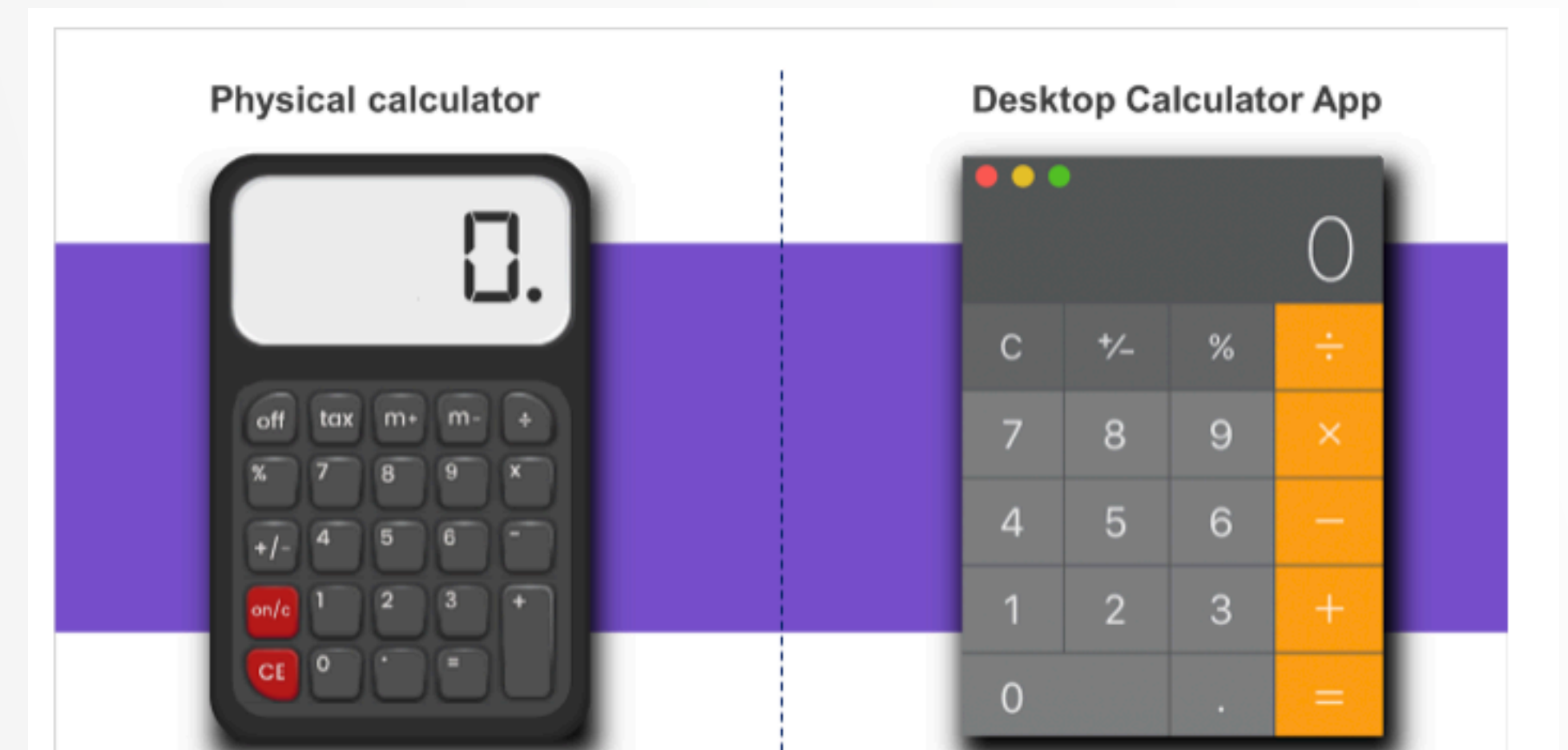
# DIRECTRIZES DE USABILIDADE

- Muito por onde escolher
  - ➡ Princípios Nielsen
    - [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)
  - ➡ Princípios Tognazzini
    - <http://www.asktog.com/basics/firstPrinciples.html>
  - ➡ Norman: Desenho de Coisas do Quotidiano
  - ➡ Mac, Windows, ... directrizes
- Ajuda para seleccionar alternativas de desenho
- Ajuda para identificar problemas em interfaces (avaliação heurística).

# HEURÍSTICAS DE USABILIDADE DE J. NEILSON

## 1. Match the real world(correspondência com o mundo real)

- ➡ "Falar a língua do utilizador"
- ➡ Usar palavras comuns
- ➡ Não impor limites aos nomes definidos pelo utilizador
- ➡ Permitir sinónimos em línguas de comando
- ➡ As metáforas são úteis, mas...tenham cuidado.



# HEURÍSTICAS DE USABILIDADE DE J. NEILSON

- Consistency & Standards(Consistência & Standards)
  - ➡ "Princípio da menor surpresa"
  - ➡ Coisas semelhantes devem parecer e comportar-se de forma semelhante
  - ➡ Coisas diferentes devem parecer diferentes
  - ➡ Palavras, cor, posição, tamanho, ordem
  - ➡ Seguir as normas da plataforma

# HEURÍSTICAS DE USABILIDADE DE J. NEILSON

## 3. Help & Documentation(Ajuda & Documentação)

- ➡ em geral, os utilizadores não lêem os manuais do utilizador
  - ... excepto quando não têm outra hipótese
- ➡ Mas os manuais do utilizador e a ajuda on-line são essenciais
- ➡ A ajuda deve ser:
  - Contextual
  - Pesquisável
  - Orientada para as tarefas
  - Concreta
  - Concisa



# HEURÍSTICAS DE USABILIDADE DE J. NEILSON

## 5. Visibility of system status(Visibilidade do estado do Sistema)

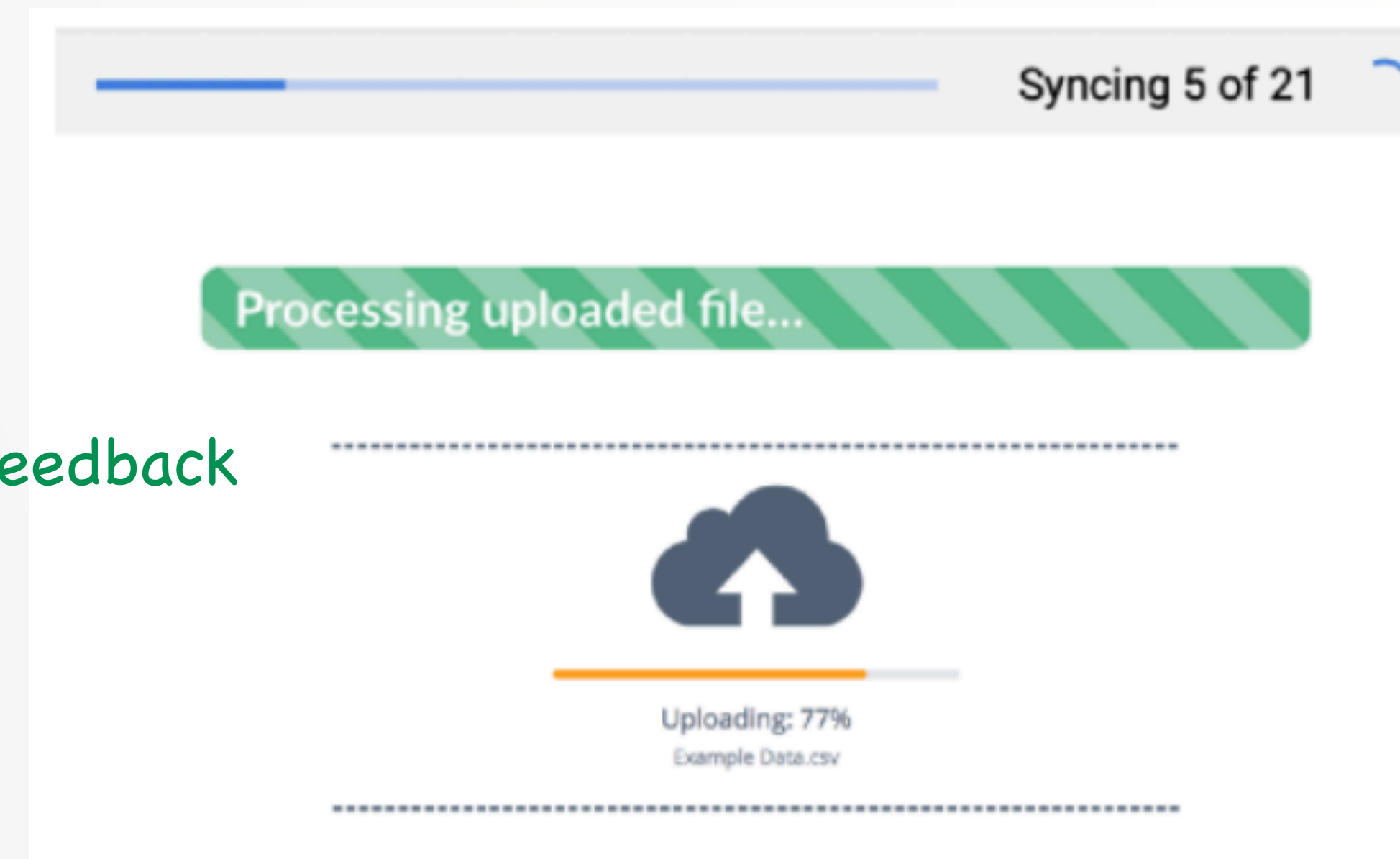
### ➡ "Feedback"

### ➡ Manter o utilizador informado sobre o estado do sistema

- Apresentar destaque de selecção
- Mudanças de cursor
- Barra de estado, barra de progresso

### ➡ Tempo de resposta

- < 0,1s: parece instantâneo
- 0,1-1s: informar o utilizador, mas não há necessidade de feedback
- 1-5s: exibir cursor ocupado
- > 5s: mostrar uma barra de progresso



<https://medium.com/nyc-design/1-visibility-of-system-status-with-examples-5e3bc9adfe7b>

# HEURÍSTICAS DE USABILIDADE DE J. NEILSON

- Flexibility & Efficiency of use(Flexibilidade & Eficiência de utilização)
  - ➡ Atalhos para operações frequentes
    - Aceleradores de teclados
    - Abreviaturas de comando
    - Favoritos
    - História
  - ➡ Macro para acções repetitivas

# HEURÍSTICAS DE USABILIDADE DE J. NEILSON

## 7. Error prevention(Prevenção de erros)

- ➡ Não dar aos utilizadores a oportunidade de cometerem erros
- ➡ A selecção é menos dada ao erro que a dactilografia
- ➡ Desactivar operações ilegais
- ➡ Evitar modos.

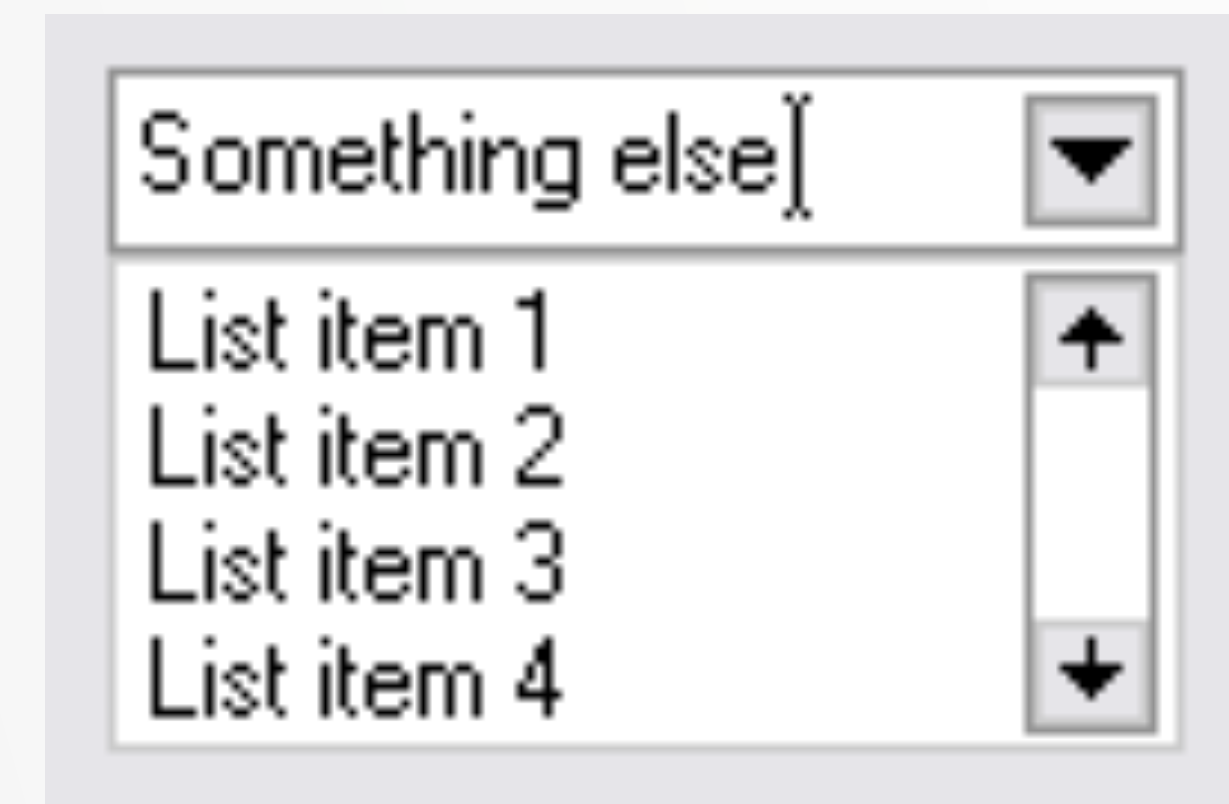
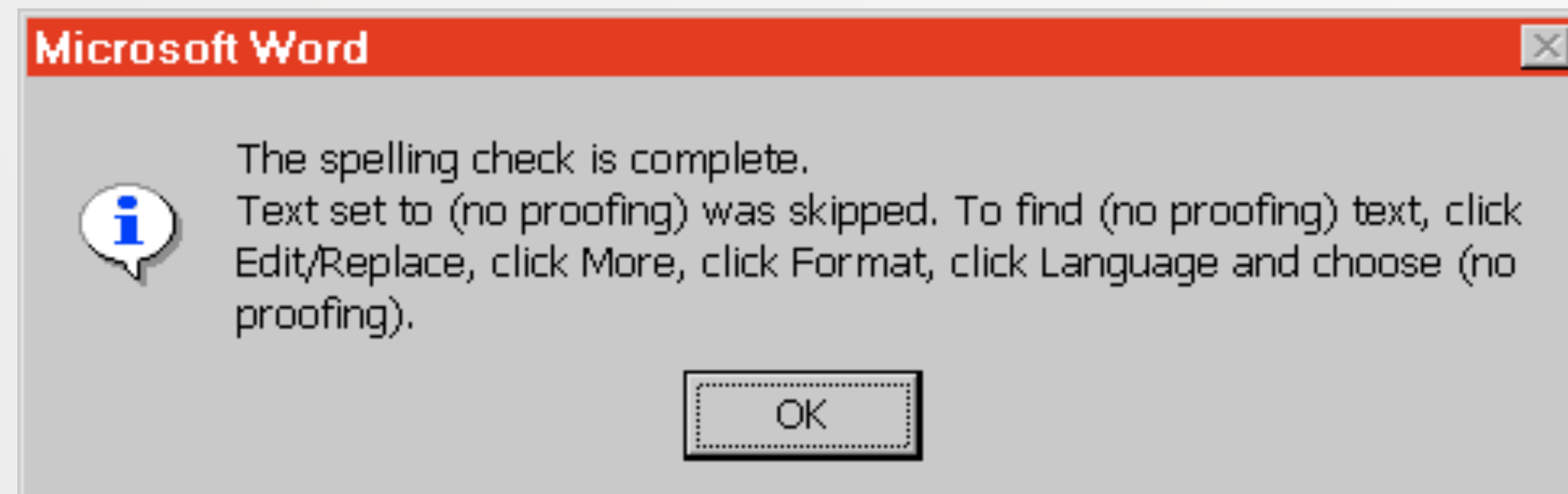




# HEURÍSTICAS DE USABILIDADE DE J. NEILSON

## 8. Recognition, Not Recall("Reconhecer é melhor que recordar")

- ➡ "Minimizar o uso da memória(MCD)"
- ➡ Norman: "Conhecimento na cabeça" vs. "conhecimento no mundo".
- ➡ Menus vs. comandos
- ➡ Caixas de Combo vs. caixas de texto
- ➡ Toda a informação necessária deve estar visível

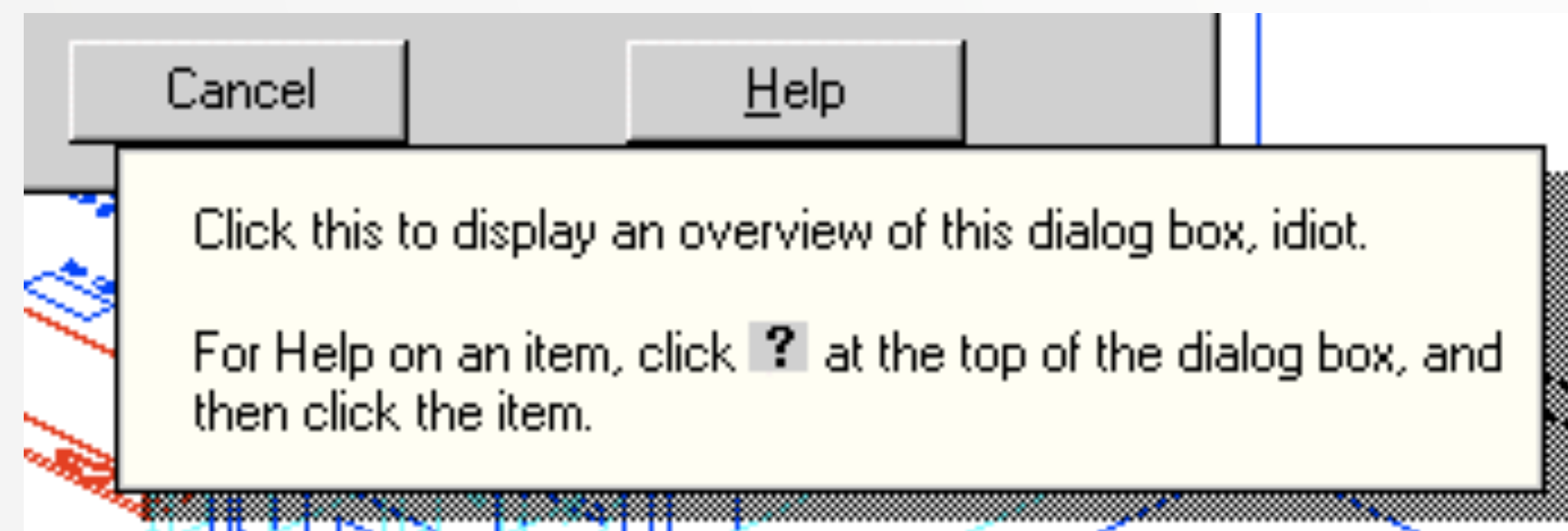


# HEURÍSTICAS DE USABILIDADE DE J. NEILSON

- Error reporting, diagnosis, and Recovery "Relatório de erros, diagnóstico e recuperação".

➡ As boas mensagens de erro devem:

- ser precisas: "Can't open file" vs. "Can't open file nnn.doc"
- ser construtivas: por que razão ocorreu o erro e como corrigi-lo
- ser educadas: devendo evitar que o utilizador se sinta culpado
- Ocultar detalhes técnicos até que tal seja solicitado



# HEURÍSTICAS DE USABILIDADE DE J. NEILSON

- Aesthetic and Minimalist Design(Estética e Desenho Minimalista)

- ➔ "Simplicidade"

- ➔ Menos é mais

- omitir tudo o que é supérfluo

- ➔ Bom desenho gráfico

- poucas cores, bem escolhidas
  - agrupar com espaços brancos
  - alinhamento de controlos

- ➔ Usar linguagem concisa

- escolher cuidadosamente os rótulos





# HEURÍSTICAS DE USABILIDADE DE J. NEILSON

- Satisfazer as expectativas do utilizador
  1. Match the real world
  2. Consistency & standards
  3. Help & documentation
- O utilizador é quem manda
  4. User control & freedom
  5. Visibility of system status
  6. Flexibility & efficiency
- Suporte ao erro
  7. Error prevention
  8. Recognition, not recall
  9. Error reporting, diagnosis, and recovery
- Simplificar
  10. Aesthetic & Minimalist design

# AValiação Heurística

- Justificar cada problema com uma heurística
  - ➡ "Permite colocar artigos fora de stock no carrinho de compras" - "Prevenção de erros"
  - ➡ "Eu não gosto da fonte" - subjectivo
- Enumerar todos os problemas encontrados
  - ➡ o mesmo elemento de interface pode ter vários problemas
- Inspeccionar a interface duas vezes
  - ➡ Uma vez, para obter uma visão geral - "entrar na onda!"
  - ➡ À segunda, para se concentrar em determinados elementos de interface
- Ir além dos 10 princípios da Nielsen
  - ➡ Affordances, visibilidade, lei de Fitt, princípios da cor
  - ➡ mas as 10 heurísticas são fáceis de comparar com

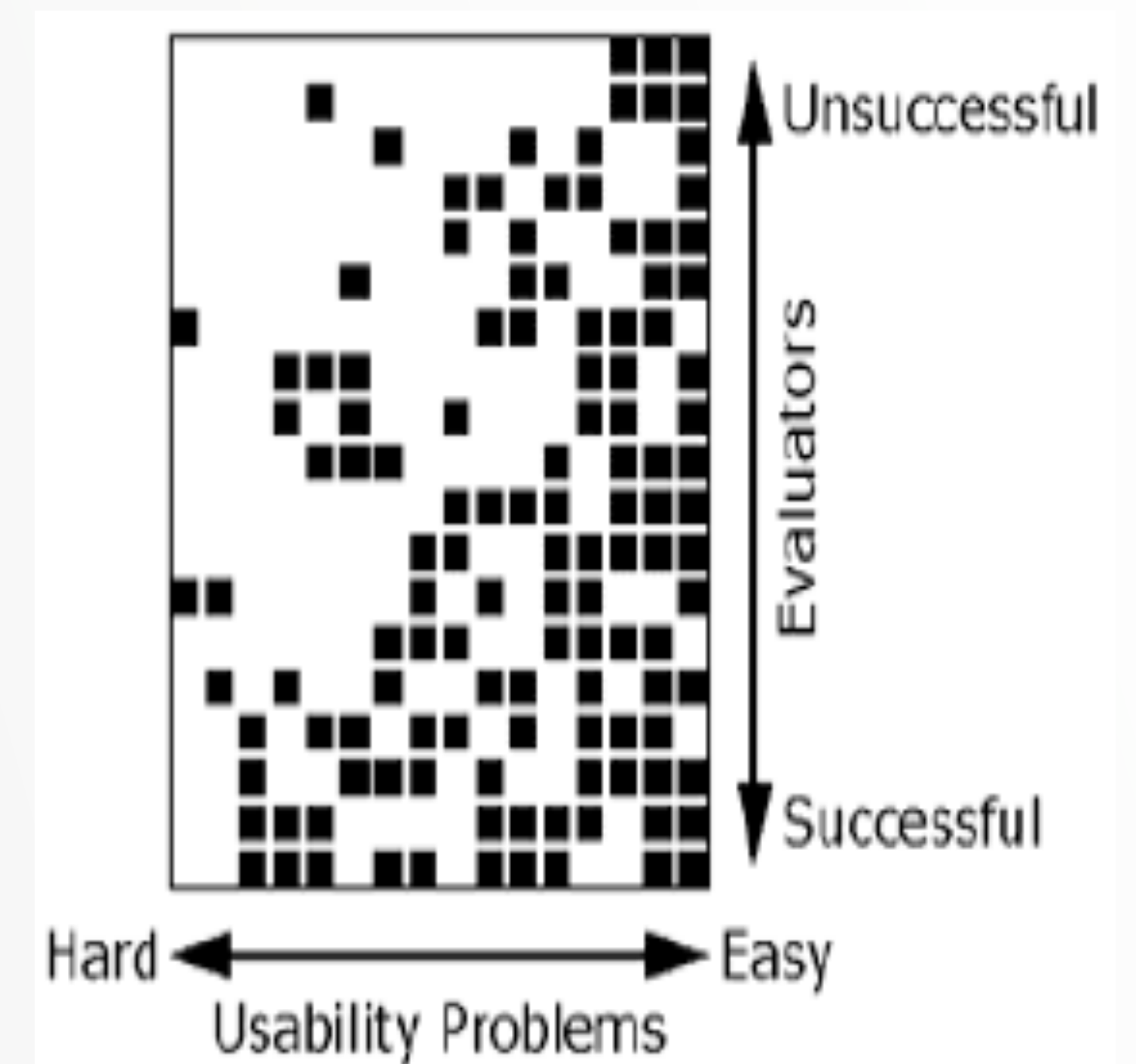
# AVALIAÇÃO HEURÍSTICA

- Avaliação heurística  $\neq$  testes com o utilizador
- O avaliador não é um utilizador
- Analogia: Inspeção de código vs. testes
- Permite descobrir problemas não detectados nos testes com utilizadores(ex: inconsistência da fonte)
- Aplica-se tanto a esboços como a protótipos funcionais



# AValiação Heurística

- Utilizar vários avaliadores
  - ➔ Diferentes avaliadores encontram problemas diferentes
  - ➔ Cada novo avaliador encontra poucos problemas novos
  - ➔ Nielsen recomenda 3-5 avaliadores
  - ➔ Os bons avaliadores encontram os problemas simples
    - e os complexos

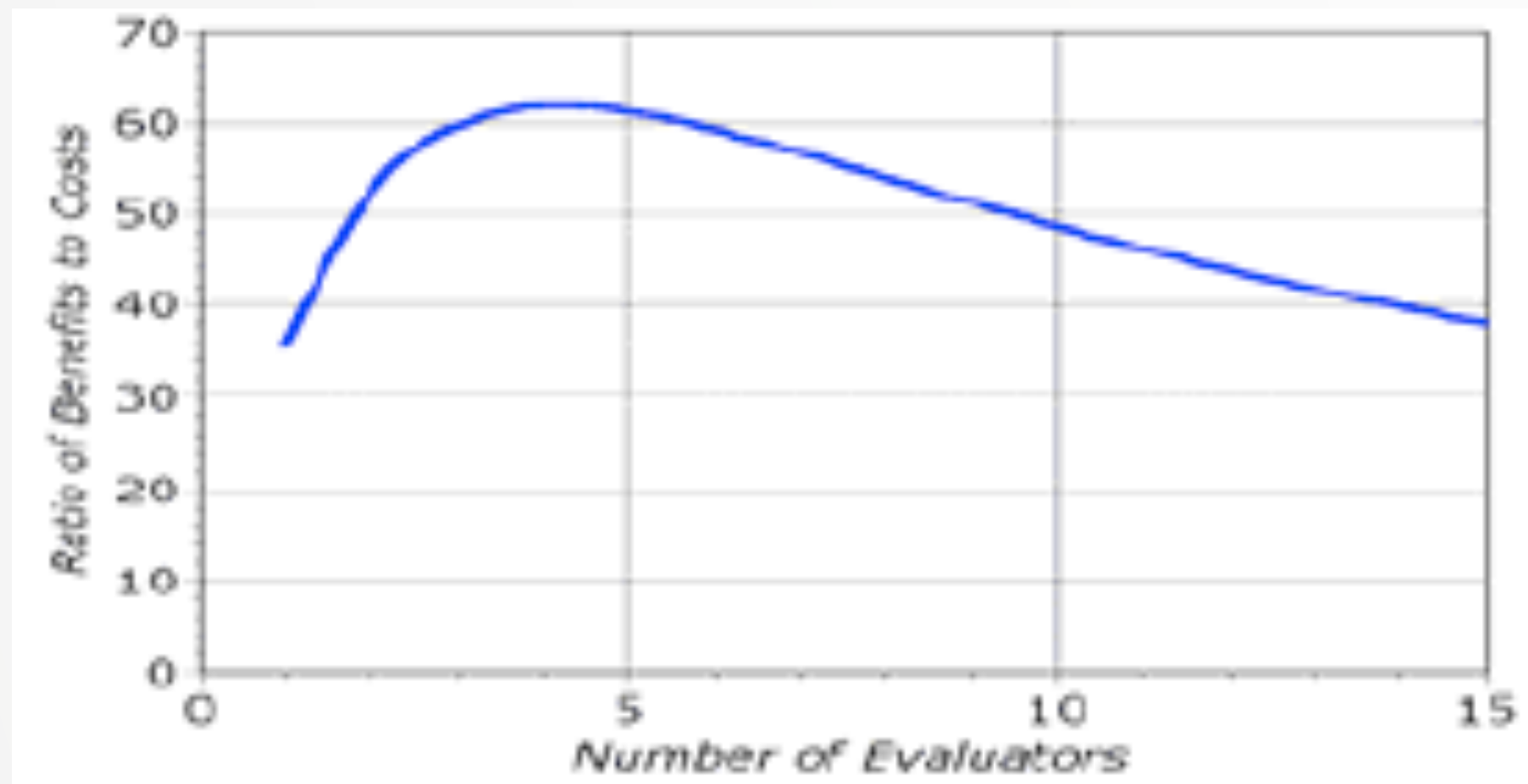
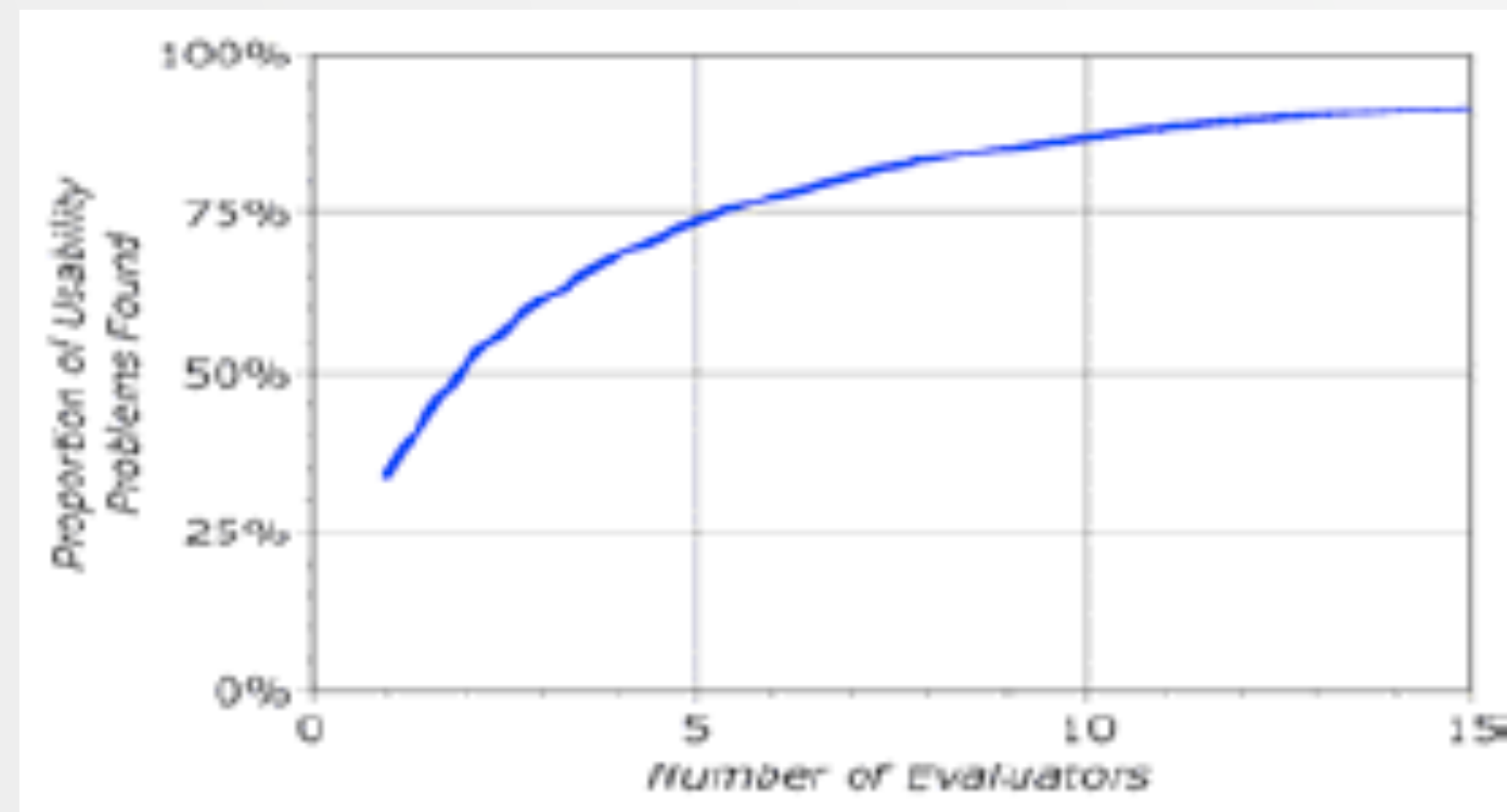


# AVALIAÇÃO HEURÍSTICA

- Um avaliador – resultados pouco fiáveis
  - ➡ detecta 35% dos problemas de usabilidade
- 5 avaliadores
  - ➡ detectar 75% dos problemas de usabilidade
- Porque não mais? 10? 20?
- Custo(\$)
  - ➡ Cada novo avaliador encontra poucos problemas novos

# AValiação HEURÍSTICA

- Rácio Custo/benefício



Nilsen, 1994



# AVALIAÇÃO HEURÍSTICA

- Processo formal

## 1. Formação/treino

- Reunião para a equipa de desenho e os avaliadores
- aborda a aplicação, utilizadores alvo, cenários,...

## 2. Avaliação

- Os avaliadores trabalham separadamente
- Produzir um relatório escrito ou comentários orais registados por um observador
- Identificam os problemas, mas não a sua gravidade

## 3. Classificação de severidade

- Todos os problemas identificados por todos os avaliadores são compilados numa lista
- Os avaliadores classificam cada um dos problemas
- Calcular a média das classificações dos avaliadores.

## 4. Discussão dos resultados

- Equipa de design e avaliadores
- Brainstorm -> soluções

# AVALIAÇÃO HEURÍSTICA

- Severidade

- ➡ Factores:

- Frequência (comum ou raro)
    - Impacto (fácil ou difícil de superar)
    - Persistência (demora muito a recuperação?)

- ➡ Escala:

- Cosmética: a correcção não é obrigatória
    - Menor: corrigir com baixa prioridade
    - Maior: corrigir com alta prioridade
    - Catastrófico: a correcção é obrigatória

# AVALIAÇÃO HEURÍSTICA

- Aplica-se a:
  - ➡ Esboços
  - ➡ Protótipos de papel
  - ➡ Protótipos instáveis
- Prós e contras
  - ✓ Mais barato
  - ✓ Rápido
  - ✓ Identifica uma série de problemas: menores e maiores.
    - Mais difícil de identificar elementos em falta num esboço.
    - Mais difícil de identificar os problemas relacionados com o domínio do problema.

# AVALIAÇÃO HEURÍSTICA

- Alternar avaliação heurística com testes de utilizador
  - ➡ Identificam diferentes problemas
  - ➡ Avaliação heurística – menos dispendiosa
- O observador pode ajudar o avaliador
  - ➡ Desde que os problemas sejam identificados.



# PRINCÍPIOS DE TOG

- |                           |                             |
|---------------------------|-----------------------------|
| 1. Anticipation           | 9. Human-interface Objects  |
| 2. Autonomy               | 10. Latency-reduction       |
| 3. Colour Blindness       | 11. Learnability            |
| 4. Consistency            | 12. Limit tradeoffs         |
| 5. Defaults               | 13. Metaphors               |
| 6. Efficiency of the user | 14. Protect the user's work |
| 7. Explorable interfaces  | 15. Readability             |
| 8. Fitt's law             | 16. Track state             |
|                           | 17. Visible Interfaces      |

# PRINCÍPIOS DE TOG

## 1. Antecipação(**Anticipation**)

➡ antecipar as necessidades do utilizador

## 2. Autonomia(**Autonomy**)

➡ Dar o controlo ao utilizador

➡ Providenciar a informação necessária para tal

## 3. Daltonismo(**Colour Blindness**)

➡ não usar exclusivamente a cor para codificar informação (ser redundante)

## 4. Consistência(**Consistency**)

➡ Sobretudo, ser consistente com as expectativas do utilizador

# PRINCÍPIOS DE TOG

## 5. Defaults

- ➡ os valores que vêm por defeito, devem poder ser substituídos facilmente. Os conteúdos por omissão devem ser inteligentes e reactivos

## 6. Eficiência do utilizador(Efficiency of the user)

- ➡ Enfoque na produtividade do utilizador, não na do computador
- ➡ Mais rápido? Ferver água no micro-ondas em 1m:10s ou 1m:11s???
- ➡ A palavra chave deve ocorrer em primeiro lugar, nas etiquetas ou menus:

| Wrong   | Right  |
|---|--|
| <ul style="list-style-type: none"><li>● Insert page break</li><li>● Add Footnote</li><li>● Update Table of Contents</li></ul> | <ul style="list-style-type: none"><li>● Insert:<ul style="list-style-type: none"><li>● Page break</li><li>● Footnote</li><li>● Table of contents</li></ul></li></ul> |

# PRINCÍPIOS DE TOG

## 7. Interfaces exploráveis(Explorable Interfaces)

- ➡ “Dar aos utilizadores pontos de referência e estradas bem assinaladas, e depois permitir que passem para todo o terreno”.
- ➡ Ações reversíveis, permitindo sempre o desfazer -> UNDO

## 8. Lei de Fitts(Fitt's Law)

- ➡ Usar objectos grandes para acções importantes
- ➡ Quanto maior o botão, maior a rapidez com que o alcançamos

## 9. Objectos típicos da interface com o Humano(Human-interface Objects)

- ➡ directorias, ficheiro, reciclagem
- ➡ consistentes, estáveis, e de significado standartizado



# PRINCÍPIOS DE TOG

- 10.Redução da latência(Latency reduction)
  - ➡Multi-tasking, sempre que possível, “empurrando” a latência para os bastidores
  - ➡Dar uma resposta visual ou auditiva no espaço de tempo de 50 ms
  - ➡Mostrar uma ampulheta por cada acção que demore mais de 1/2 a 2 s
  - ➡Comunicar o tempo de demora efectiva através de um indicador de progresso.
  - ➡Enviar mensagens que indiquem as acções do sistema
  - ➡Impedir o pressionar múltiplo do mesmo botão ou objecto.

# PRINCÍPIOS DE TOG

## 11. Capacidade de ser aprendido(Learnability)

- ➡ O ideal é não existir curva de aprendizagem: isto é aquando da 1ª utilização dum produto os utilizadores deveriam transformar-se em peritos.
- ➡ O ideal não existe.

## 12. Limitar compromissos(Limit Tradeoffs)

- ➡ Usabilidade e capacidade de ser aprendido, não são mutuamente exclusivos. Decida-se qual o mais importante e ataquem-se ambos.

## 13. Metáforas(Metaphors)

- ➡ escolher metáforas correctamente, metáforas que permitam que os utilizadores, intuitivamente vislumbrem os detalhes mais requintados do modelo conceptual.
- ➡ Boas metáforas são historias, criam retratos visuais na mente.

# PRINCÍPIOS DE TOG

## 14. Proteger o trabalho dos utilizadores(Protect the user's work)

- ➡ Garantir que em caso de erro da sua parte, o trabalho dos utilizadores nunca se perde

## 15. Legibilidade(Readability)

- ➡ O texto que deve ser lido, deve ter alto contraste. Favorecer texto preto em fundos brancos ou amarelo pálido. Evitar fundos cinzentos.
- ➡ Utilizar tamanhos de fonte que sejam suficientemente grandes por forma a serem legíveis nos monitores standard.
- ➡ Prestar uma atenção particular às necessidades das pessoas mais idosas.

# PRINCÍPIOS DE TOG

## 16. Monitorar o estado(**track state**)

- ➡ Onde estava o utilizador aquando da última sessão?
- ➡ Cookies

## 17. Navegação visível(**Visible Interfaces**)

- ➡ Evitar navegação invisível: A maioria dos utilizadores não pode e não vai construir mapas mentais
- ➡ A World Wide Web, para todos é composta de bonitos ecrãs e botões, é, de facto, um espaço de navegação invisível. É um facto que se consegue ver sempre a página onde estamos, mas não se consegue ver nada do vasto espaço entre páginas. Uma vez que os utilizadores chegam às nossas aplicações, temos que providenciar a redução da navegação ao mínimo e da navegação que resta torna-la clara e natural. Apresentar a ilusão de que os utilizadores estão sempre no mesmo local, com o trabalho a ser trazido até si. Isto não só elimina a necessidade de mapas ou outras ajudas de navegação, como oferece ao utilizador um maior sentido de domínio e autonomia.



# PASSAGEM COGNITIVA

## PASSEIO...CAMINHADA....

- Foco na capacidade de aprendizagem: os utilizadores preferem aprender enquanto exploram.
- Passagem de código(code walkthrough) na engenharia de software.
- Os avaliadores executam uma sequência de acções para atingir um objectivo, procurando potenciais problemas de usabilidade.

# PASSAGEM COGNITIVA

- Para a passagem, é necessário:
  - 1.Descrição do protótipo (não tem de ser completa, mas tem de ser detalhada).
  - 2.Descrição da tarefa (tarefa frequente).
  - 3.Lista completa das acções necessárias para completar a tarefa com o protótipo dado.
  - 4.Indicações sobre os utilizadores e a sua experiência.

# PASSAGEM COGNITIVA

- Os avaliadores passam pela sequência de acções (passo 3) e, para cada acção, tentam perceber se:
  - ➡ os utilizadores tentam executar a acção para executar a tarefa
  - ➡ os utilizadores notam que a acção correcta está disponível (visibilidade)
  - ➡ os utilizadores identificam a acção que devem fazer (ver o botão e saber que esse é o botão certo)
  - ➡ Compreendem o feedback da acção.

# PASSAGEM COGNITIVA

- Resultados:
  - ➡ Formulário com informações 1,2 e 4, data, hora, avaliador.
  - ➡ Formulário separado para cada acção (a partir de 3) para responder às perguntas anteriores.