

Universidade do Minho
Escola de Engenharia
Departamento de Informática

Interacção Humano-Computador em Ambientes Ubíquos

Módulo 1

Mestrado em Sistemas Móveis
2004/2005

José Creissac Campos
jose.campos@di.uminho.pt
Departamento de Informática, Universidade do Minho, Braga, Portugal.

<http://www.di.uminho.pt>



Bibliografia, etc.

- Livros
 - Human-Computer Interaction (3rd ed.). Dix et al. (2004) Pearson/Prentice-Hall.
 - Interactive Systems Design. Newman & Lamming (1995) Addison-Wesley.
 - The Design of Everyday Things. Norman (2002) Basic Books (POET, 1988)
- Web
 - www.usabilitynews.com (British HCI Group)
 - www.usabilitynet.org (Projecto financiado pela UE)
 - www.usabilityfirst.com



Computação Ubíqua

- Tentativa de libertação do paradigma de interacção *desktop*.
- Poder computacional muda-se para o ambiente que rodeia os utilizadores.
- A interface deve tomar a responsabilidade de localizar e servir o utilizador.

“Inspired by the social scientists, philosophers, and anthropologists at PARC, we have been trying to take a radical look at what computing and networking ought to be like. We believe that people live through their practices and tacit knowledge so that **the most powerful things are those that are effectively invisible to use**. This is a challenge that affects all of computer science. (...) We call our work '**ubiquitous computing**'. This is different from PDAs, dynabooks, or information at your fingertips. It **is invisible, everywhere computing that** does not live on a personal device of any sort, but **is in the woodwork everywhere.**”
(Wiser, 1994)



Motivação (I)

- Computadores cada vez mais pequenos.
De sala enormes passamos para computadores *invisíveis*.
- Complexidade dos sistemas desenvolvidos aumenta.
Quanto mais o hardware evolui, mais somos tentados a *atacar* problemas cada vez mais complexos.
- Em particular, a complexidade das técnicas de interacção aumenta.
Passamos de interruptores para reconhecimento de voz e gestos e daí para interacção implícita.
- Software utilizado em situações cada vez menos controladas.
O computador está em todo o lado.
- Deixamos de fazer software para nós para fazer software para os outros!
Aspecto muito importante: o que é que os outros querem do *nosso(?)* software?



Motivação (II)

- Dificuldades com computadores causa a perda de 5-10% do tempo de trabalho (Allwood, 1984).
- Um grupo de utilizadores experientes perdia pelo menos 10 minutos por dia devido a problemas de usabilidade (Nielsen, 1993)
- Os utilizadores estão cada vez menos dispostos a tolerar produtos de difícil utilização (Jordan, 1996)
- Três em cada quatro utilizadores de computadores não conseguem que as suas máquinas façam o que pretendem (The Sunday Times, 22 de Fevereiro, 1998).
- O design (incluindo a usabilidade) é um dos poucos pontos em que é possível ganhar vantagens comerciais significativas sobre a concorrência (Jordan, 1998).

[Ver filme](#)



Breve história da IHC

Anos 60 Programadores

- Software elaborado por programadores para programadores
- Software funciona essencialmente em **batch**. Não existem grandes preocupações com a interface¹.

Anos 70 Utilizadores profissionais

- Computadores *mainframe*. Interfaces de “linha de comando”.
- Noção de que a *interface com o utilizador* é importante . Atenção centra-se no **desempenho** dos utilizadores.

Anos 80 Grupos vastos de utilizadores indiferenciados

- Pcs. Interfaces WIMP, Manipulação directa.
- Noção de que desempenho não é tudo. Necessidade de **auto-aprendizagem**.
- Surge o termo **Interacção Humano-Computador** (ciências da computação + psicologia cognitiva)

Anos 90 A internet!

- CSCW. Aspectos sociais e organizacionais.

2000+ Computação Ubíqua – Novos desafios...

- O computador desaparece?

¹ Nos anos 40 tinha surgido interesse na Ergonomia, mas esta lidava com as características físicas das máquinas e o seu impacto no desempenho dos operadores.



- **Interface com o utilizador**
 - “Aqueles aspectos do sistema com os quais o utilizador entra em contacto” (Moran 81)
 - Surge o conceito de *user-friendly* – muitas vezes pouco mais que interfaces *bonitinhas*.
- **Interacção Humano-Computador**
 - “conjunto de processos, diálogos e acções através dos quais um utilizador humano emprega e interage com um computador” (Baecker & Buxton 87)
 - “disciplina que se ocupa do desenho, avaliação e implementação de sistemas de computação interactivos para uso humano e com o estudo dos principais fenómenos que os rodeiam” (ACM SIGHCI 92)
 - “IHC concerne o desenho, implementação e avaliação de sistemas interactivos no contexto das tarefas e trabalho do utilizador” (Dix et al. 2004)

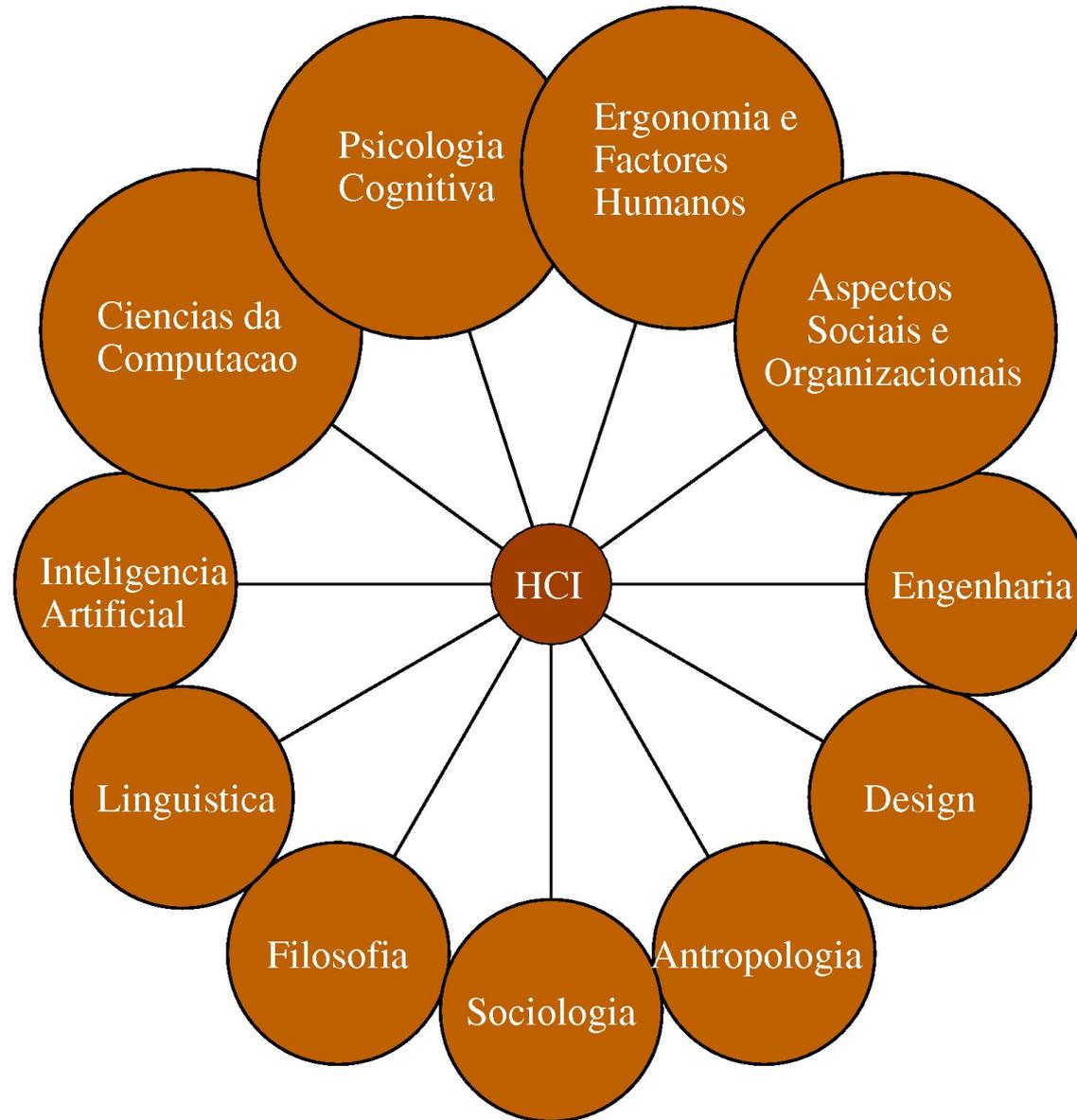
Houve uma clara evolução de conceitos!



- **Standards**
 - ISO 9241-11:1998 – *Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability*
 - ISO 13407:1999 – Human-centered design processes for interactive systems
- **Processos de desenvolvimento**
 - Human-centered design (ISO 13407)
 - User-centered design/User Engineering (IBM - www.ibm.com/easy)
 - Usage-centered design (Constantine & Lockwood - www.foruse.com)



Uma área multi-disciplinar





Usabilidade - Uma definição

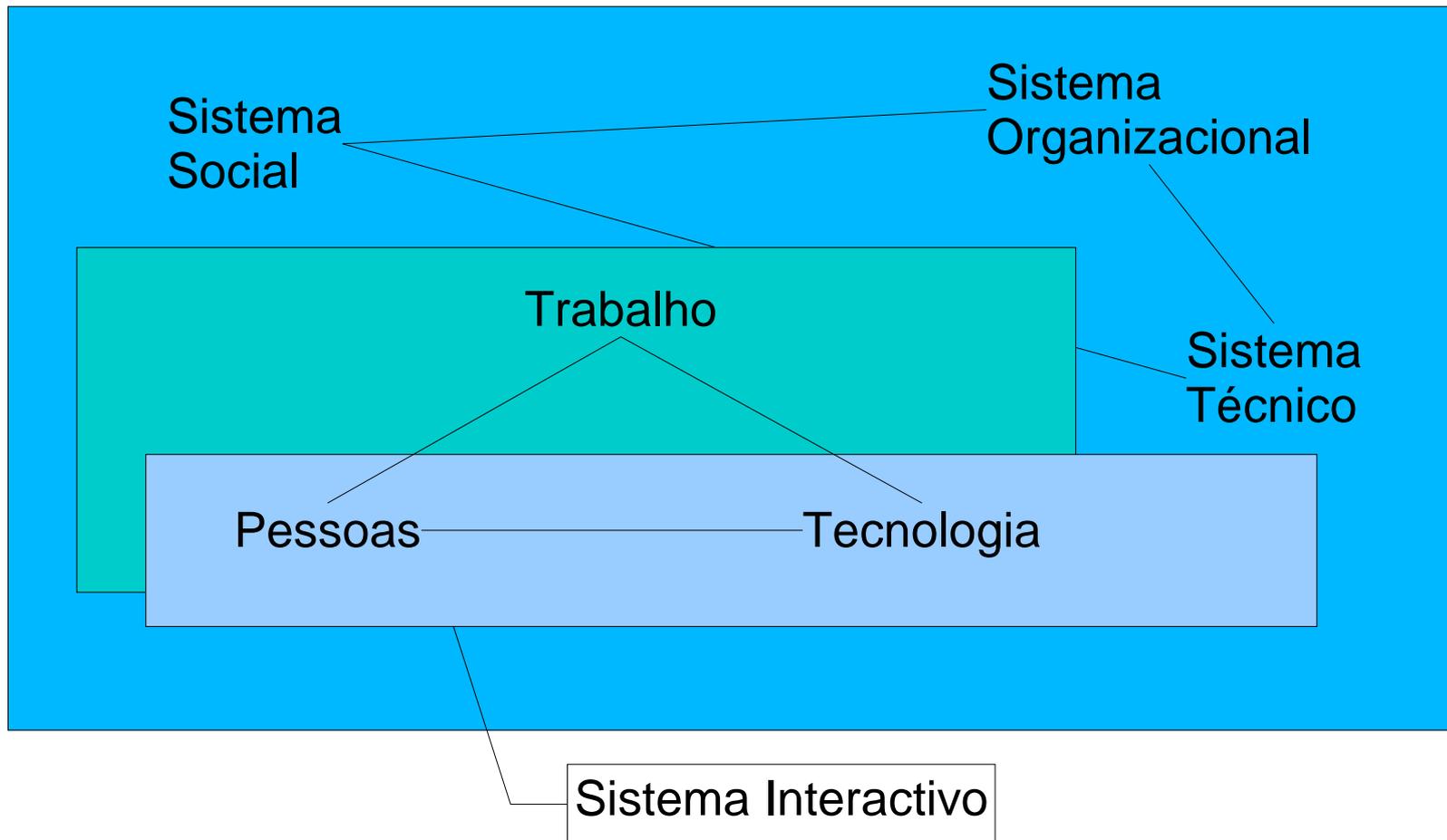
- A eficácia, eficiência e satisfação com **que utilizadores** determinados atingem **objectivos determinados** em **ambientes específicos** (Norma ISO DIS 9241-11).
 - *eficácia* - **o utilizador** consegue **realizar as tarefas** pretendidas;
 - *eficiência* - o custo de **atingir os objectivos** é aceitável (em termos de tempo, facilidade de aprendizagem, etc.)
 - *satisfação* - quão **confortáveis se sentem os utilizadores** com o sistema?

Usabilidade \neq *user friendliness*

- *User friendly*: fácil de usar (mas serve para alguma coisa?)



Usabilidade - Uma visão por níveis





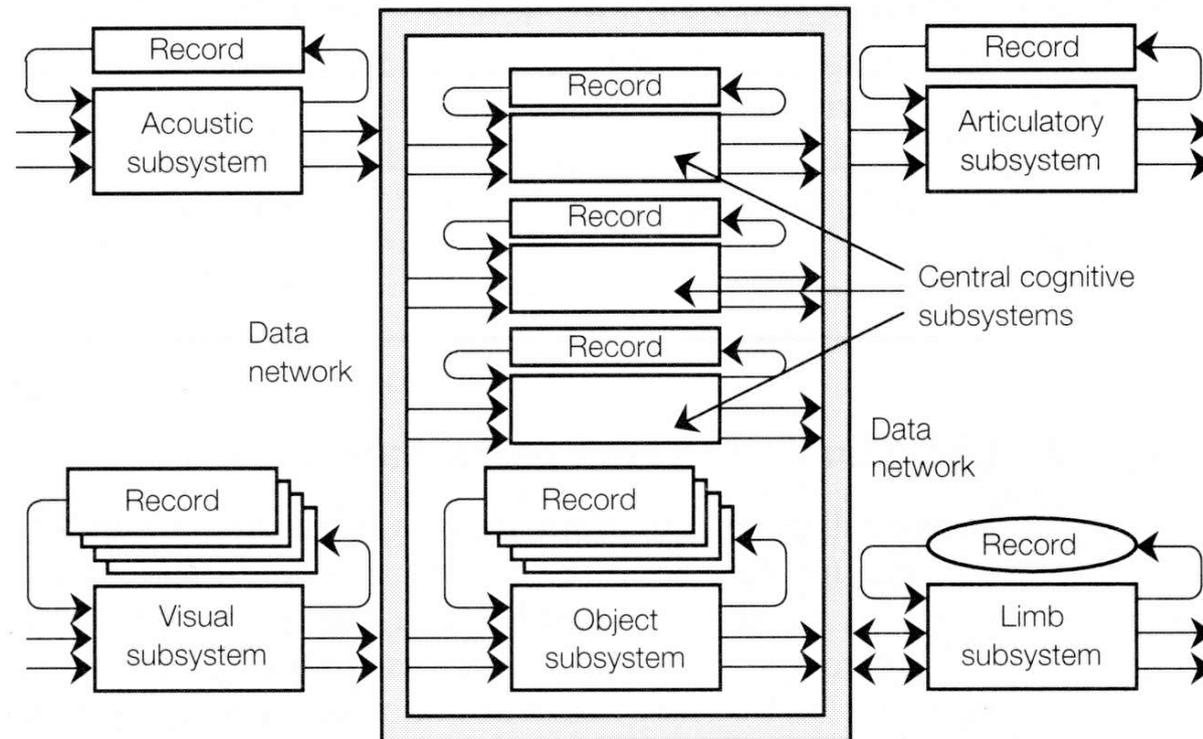
É necessário compreender

- Os utilizadores
 - A usabilidade deve ser definida em relação a um tipo específico de utilizador. Utilizadores experientes têm necessidades diferentes de utilizadores noviços.
- As actividades que pretendem/devem realizar
 - A usabilidade deve ser definida para tarefas específicas que o sistema deve suportar. No entanto, o sistema acabará muitas vezes por ser utilizado de formas não previstas inicialmente.
- O ambiente em que o devem fazer
 - O tipo de ambiente em que o sistema vai ser utilizado pode influenciar não só a usabilidade do sistema, mas a forma como os testes podem decorrer.



Conhecer o Utilizador / Arquitecturas Cognitivas

- Interacting Cognitive Subsystems
 - Procura definir uma arquitectura para o sistema cognitivo humano.



- Demasiado complexo para ser prático em termos de aplicação directa.

(Barnard, 1985)



Conhecer o Utilizador / Arquitecturas Cognitivas

- Programmable User Models
 - Definir uma arquitectura cognitiva que é *programada* com o conhecimento que o utilizador é suposto possuir.

OBJECTS

text: tx-1, tx-2, tx-3.

location: a, b, c, d.

RELATIONS

cursor-at(location)

buffer-contains(text)

text-at(text, location)

is-marked(text, location)

OPERATIONS

operation locate-cursor (location: L)

user-purpose: cursor-at(L)

subgoaling-precond: locate(L)

action: locate-at(L)

operation mark-text-at-loc (text: TX, location: L)

user-purpose: is-marked(TX, L)

subgoaling-precond: cursor-at(L)

filtering-precond: text-at(TX, L)

action: mark-text(TX)

operation cut-text-from-loc (text: TX, location: L)

user-purpose: not text-at(TX, L)

subgoaling-precond: is-marked(TX,L)

action: cut-text

operation copy-text-to-buffer (text: TX, location: L)

user-purpose: buffer-contains(TX)

subgoaling-precond: is-marked(TX,L)

predicted-effect: buffer-contains(TX)

action: copy-to-buffer

operation paste-text-at-loc (text: TX, location: L)

user-purpose: text-at(TX,L)

subgoaling-precond: buffer-contains(TX)

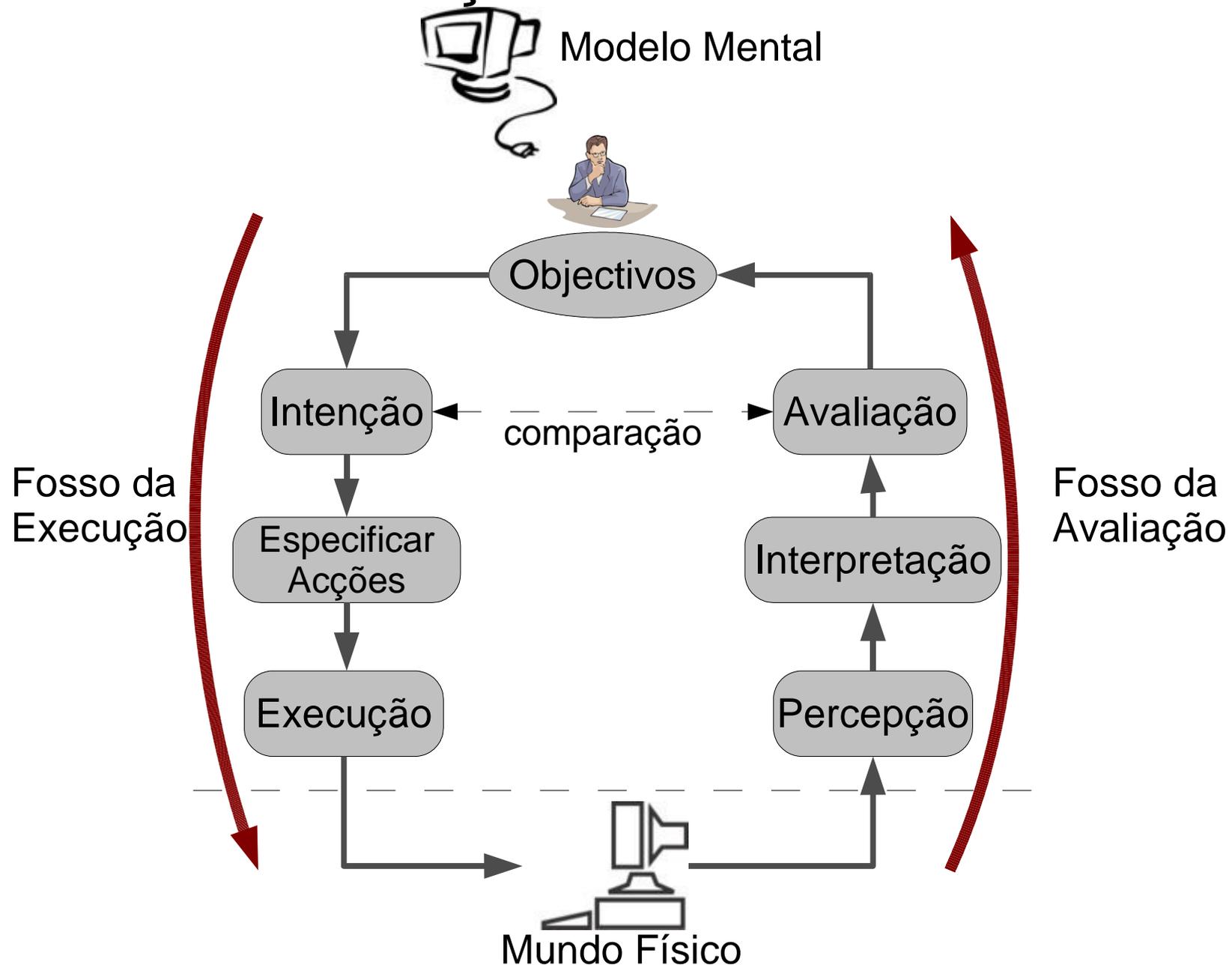
cursor-at(L)

action: paste-text

- Assume “comportamento racional”.



Modelo de Interação de Norman





- **Fosso da Execução**

- Esforço que o utilizador tem que realizar para efectuar determinada tarefa.
- Distância entre os objectivos do utilizador e a forma como pode atingi-los.
- Atenção à definição das tarefas!

- **Fosso da Avaliação**

- Esforço que o utilizador tem que realizar para perceber a interface.
- Distância entre a informação que a interface fornece (e de que forma) e aquela que o utilizador pretende.
- Atenção ao modo como a informação é apresentada!



3 desafios imediatos (+1)

- Como lidar com os Erros
- Como minimizar o Fosso da Execução
- Como minimizar o Fosso da Avaliação

- *Design*



Desafio 1: Errar é Humano!

- “é consensual que 60%-90% de todas as falhas são atribuíveis a erro humano” (Hollnagel, 1993)
- 92% das fatalidades consideradas num estudo entre 1979 e 1992 podiam ser atribuidas a problemas na interacção humano-computador (4% a causas físicas e 3% a erros de software) (MacKenzie, 1994)
- Mas...
 - ... erro de qual humano?
 - do que utiliza o sistema, ou do que o concebeu/implementou? (Leveson, 1995)



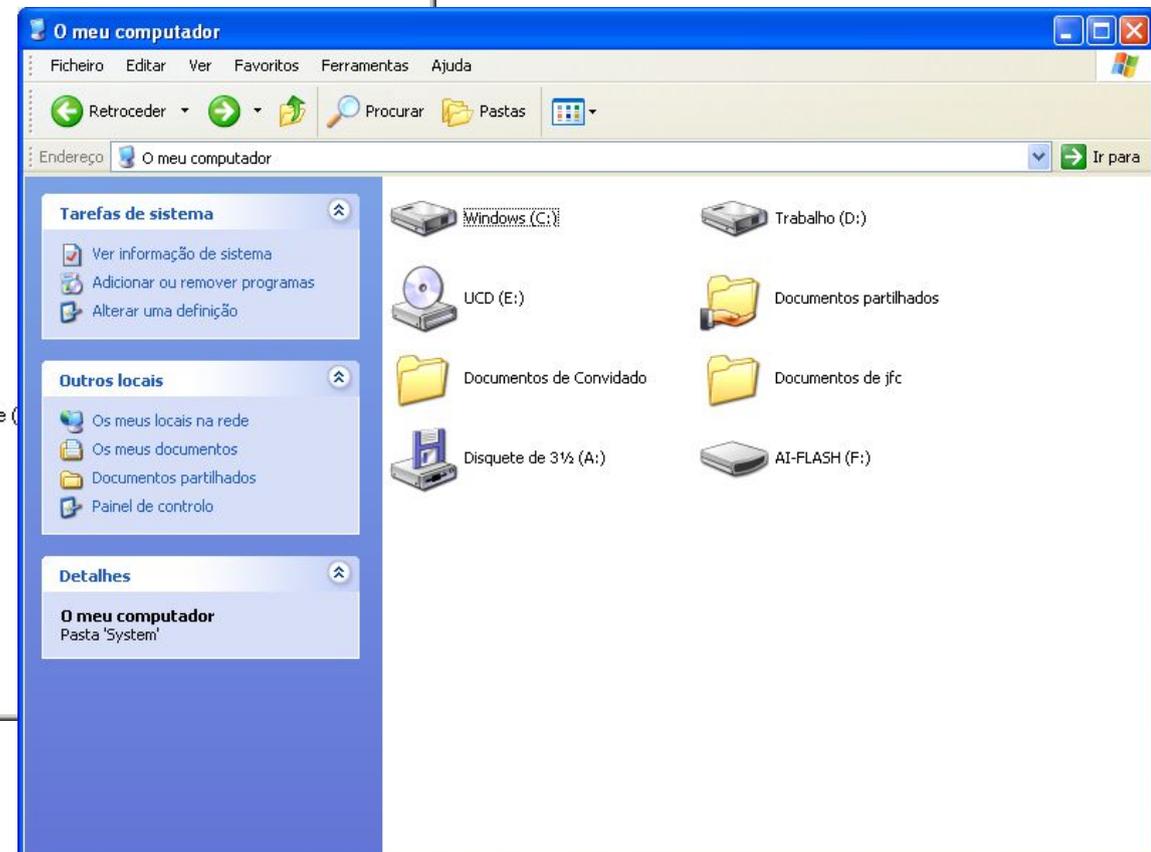
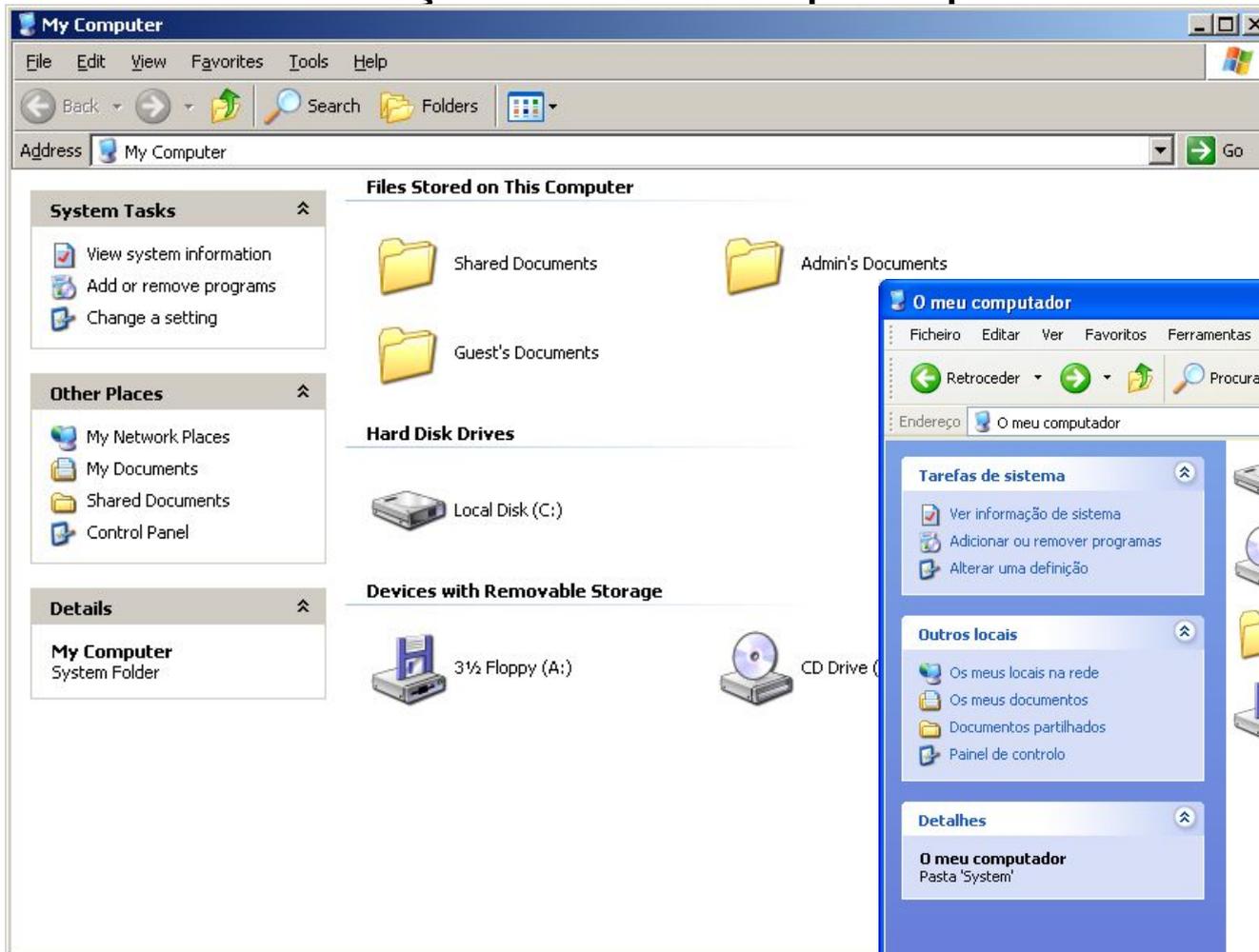
Tipos de Erro

- Enganos
- Slips
 - Erros de captura
 - Erros de descrição
 - Erros *data-driven*
 - Erros de perda de activação
 - Erros de modo
- Soluções para lidar com erros
 - Interlocks
 - Lockins
 - Lockouts



Desafio 2: Fosso da Avaliação

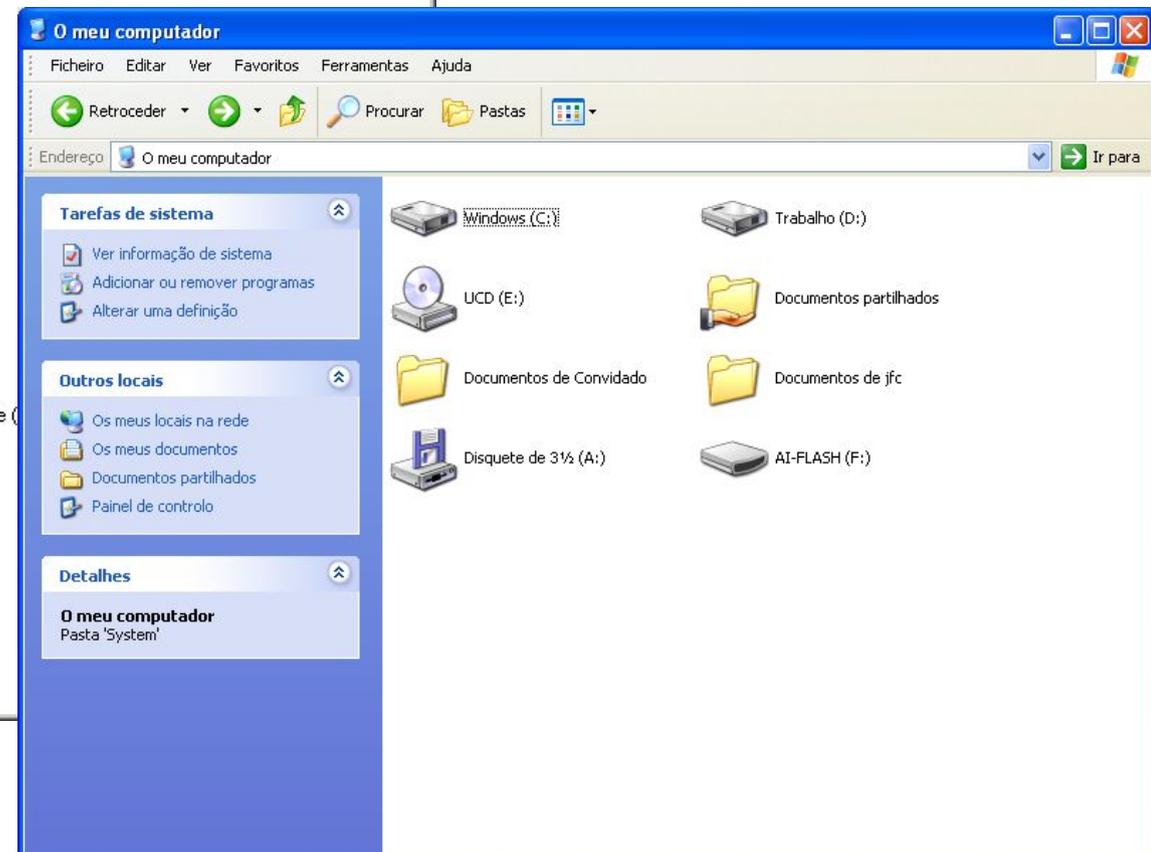
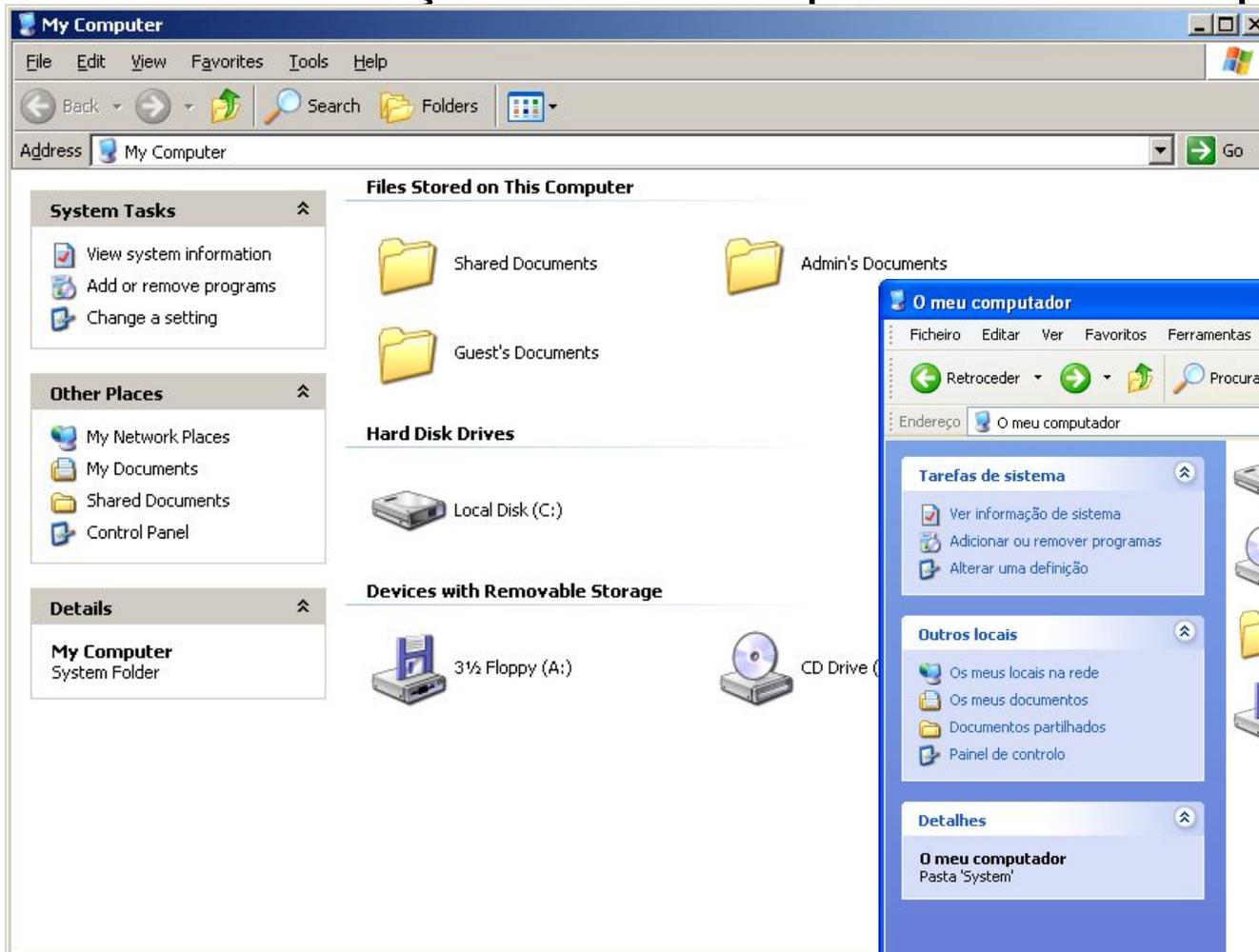
- O esforço necessário para perceber o estado do sistema





Desafio 3: Fosso da Execução

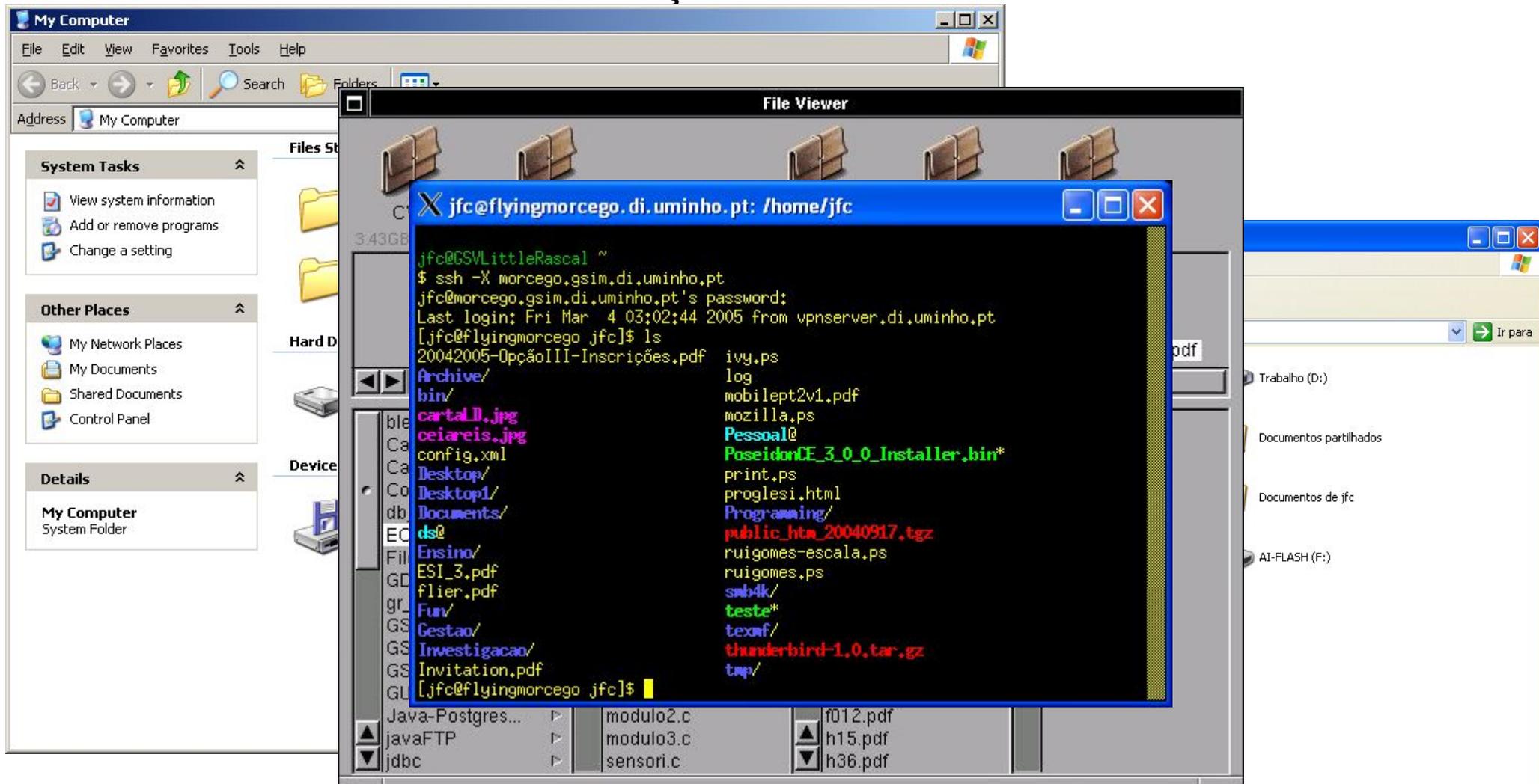
- O esforço necessário para realizar o que se pretende





Soluções

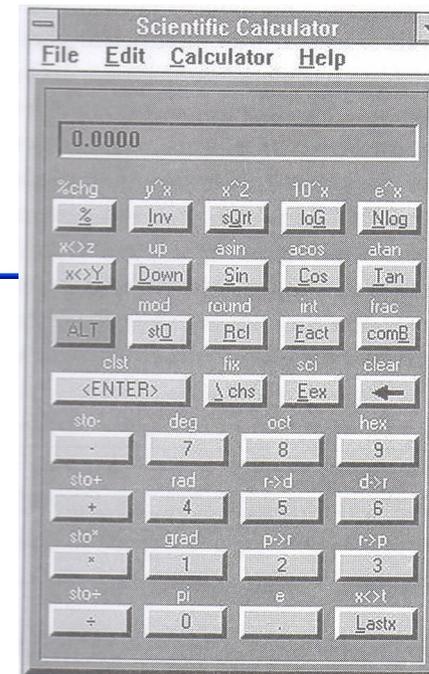
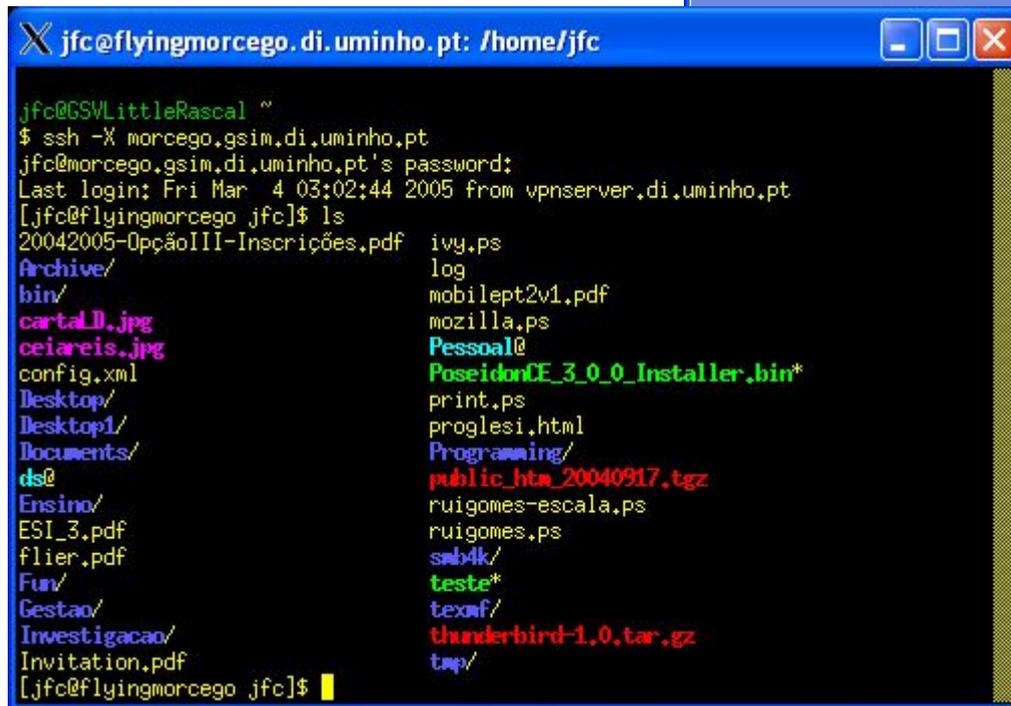
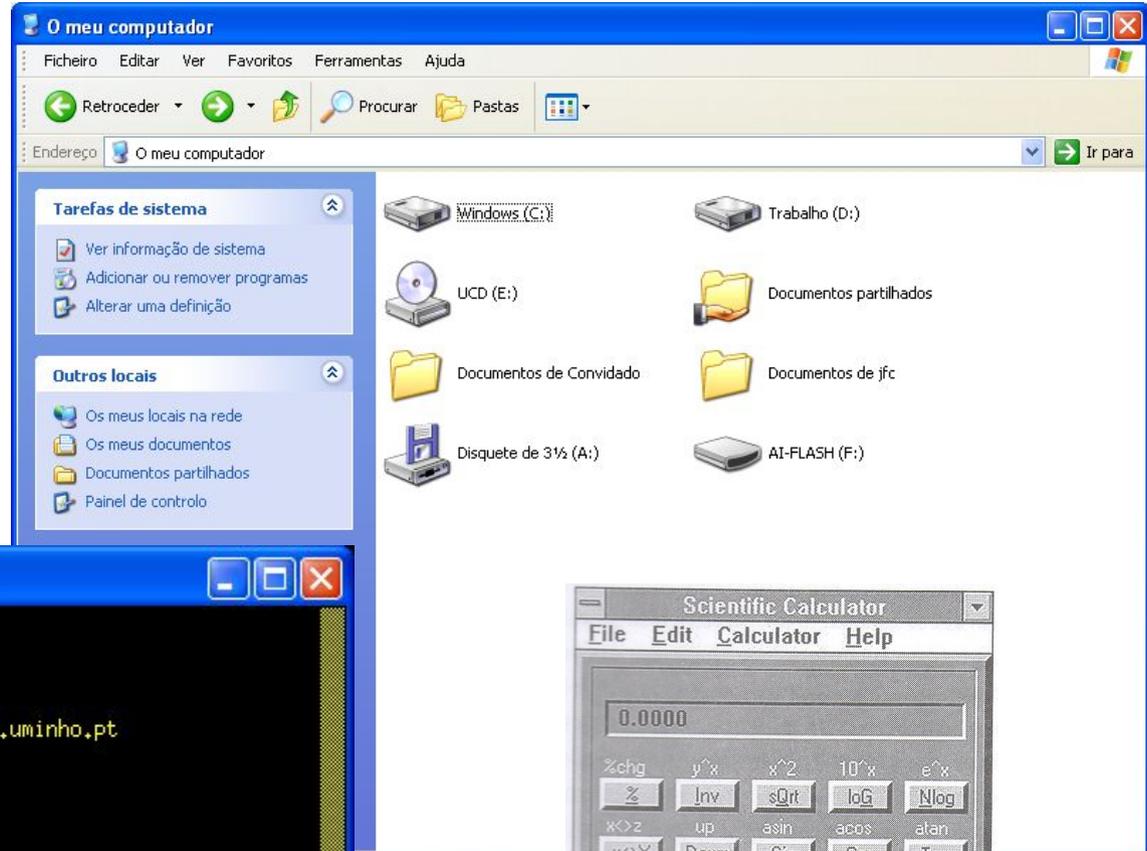
- Conhecimento *na cabeça* vs. Conhecimento no mundo





Soluções

- Metáforas
- Restrições
- *Affordances*





Metáforas

- Permitem explorar conhecimento pré-existente para diminuir os Golfos da Execução e da Avaliação.
(exemplo: processador de texto vs. máquina de escrever)
- Podem surgir problemas:
 - Quando partes da interface divergem da metáfora.
(exemplo: ejectar um disco num Mac!)
 - Quando partes da interface não tem correspondência na metáfora.
(exemplo: máquina de escrever não tem itálico)



Desafio 4: Forma vs. Função

useit.com: Jakob Nielsen on Usability and Web Design - Mozilla Firefox
 File Edit View Go Bookmarks Tools Help
 http://www.useit.com/ Go motores d

maruto - once upon a forest - Mozilla Firefox
 File Edit View Go Bookmarks Tools Help
 http://www.once-upon-a-forest.com/ Go flash menus

useit.com: usable information technology

useit.com: Jakob Nielsen

Permanent Content

Alertbox
 Jakob's column on Web usability

[Ten Best Intranets of 2005](#) (Feb. 28)
 On average, this year's winning intranets use by 149% with designs that supported screens, multinational users, collaborative updated content, and factory-floor work

[User Testing](#) (Feb. 14)
[Teenagers](#) (Jan. 31)
[Usability Guidelines](#) (Jan. 17)

[All Alertbox columns](#) from 1995 to 2005

[Sign up for short newsletter](#) by email when Alertbox is published

Reports

[Intranet usability](#)
 > [Intranet design annual](#) **NEW**

Done

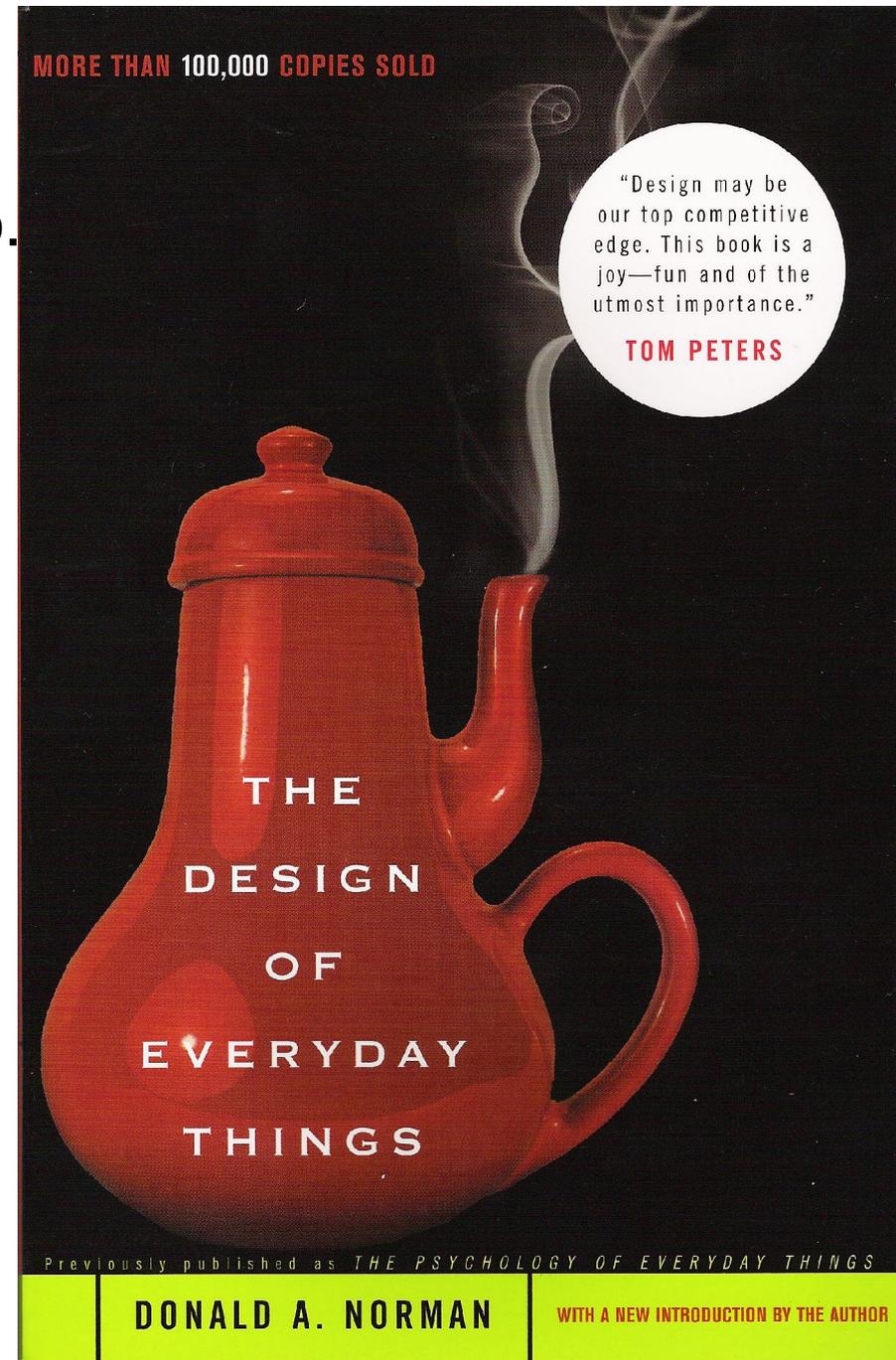
maruto - once upon a forest

Read www.once-upon-a-forest.com



Soluções

- A forma deve que servir a função.
- Identificar função do produto.
- Identificar tarefas.
- Identificar gostos.





Usability isn't skin deep

The image illustrates usability issues in email clients through three screenshots:

- Top Right (Mozilla Thunderbird):** Shows an email client interface with a cluttered menu bar (File, Edit, View, Go, Message, Tools, Help) and a toolbar. The left pane shows a folder tree with 'mail.di' expanded, containing 'Inbox (241)', 'Drafts (2)', 'Sent', 'Trash', 'Junk (141)', 'Para arquivar', 'Searches', and 'Helena (1)'. The main pane shows a message with a subject '[PVS] CFP: APPSEM05 Workshop' and a body containing '[apologies for multiple copies of this CFP]', '1st CALL FOR PAPERS', and '3rd APPSEM II Workshop (APPSEM05) Frauenchiemsee, Germany, 12-15 September 2005'. The status bar at the bottom indicates 'Unread: 241 Total: 975'.
- Bottom Left (Outlook Express):** Shows an email client interface with a menu bar (Ficheiro, Editar, Ver, Ferramentas, Mensagem, Ajuda) and a toolbar. The left pane shows a folder tree with 'Pastas locais' expanded, containing 'A receber', 'A enviar', 'Itens enviados', 'Itens eliminados', and 'Rascunhos'. The main pane shows a message with a subject 'Bem-vindo ao Outlook Express' and a body containing 'Outlook Express A solução para as suas necessidades de processamento de m' and 'Características' with a list of features: 'Correio electrónico e newsgroups' and 'Contas e identidades múltiplas'. The status bar at the bottom indicates '1 mensagens, 0 não lidas'.
- Bottom Right (Outlook Express Contact List):** Shows a contact list in Outlook Express with the text 'Não existem contactos para ver. Clique em 'Contactos' para criar um novo contacto.'



Do it yourself

- 2-3 grupos de (4-5 elementos)
- Um sistema de gestão de horário para o DI
- Pensar/propor:
 - Uma abordagem ao desenvolvimento do sistema que garanta a sua usabilidade.
 - Um sistema de gestão de horários:
 - Uma proposta.
 - Uma justificação para a proposta.
 - (Como demonstrar/comprovar a usabilidade do desenho?)



Usabilidade - Uma definição

- A eficácia, eficiência e satisfação com que utilizadores determinados atingem objectivos determinados em ambientes específicos (Norma ISO DIS 9241-11).
 - *eficácia* - o utilizador consegue realizar as tarefas pretendidas;
 - *eficiência* - o custo de atingir os objectivos é aceitável (em termos de tempo, facilidade de aprendizagem, etc.)
 - *satisfação* - quão confortáveis se sentem os utilizadores com o sistema?

Usabilidade \neq *user friendliness*

- *User friendly*: fácil de usar (mas serve para alguma coisa?)



Engenharia da Usabilidade

- Uma resposta às dificuldades em desenvolver sistemas realmente eficazes e 'usáveis'.
- Um processo que deverá permitir verificar e (idealmente) garantir a usabilidade de um sistema.
- Uma abordagem ao *User Centered Design*
 - Envolver os utilizadores no desenvolvimento do software.
- Uma forma de incluir questões de usabilidade no processo de desenvolvimento de software.
- Muito resumidamente:
 1. Especificar métricas de usabilidade;
 2. Iterativamente Desenvolver e Avaliar (usabilidade).



Processo de Engenharia de Usabilidade

Actividade	Informação produzida	Técnicas
Conhecer os utilizadores	Características dos utilizadores	Entrevistas, questionários e observação
Conhecer as tarefas	Tarefas actuais dos utilizadores	Entrevistas, questionários e observação
	Novo modelo de tarefas	Análise de tarefas
Capturar requisitos dos utilizadores	Requisitos dos utilizadores	Entrevistas, questionários e observação
Definir objectivos de usabilidade	Especificação de usabilidade	
Concepção do produto	Concepção	<i>Guidelines</i>
Avaliação analítica	<i>Feedback</i> para a concepção	Heurísticas, <i>walkthroughs</i>
Prototipagem	Protótipo para testes com os utilizadores	
Avaliação com utilizadores	<i>Feedback</i> para redesenho	Entrevistas, questionários e observação
Redesenho e avaliação com utilizadores	Produto acabado	
Avaliação final com utilizadores	<i>Feedback</i> para o futuro	Entrevistas, questionários e observação

Adaptado de (Faulkner, 2000)



Conhecer os Utilizadores (I)

- Quem são os utilizadores? - tem implicações nos objectivos de usabilidade e no modo como podem ser atingidos.
- Classes de utilizadores:
 - Directos – utilizam o sistema eles próprios;
 - Indirectos – pedem a outros que utilizem o sistema por eles;
 - Remotos – não utilizam o sistema directamente mas dependem de resultados por ele produzidos;
 - de Suporte – são responsáveis pelo bom funcionamento do sistema.



Conhecer os Utilizadores (II)

- Outros pontos a considerar:
 - Os utilizadores são *obrigados* a utilizar o sistema, ou têm poder de escolher?
 - Qual o nível de perícia dos utilizadores?
 - Inexperientes, intermédios ou peritos.
 - Utilizadores intermitentes ou continuados?
- Todos estes aspectos influenciam o tipo de sistema a construir, bem como o tipo de suporte e de treino a fornecer.



Utilizadores Inexperientes

- Pouca ou nenhuma experiência com computadores
- Podem reacear utilizar o sistema
 - sistemas de ajuda simples e eficazes
- Necessitam de feedback frequente
- Preferem ser 'guiados' na interacção
 - 'wizards', noção de tarefa.
- Têm de se sentir seguros que não vão 'estragar' o sistema
 - facilidades de 'undo'.



Utilizadores Peritos

- Sentem-se mais confiantes na sua interacção com o sistema
- São capazes de procurar 'feedback' quando necessitam
 - Menor necessidade de feedback e suporte.
- Preferem ter a possibilidade de utilizar 'short-cuts' e comandos abreviados.
- Preferem interfaces que possam configurar
- Gostam de sentir que detém o controlo da interacção.

Quão perito é um perito? (utilizadores tendem a especializar)



Métodos para recolha de Informação

- Realizar entrevistas
 - O método base. É rápido e não necessita de muito planeamento.
 - Nem sempre é possível ter acesso aos utilizadores, nem sempre as descrições correspondem à realidade.
- Observação directa
 - Captura do modo como actividades são realizadas.
 - Pode interferir com a realização das actividades.
- Integração de utilizadores na equipa de desenvolvimento
 - Presença constante do *cliente* no processo de desenvolvimento.
 - Utilizadores podem ficar 'viciados'
- Questionários
 - Boa fonte de informação subjectiva.
 - Menos fiáveis na recolha de informação objectiva. Fazer um bom questionário não é trivial



Processo de Engenharia de Usabilidade

Actividade	Informação produzida	Técnicas
Conhecer os utilizadores	Características dos utilizadores	Entrevistas, questionários e observação
Conhecer as tarefas	Tarefas actuais dos utilizadores	Entrevistas, questionários e observação
	Novo modelo de tarefas	Análise de tarefas
Capturar requisitos dos utilizadores	Requisitos dos utilizadores	Entrevistas, questionários e observação
Definir objectivos de usabilidade	Especificação de usabilidade	
Concepção do produto	Concepção	<i>Guidelines</i>
Avaliação analítica	<i>Feedback</i> para a concepção	Heurísticas, <i>walkthroughs</i>
Prototipagem	Protótipo para testes com os utilizadores	
Avaliação com utilizadores	<i>Feedback</i> para redesenho	Entrevistas, questionários e observação
Redesenho e avaliação com utilizadores	Produto acabado	
Avaliação final com utilizadores	<i>Feedback</i> para o futuro	Entrevistas, questionários e observação

Adaptado de (Faulkner, 2000)



Análise de Tarefas

- Tarefa – uma sequência de acções executadas para atingir um objectivo.
- Começar por estudar os objectivos e as tarefas existentes.
- Definir como as tarefas serão executadas no novo sistema.
- Tarefas devem ser:
 - Eficazes.
 - Compreensíveis.
 - Satisfatórias.
- Não esquecer que o essencial são os objectivos da interacção e não o modo como eles são atingidos
 - Manter um espírito aberto!
 - Considerar interacções *oportunistas*.



- Para cada tarefa é necessário definir:

Input

Que informação é necessária?
 Quais são as características das fontes de informação?
 Qual é o grau de disponibilidade da informação?
 Que possíveis erros podem ocorrer?
 Quem ou o quê dá início à tarefa?

Output

Quais são os critérios de desempenho?
 O que acontece aos resultados da tarefa?
 De que modo é fornecido *feedback* sobre a execução da tarefa?

Processo

Qual é o processo de decisão?
 Que estratégias existem para esse processo?
 Que competências são necessárias para o processo?
 Que interrupções poderão ocorrer e quando?

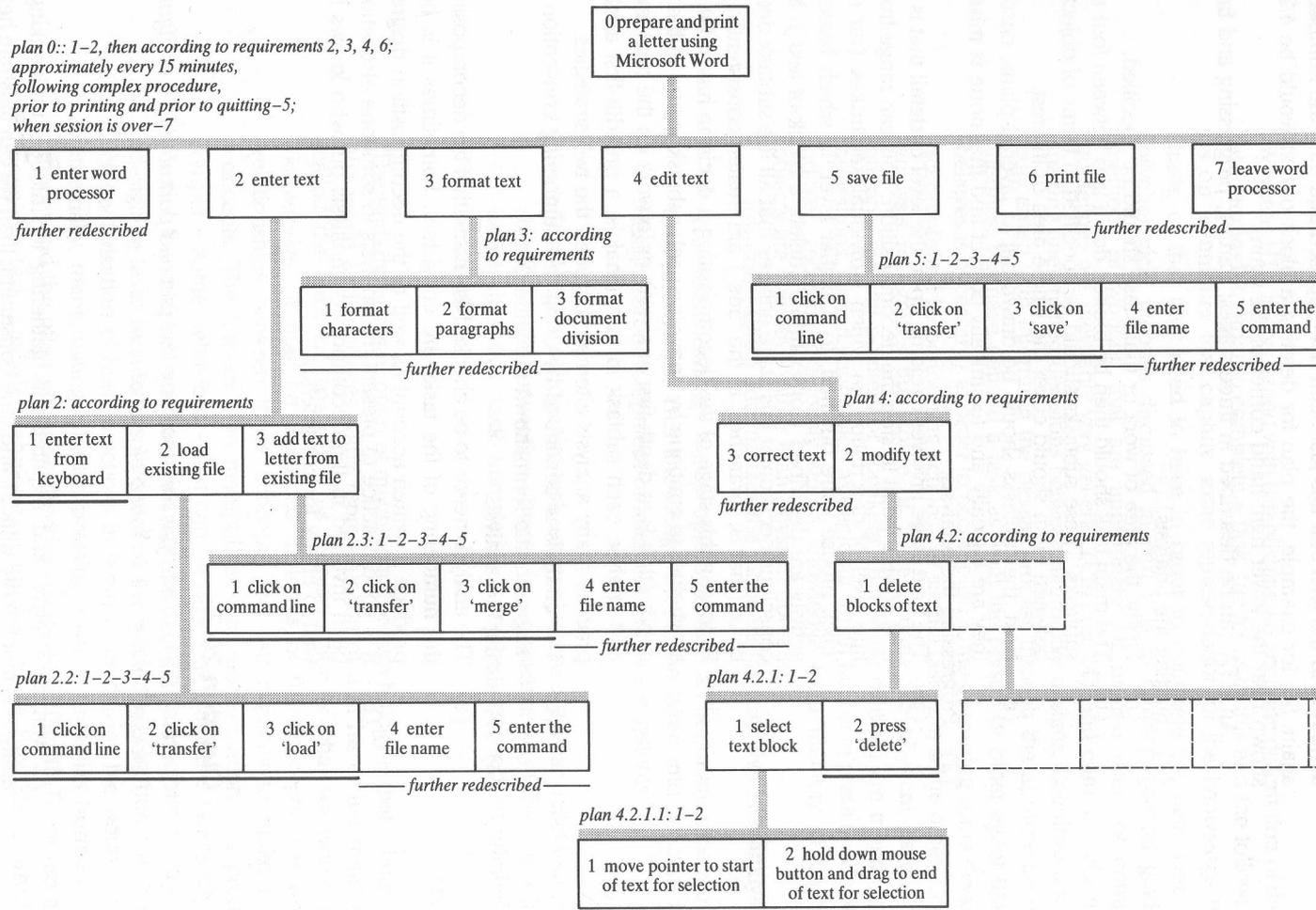
Composição

Qual é a frequência de utilização da tarefa?
 A tarefa depende de outras tarefas?
 O que é uma carga de trabalho normal/anormal?



Modelo de Tarefas

- Abordagem mais comum: *Hierarchical Task Analysis*.



- CTTE/TERESA: <http://giove.cnuce.cnr.it/ctte.html>



Processo de Engenharia de Usabilidade

Actividade	Informação produzida	Técnicas
Conhecer os utilizadores	Características dos utilizadores	Entrevistas, questionários e observação
Conhecer as tarefas	Tarefas actuais dos utilizadores Novo modelo de tarefas	Entrevistas, questionários e observação Análise de tarefas
Capturar requisitos dos utilizadores	Requisitos dos utilizadores	Entrevistas, questionários e observação
Definir objectivos de usabilidade	Especificação de usabilidade	
Concepção do produto	Concepção	<i>Guidelines</i>
Avaliação analítica	<i>Feedback</i> para a concepção	Heurísticas, <i>walkthroughs</i>
Prototipagem	Protótipo para testes com os utilizadores	
Avaliação com utilizadores	<i>Feedback</i> para redesenho	Entrevistas, questionários e observação
Redesenho e avaliação com utilizadores	Produto acabado	
Avaliação final com utilizadores	<i>Feedback</i> para o futuro	Entrevistas, questionários e observação

Adaptado de (Faulkner, 2000)



Analisar Requisitos

- Os requisitos definem o sistema a construir.
- Tipos de requisitos:
 - Funcionais – o que deve o sistema fazer?
 - de Dados – que dados são manipulados pelo sistema?
 - de Usabilidade – que características de usabilidade deve o sistema ter?
 1. Identificar **utilizadores**.
 2. Identificar **tarefas**.
 3. Definir **objectivos de usabilidade**.
- Outros requisitos Não Funcionais.



Processo de Engenharia de Usabilidade

Actividade	Informação produzida	Técnicas
Conhecer os utilizadores	Características dos utilizadores	Entrevistas, questionários e observação
Conhecer as tarefas	Tarefas actuais dos utilizadores Novo modelo de tarefas	Entrevistas, questionários e observação Análise de tarefas
Capturar requisitos dos utilizadores	Requisitos dos utilizadores	Entrevistas, questionários e observação
Definir objectivos de usabilidade	Especificação de usabilidade	
Concepção do produto	Concepção	<i>Guidelines</i>
Avaliação analítica	<i>Feedback</i> para a concepção	Heurísticas, <i>walkthroughs</i>
Prototipagem	Protótipo para testes com os utilizadores	
Avaliação com utilizadores	<i>Feedback</i> para redesenho	Entrevistas, questionários e observação
Redesenho e avaliação com utilizadores	Produto acabado	
Avaliação final com utilizadores	<i>Feedback</i> para o futuro	Entrevistas, questionários e observação

Adaptado de (Faulkner, 2000)



Métricas de Usabilidade

Eficácia

Rácio de sucesso / insucesso na realização das tarefas
Frequência de utilização dos vários comandos ou funcionalidades específicas
Quantificação dos problemas sentidos pelos utilizadores
Qualidade dos resultados obtidos na execução das tarefas

Eficiência

O tempo necessário para efectuar determinadas tarefas
O número de acções requeridas para realizar uma tarefa
O tempo gasto a consultar a documentação
O tempo gasto a utilizar ajuda on-line
O tempo gasto a resolver erros

Satisfação

Medição da satisfação utilizando questionários



Especificação de Usabilidade

Especificação de Usabilidade						
<i>Utilizadores: todos</i>						
<i>Pré-condições: medições a realizar após uma semana de utilização</i>						
		Pior caso	Nível mínimo aceitável	Nível planeado	Melhor nível	Nível actual
Atributo	Medição					
Eficiência	Erros em tarefa	5 ou mais	2	1	0	3
Eficiência	Tempo necessário	10 ou mais	5 minutos	3 minutos	1 minuto	10 minutos
...

Adaptado de (Faulkner, 2000)



Processo de Engenharia de Usabilidade

Actividade	Informação produzida	Técnicas
Conhecer os utilizadores	Características dos utilizadores	Entrevistas, questionários e observação
Conhecer as tarefas	Tarefas actuais dos utilizadores	Entrevistas, questionários e observação
	Novo modelo de tarefas	Análise de tarefas
Capturar requisitos dos utilizadores	Requisitos dos utilizadores	Entrevistas, questionários e observação
Definir objectivos de usabilidade	Especificação de usabilidade	
Concepção do produto	Concepção	<i>Guidelines</i>
Avaliação analítica	<i>Feedback</i> para a concepção	Heurísticas, <i>walkthroughs</i>
Prototipagem	Protótipo para testes com os utilizadores	
Avaliação com utilizadores	<i>Feedback</i> para redesenho	Entrevistas, questionários e observação
Redesenho e avaliação com utilizadores	Produto acabado	
Avaliação final com utilizadores	<i>Feedback</i> para o futuro	Entrevistas, questionários e observação

Adaptado de (Faulkner, 2000)



Concepção do Produto (I)

- **Modelação de Informação**
 - Desenhar uma interface que suporte as tarefas identificadas.
 - Para cada tarefa, que informação é relevante?
 - Como representar o estado do sistema?
 - Que metáforas utilizar?
 - Consistência da interface
 - interna, externa e metafórica
 - Atenção ao *overdesign!*
 - *The inch the foot and the yard...*



Concepção do Produto (II)

- **Modelação da Interação**
 - Desenhar uma interface que suporte as tarefas identificadas.
 - Que tecnologia utilizar?
 - *Affordances*
 - *Feedback*
 - Tratamento de erros
 - Suporte à exploração (*undo*)
 - Alocação de funções
 - Quem faz o quê na interação?
 - *Interação implícita, integração física digital*



Concepção do Produto (III)

- **Guidelines**
 - Capturam “*best practice*”
 - Duas origens: psicologia e experiência prática
 - Não confundir com regras de desenho
 - Utilidade das *guidelines*:
 - Novos conceitos e estratégias;
 - Auxílio na escolha de alternativas;
 - Auxílio na avaliação.
 - Limitações das *guidelines*:
 - Como escolher mais apropriada?
 - Como aplicá-las correctamente?
 - Como aplicar múltiplas *guidelines*? (potencialmente contraditórias)



Concepção do Produto (IV)

- **Algumas *Guidelines* (www.emmus.org)**
 - Forneça um desenho simples
 - Procure a consistência (?)
 - Forneça *feedback* informativo
 - Minimize a carga de memória do utilizador
 - Forneça “fecho” (*closure*)
 - Forneça *short cuts* para utilizadores frequentes
 - Forneça um bom suporte para o tratamento de erros
 - Forneça *undo* e pontos de saída claramente identificados
- Algumas fontes de *Guidelines*:
 - Designing the User Interface: Strategies for Effective Human-Computer Interaction. B. Shneiderman (1992) Addison-Wesley.
 - Principles and Guidelines in Software User Interface Design. D. J. Mayhew, Prentice-Hall.



Processo de Engenharia de Usabilidade

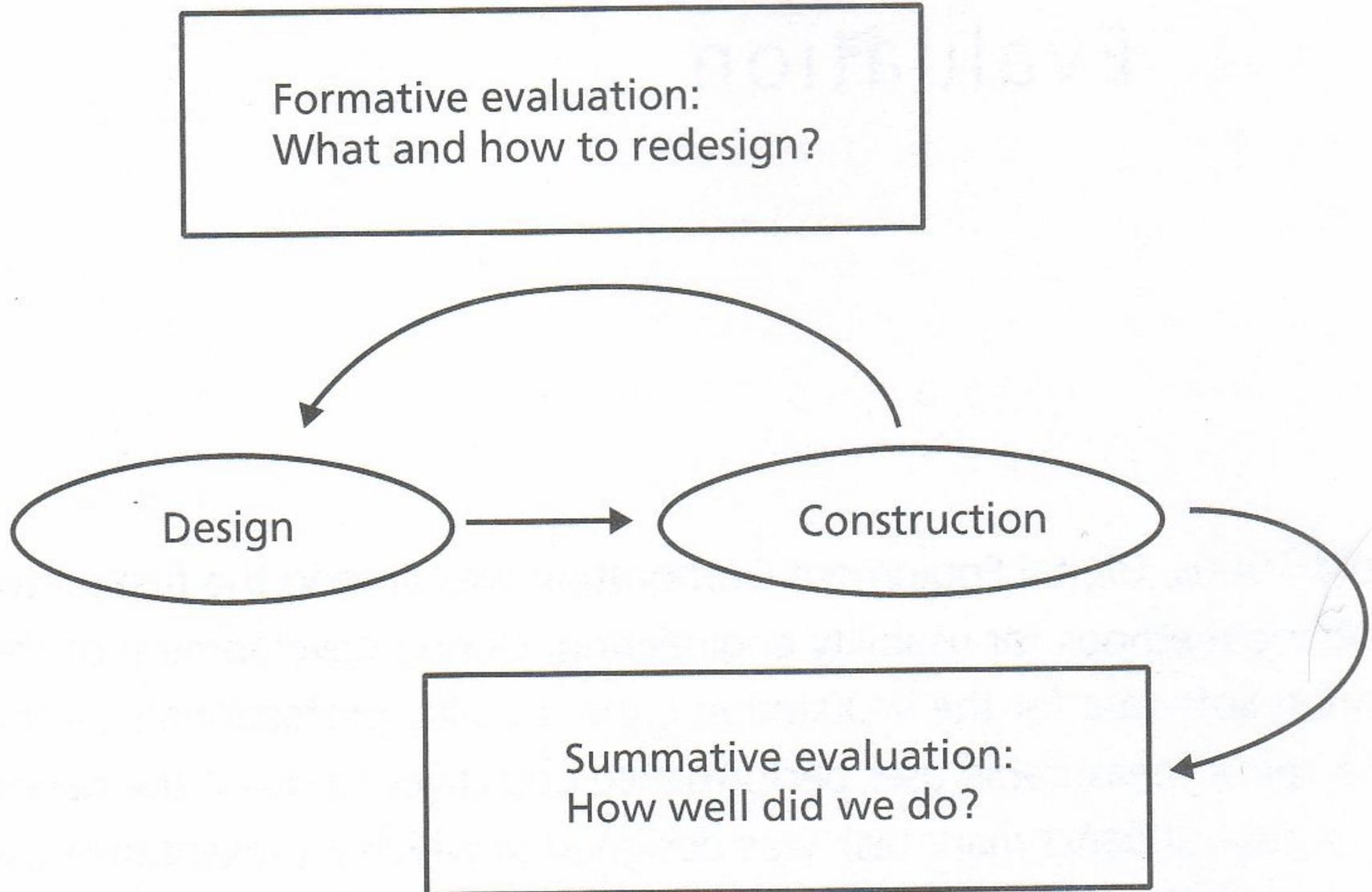
Actividade	Informação produzida	Técnicas
Conhecer os utilizadores	Características dos utilizadores	Entrevistas, questionários e observação
Conhecer as tarefas	Tarefas actuais dos utilizadores	Entrevistas, questionários e observação
	Novo modelo de tarefas	Análise de tarefas
Capturar requisitos dos utilizadores	Requisitos dos utilizadores	Entrevistas, questionários e observação
Definir objectivos de usabilidade	Especificação de usabilidade	
Concepção do produto	Concepção	<i>Guidelines</i>
Avaliação analítica	<i>Feedback</i> para a concepção	Heurísticas, <i>walkthroughs</i>
Prototipagem	Protótipo para testes com os utilizadores	
Avaliação com utilizadores	<i>Feedback</i> para redesenho	Entrevistas, questionários e observação
Redesenho e avaliação com utilizadores	Produto acabado	
Avaliação final com utilizadores	<i>Feedback</i> para o futuro	Entrevistas, questionários e observação

Adaptado de (Faulkner, 2000)



Avaliação

- **Que tipo de avaliação?**





Técnicas de Avaliação

- **Métodos empíricos** (experimentais)
 - Prototipagem e testes com utilizadores reais.
 - Úteis para validar o sistema sob condições reais de utilização.
 - Normalmente dispendiosos, requerem muito tempo e organização.

- **Métodos analíticos**
 - Baseados na análise de modelos dos utilizadores/sistema.
 - Úteis na validação de decisões nas fases iniciais de desenvolvimento.
 - Tradicionalmente utilizados de forma manual e, mais ou menos, informal.



Métodos empíricos (I)

- O principal objectivo é avaliar a usabilidade do sistema a partir da observação de como ele é utilizado por utilizadores finais.
 - Por este motivo são também chamados de **métodos interpretativos**.
- O custo de aplicação é normalmente muito elevado:
 - Necessidade de recrutar utilizadores **adequados**;
 - Necessidade de obter condições de utilização realistas.
- Podem ser realizados:
 - com o sistema final – avaliação *final*;
 - com um protótipo – avaliação formativa.



Métodos empíricos (II)

- Classes de métodos
 - Estudos de campo contextual enquiry, análise participativa, abordagem etnográfica
 - Análises em laboratório think aloud, avaliação cooperativa, *wizard of oz*
- Problema sério
 - Estudos de campo – até que ponto o método interfere com as condições reais de utilização?
 - Análises em laboratório – até que ponto o método cria condições reais de utilização?



Métodos analíticos (I)

- A análise é realizada sem recorrer a utilizadores reais – métodos **baseados em modelos**.
- O custo de aplicação é muito mais baixo que o dos métodos empíricos.
- O principal objectivo é prever potenciais problemas de usabilidade (o que é diferente de avaliar a usabilidade).
 - Por este motivo são também chamados de **métodos preditivos**.



Métodos analíticos (II)

- **Principais métodos**

- Métodos mais formais

- *Keystroke Level Analysis*
Previsão do desempenho dos utilizadores.
- ...

- Métodos menos formais

- Avaliação Heurística
Método de inspecção *manual* baseado em *guidelines*.
- *Cognitive Walkthroughs*
Método de inspecção *manual* baseado num modelo de aprendizagem exploratória.



Keystroke Level Analysis (I)

- Permite analisar desempenho na realização de uma tarefa conhecida.
- Aplicável quando o método de operação é bem conhecido.
- Útil para comparar o desempenho previsto de métodos de operação alternativos.
- Cada operação é dividida em componentes e a cada componente está associado um tempo de execução.
- Utilizam-se heurísticas para introduzir componentes de “preparação mental” (pausas).



Keystroke Level Analysis (II)

- Operadores:
 - K premir tecla do teclado (de .08 a 1.10)
 - B premir botão do rato (premir/largar .10 / clique .20)
 - P apontar com o rato (média 1.1 – utilizar Lei de Fitt)
 - H mover a mão para o rato/teclado (.40)
 - M preparação mental (1.35)
- Procedimento
 - Especificar o comportamento com os operadores.
 - Calcular o tempo de execução
- Exemplo:
 - Fechar uma aplicação em Windows



Avaliação Heurística

- Uma **equipa de avaliadores** realiza a análise.
 - Os avaliadores devem ser exteriores à equipa de desenvolvimento.
 - Cada avaliador realiza a sua análise de forma independente (os resultados são agregados no final).
- Um **conjunto de heurísticas** é utilizado para guiar a análise.
 - Heurística: Metodologia que conduz à sabedoria científica ou à resolução de problemas.
 - Neste contexto, *heurísticas* podem ser vistas como guidelines de alto-nível.
- Procedimento:
 - Inspeccionar o fluxo da interface de écran para écran;
 - Inspeccionar cada écran face às heurísticas.

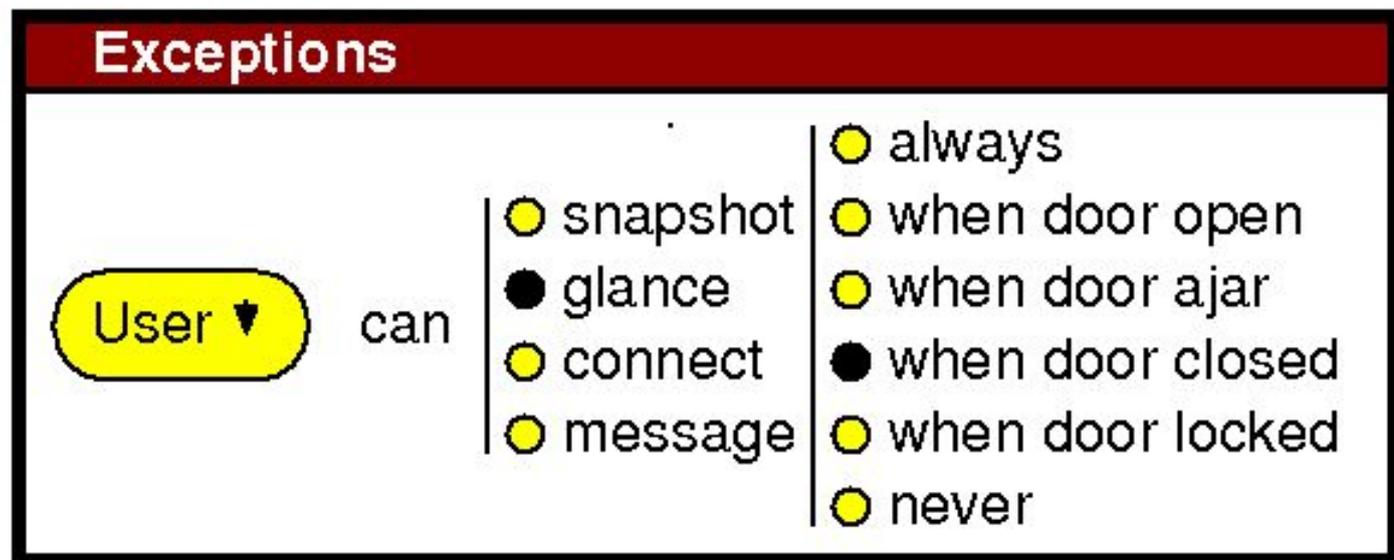
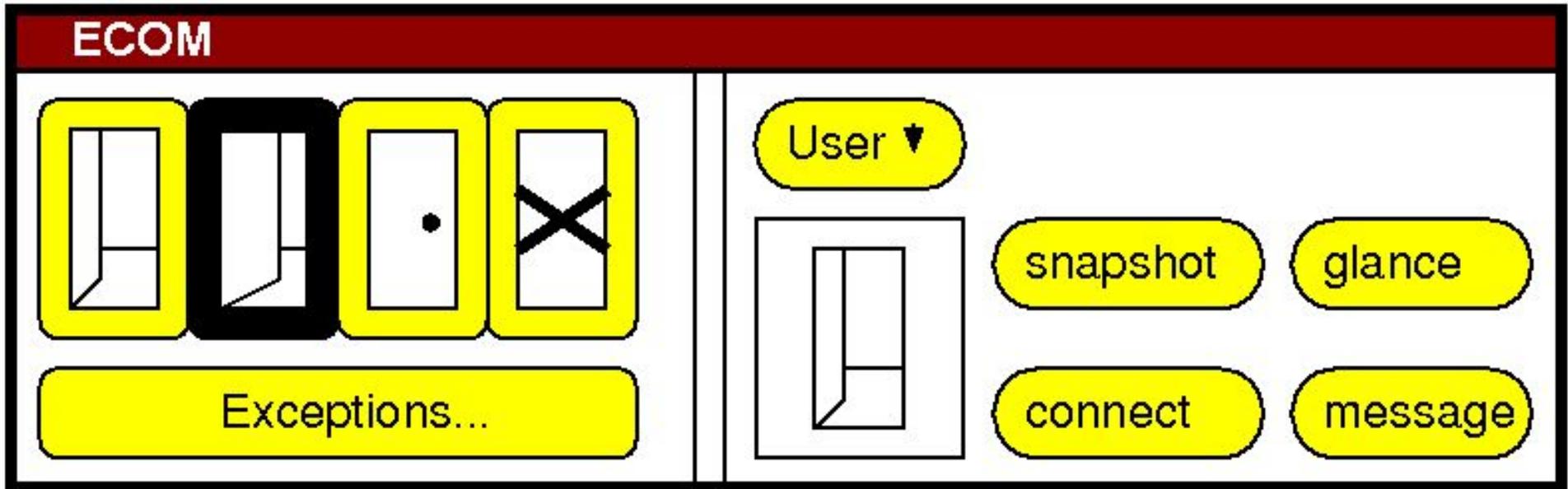


Heurísticas de Nielsen

- Utilizar diálogo simples e natural.
- Falar a linguagem dos utilizadores.
- Minimizar a carga de memória dos utilizadores.
- Ser consistente.
- Fornecer *feedback*.
- Fornecer saídas claramente assinaladas.
- Fornecer atalhos (*short-cuts*).
- Fornecer boas mensagens de erro.
- Prevenir erros.



Exemplo





Cognitive Walkthroughs (I)

- Um método para analisar a usabilidade de sistemas em termos de *aprendizagem exploratória*.
- Útil para analisar sistemas que vão ser utilizados sem treino prévio.
- Fornece resposta à questão:
 - Até que ponto vai este sistema *guiar* um utilizador não treinado na execução de uma tarefa?



Cognitive Walkthroughs (II)

- **Procedimento**

- O perito coloca-se no lugar do utilizador e procura executar uma dada tarefa enquanto vai respondendo às seguintes questões:
 1. A acção correcta é suficientemente evidente para o utilizador?
 2. Irá o utilizador associar a acção correcta ao que pretende fazer?
 3. Irá o utilizador interpretar de forma correcta a resposta do sistema à acção escolhida? (saberá se fez a escolha certa?)



Avaliação – Conclusão

- Métodos empíricos são os mais dispendiosos, mas também aqueles em que mais problemas são identificados.
- Técnicas de *walkthrough* são úteis nas fases iniciais de desenvolvimento.
- Avaliação heurística é pouco dispendiosa e eficaz, mas deve ser realizada por mais que um perito.
- Os métodos baseados em modelos formais permitem obter resultados quantitativos, mas podem ser de difícil aplicação (muito trabalho nesta área tem ainda que ser realizado).



Onde procurar mais informação...

- www.usabilitynet.org - projecto financiado pela UE.
- www.usabilitynews.com - British HCI Group.
- www.upassoc.org - Usability Professional's Association.
- www.usabilityfirst.com - mantido pela DBD (uma consultora).
- *Interactive Systems Design*. Newman & Lamming, Addison-Wesley, 1995.
- *Human-Computer Interaction, third edition*. Dix et al., Pearson/Prentice-Hall, 2004.
- *Usability Engineering*. Kristine Faulkner, Palgrave, 2000.



Referências

- Allwood, C. (1984) Analysis of the field survey in Allwood & Lieff (eds.) *Better Terminal Use*, pp. 72-77. Syslab-G.
- Jordan, P. & Thomas, B. & Weermeester, A. & McClelland, I. (1996) *Usability Evaluation in Industry*, Taylor and Francis.
- Jordan, P (1998) *An Introduction to Usability*. Taylor and Francis.
- Nielsen, J. (1993) *Usability Engineering*. AP Professional.
- Hollnagel, E. (1993) *Human reliability analysis: context and control*. Academic Press.
- MacKenzie, D. (1994) Computer Related Accidental Death: an empirical exploration. *Science and Public Policy*. 21(4), pp. 233-248.
- Leveson, N. (1995) *Safeware: System Safety and Computers*. Addison-Wesley.
- Norman, D. (1988) *The Design of Everyday Things*. Basic Books.
- Faulkner, X. (2000) *Usability Engineering*. Palgrave.
- Barnard, P.J. (1985) Interacting Cognitive Subsystems: A psycholinguistic approach to short term memory. In A. Ellis (Ed.) *Progress in the Psychology of Language*, (Vol. 2), Chapter 6, London: Lawrence Erlbaum Associates, 197-258.
- Moran, P. (1981) The command language grammar: a representation for the user interface of interactive systems. *International Journal of Man-Machine Studies*, 15(1), 3-50.
- Baecker, R. & Buxton, W., eds. (1987) *Readings in Human-Computer: A Multi-disciplinary Approach*. Morgan Kaufmann.