



**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](mailto:jose.campos@di.uminho.pt)  
Departamento de Informática, Universidade do Minho, Braga, Portugal.

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

*Interacção Humano-Computador para Ambientes Ubíquos – Módulo 1*



## Bibliografia, etc.

- Livros
  - Human-Computer Interaction (3<sup>rd</sup> 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](http://www.usabilitynews.com) (British HCI Group)
  - [www.usabilitynet.org](http://www.usabilitynet.org) (Projecto financiado pela UE)
  - [www.usabilityfirst.com](http://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 interface<sup>1</sup>.

### 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?

<sup>1</sup> 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](http://www.ibm.com/easy))
  - Usage-centered design (Constantine & Lockwood - [www.foruse.com](http://www.foruse.com))



## Uma área multi-disciplinar



## Usabilidade - Uma definição

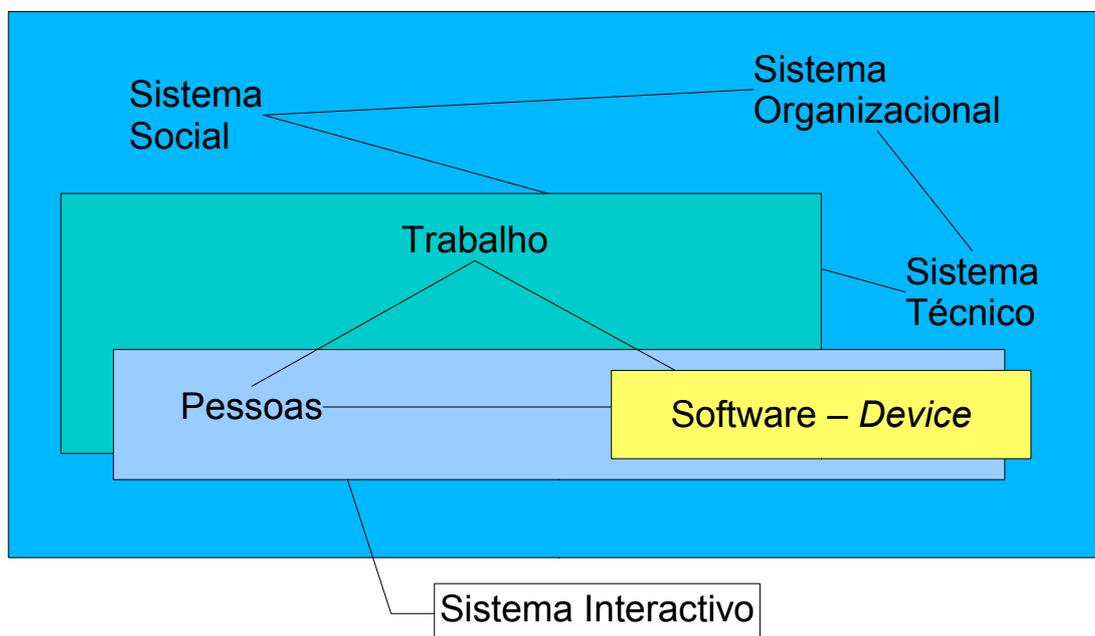
- A **eficácia**, **eficiência** e **satisfação** com **que utilizadores determinados atingem objectivos determinados em ambientes específicos** (Norma ISO DIS 9241-11).
  - **eficácia** - **o utilizador consegue realizar as tarefas pretendidas**;
  - **eficiência** - o custo de **atingir os objectivos** 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



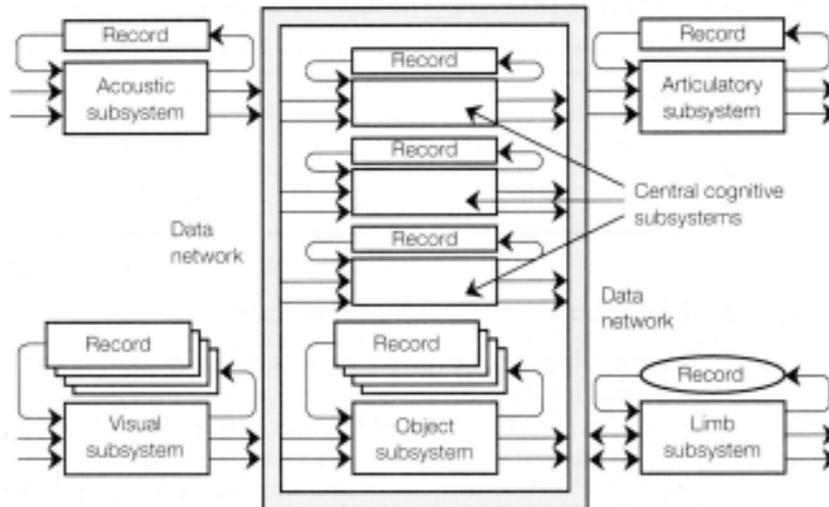
## É necessário compreender

- 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**

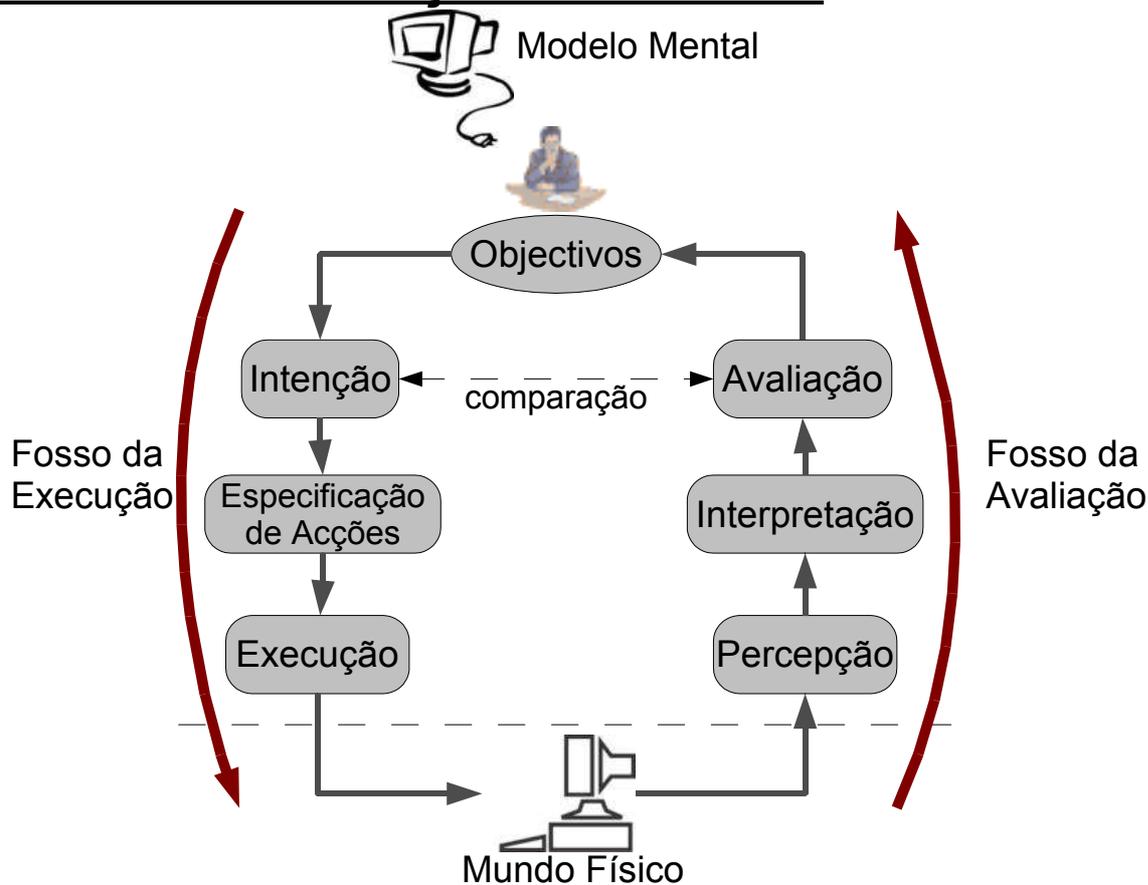
operation locate-cursor (location: L)  
user-purpose: cursor-at(L)  
subgoaling-precond: locate(L)  
action: locate-at(L)  
operation mark-text-at-loc (text: TX, location: L)  
user-purpose: is-marked(TX, L)  
subgoaling-precond: cursor-at(L)  
filtering-precond: text-at(TX, L)  
action: mark-text(TX)

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-constains(TX)  
action: copy-to-buffer  
operation oaste-text-at-loc (text: TX, location: L)  
user-purpose: text-at(TX,L)  
subgoaling-precond: buffer-constains(TX)  
cursor-at(L)  
action: paste-text

- Assume “comportamento racional”.



# Modelo de Interacção 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*



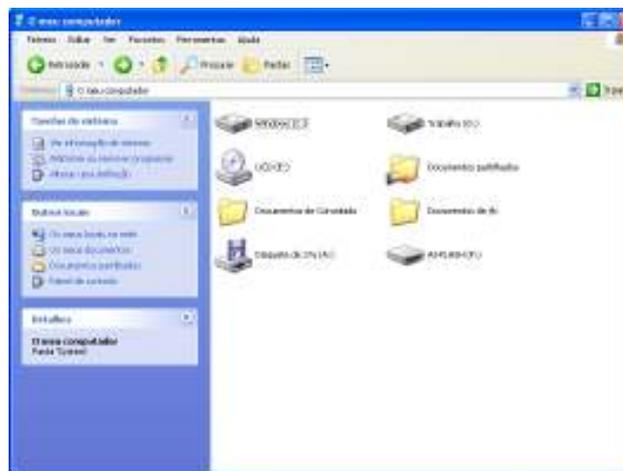
## Desafio 1: Errar é Humano!

- “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 atribuídas 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)



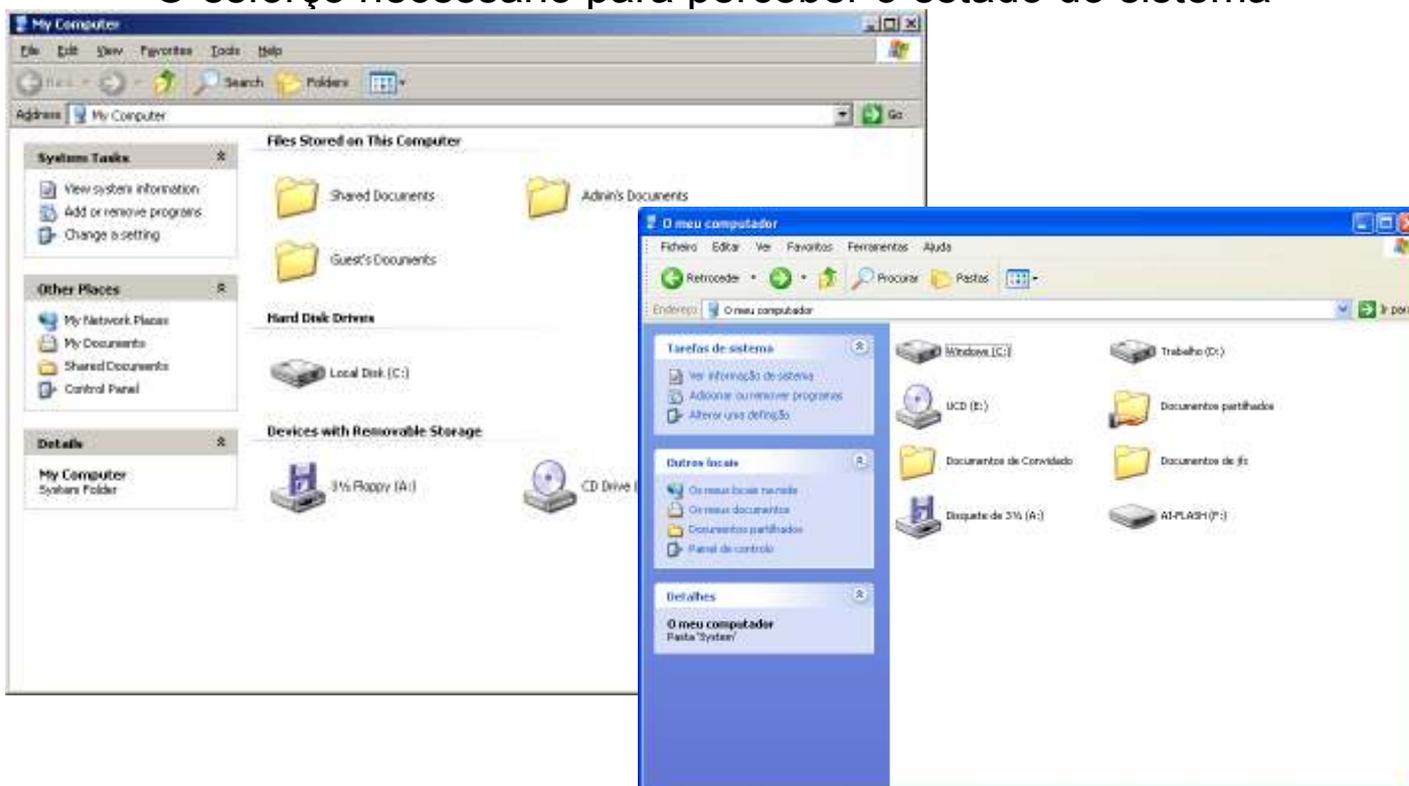
## Tipos de Erro

- Enganos
- Lapsos
  - Erros de captura
  - Erros de descrição
  - Erros *data-driven*
  - Erros de activação associativa
  - Erros de perda de activação
  - Erros de modo
- Soluções para lidar com erros (*forcing functions*)
  - *Interlocks* – forçam sequência
  - *Lockins* – mantém processo activo até poder ser terminado
  - *Lockouts* – impedem acções ilegais



## Desafio 2: Fosso da Avaliação

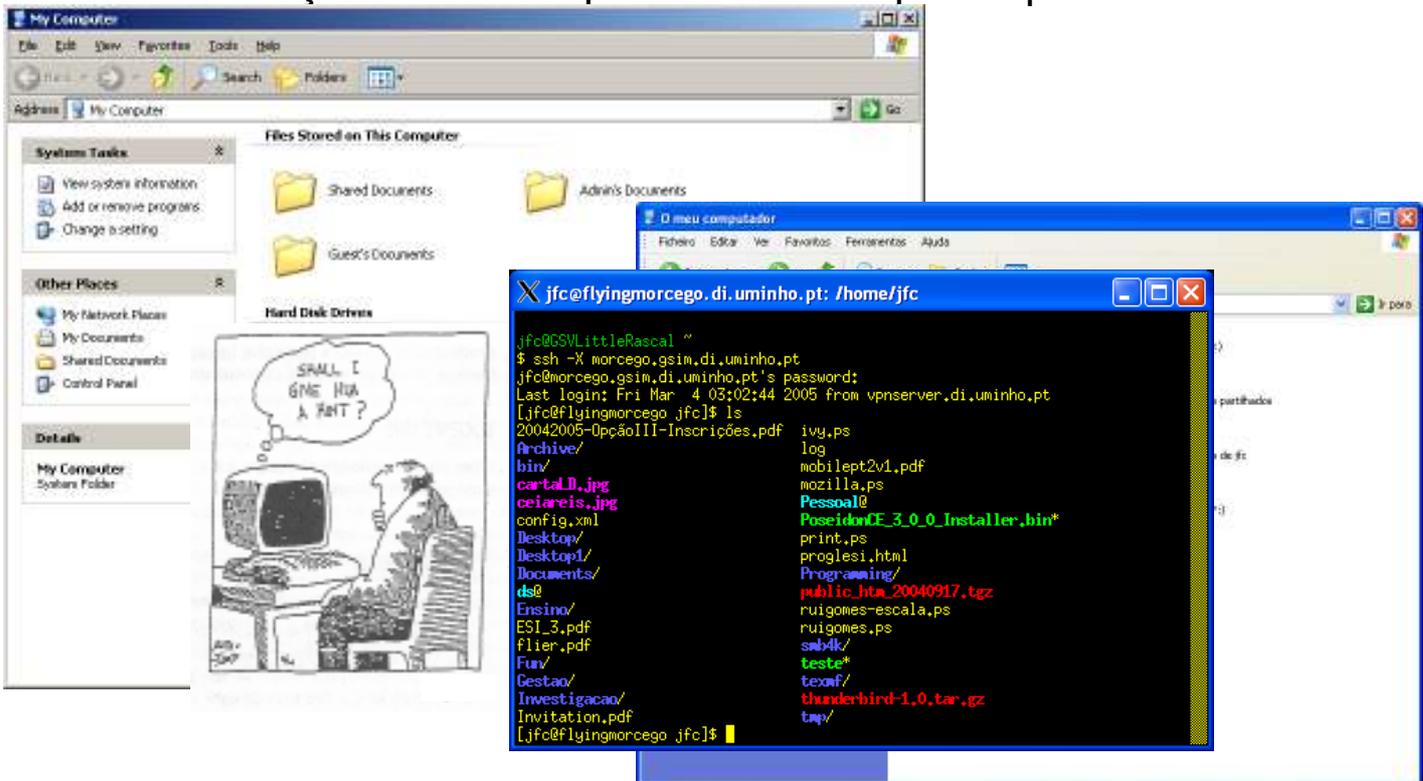
- O esforço necessário para perceber o estado do sistema





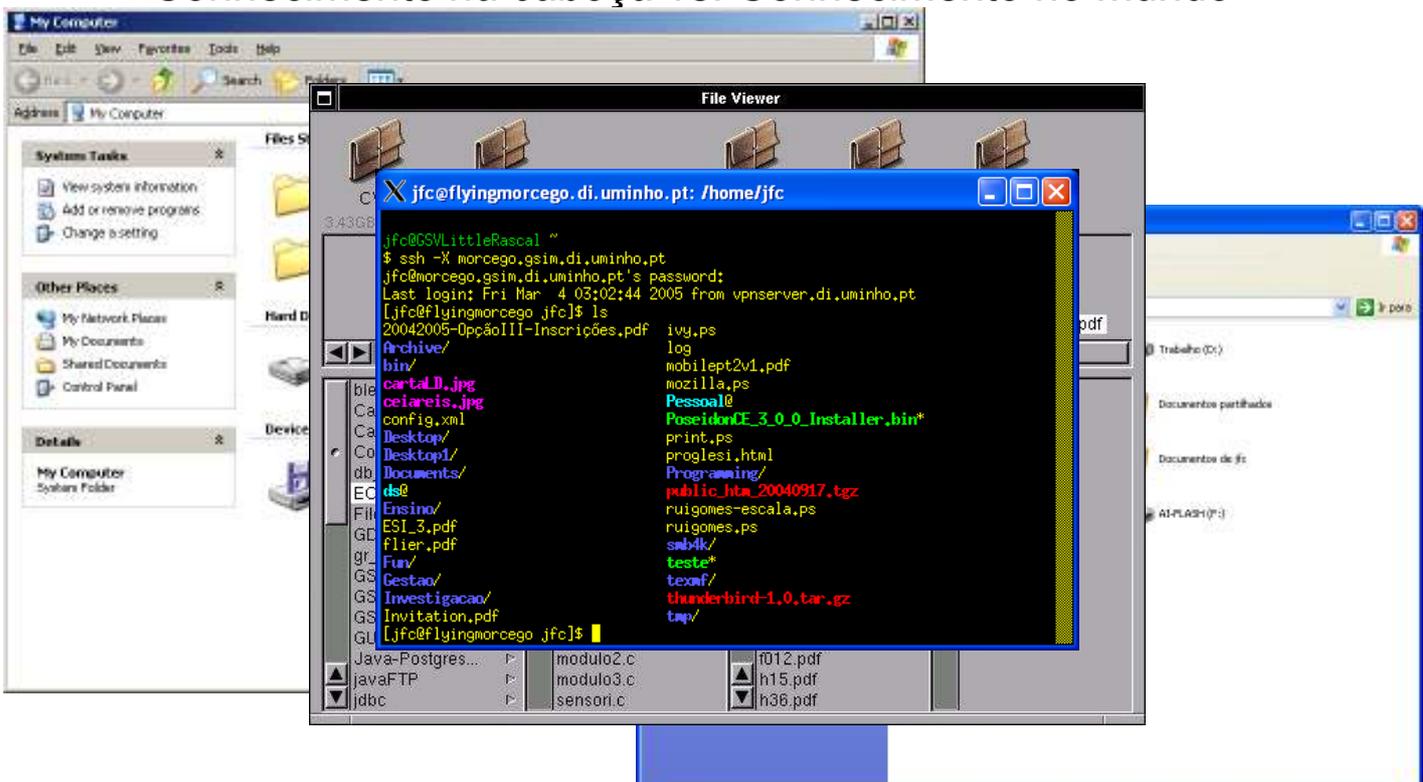
## Desafio 3: Fosso da Execução

- O esforço necessário para realizar o que se pretende



## Soluções

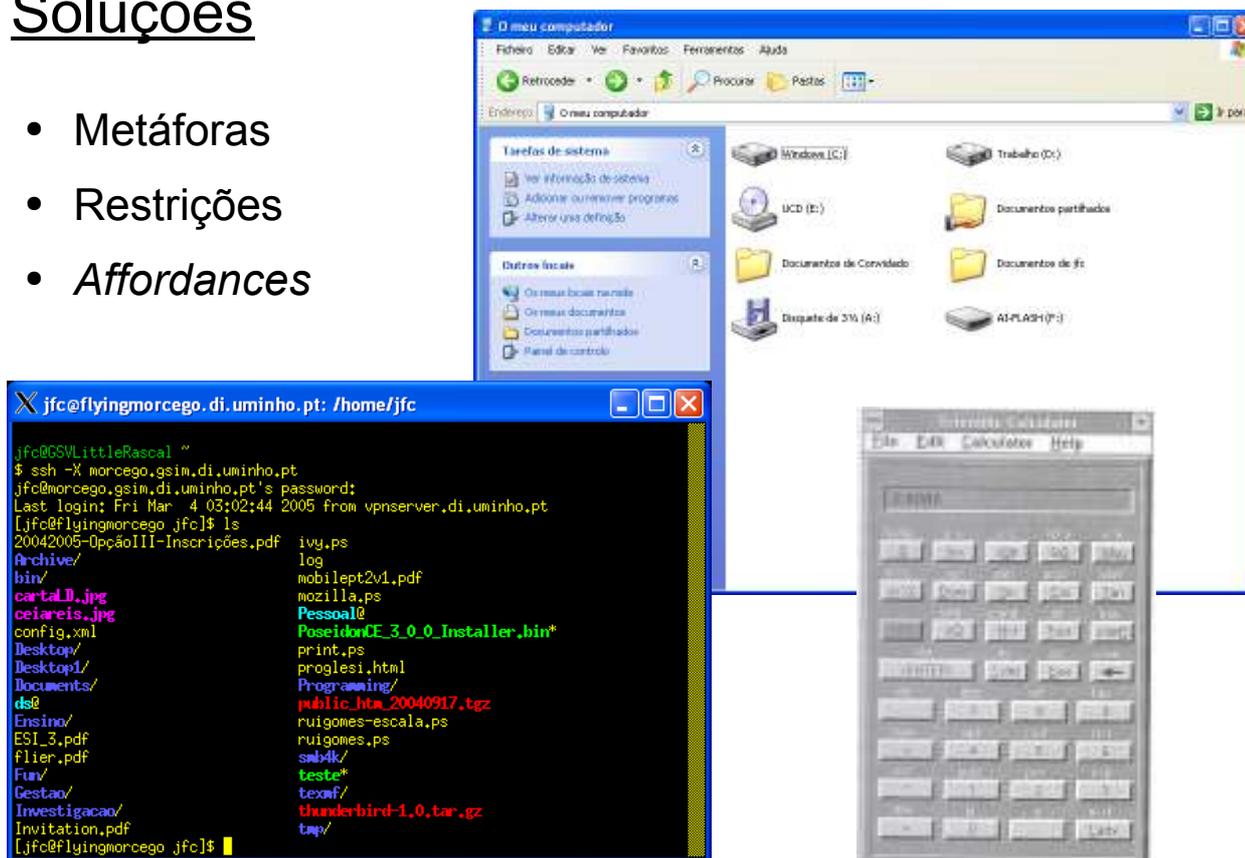
- Conhecimento *na cabeça* vs. Conhecimento no mundo





## Soluções

- Metáforas
- Restrições
- *Affordances*

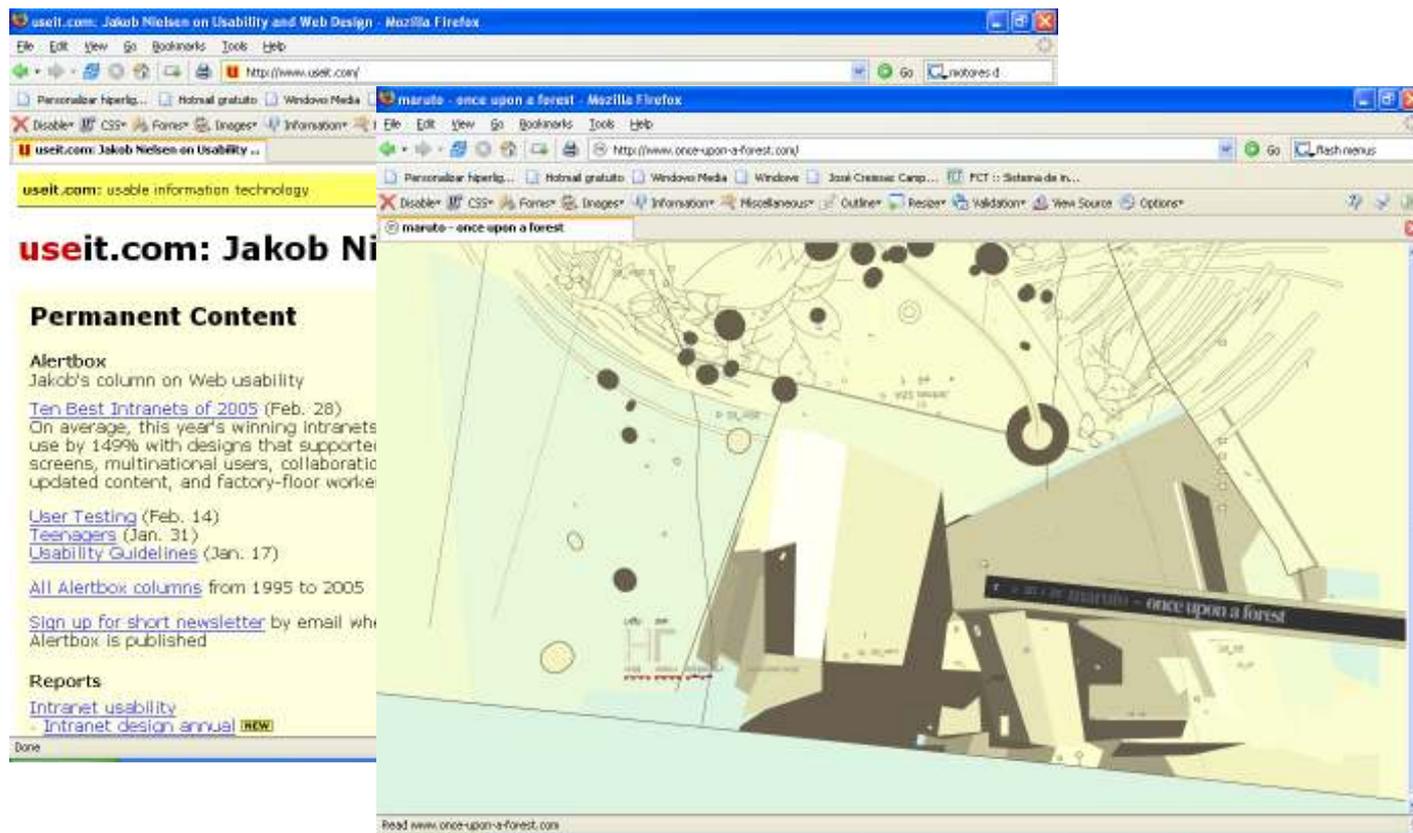


## Metáforas

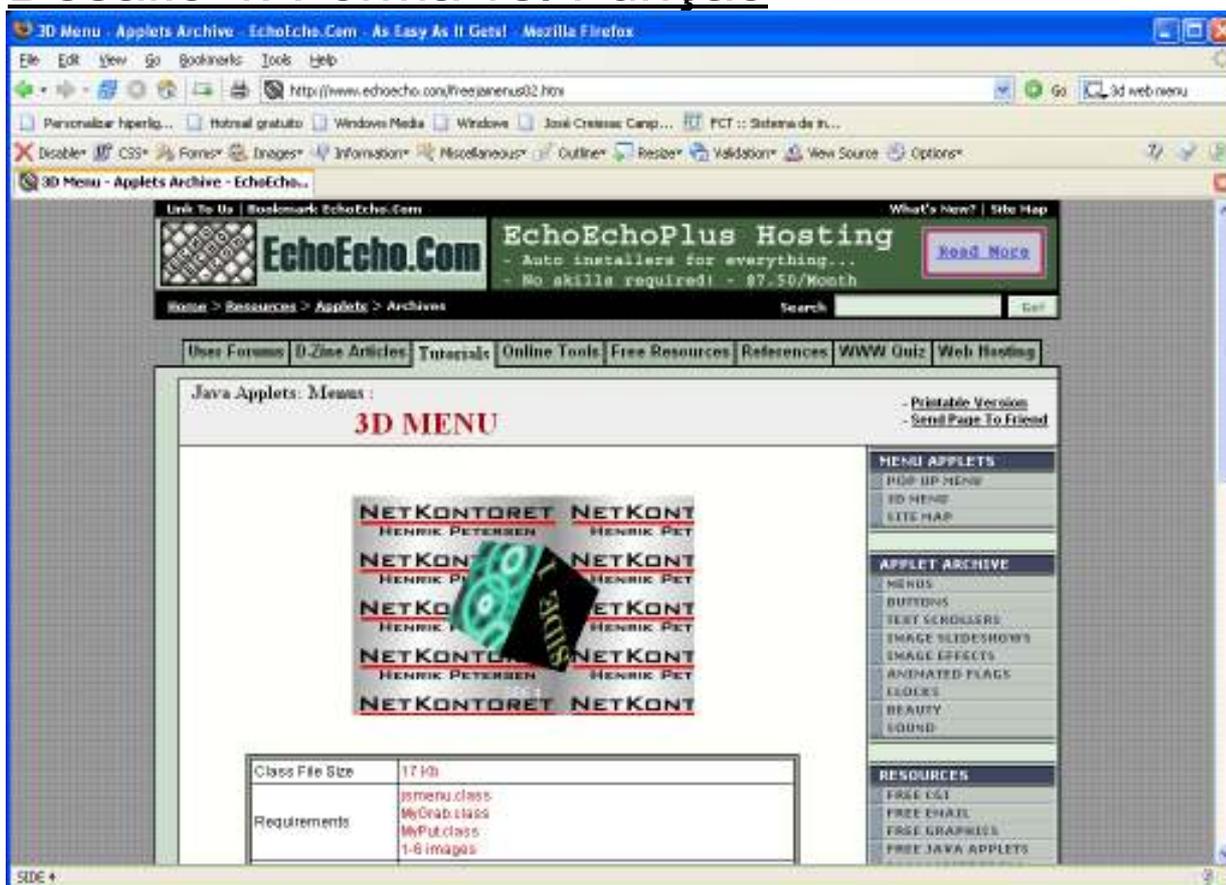
- Permitem explorar conhecimento pré-existente para diminuir os Golfos da Execução e da Avaliação.  
(exemplo: processador de texto vs. máquina de escrever)
- Podem surgir problemas:
  - Quando partes da interface divergem da metáfora.  
(exemplo: ejectar um disco num Mac!)
  - Quando partes da interface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



## Desafio 4: Forma vs. Função



## Desafio 4: Forma vs. Função





## Desafio 4: Forma vs. Função



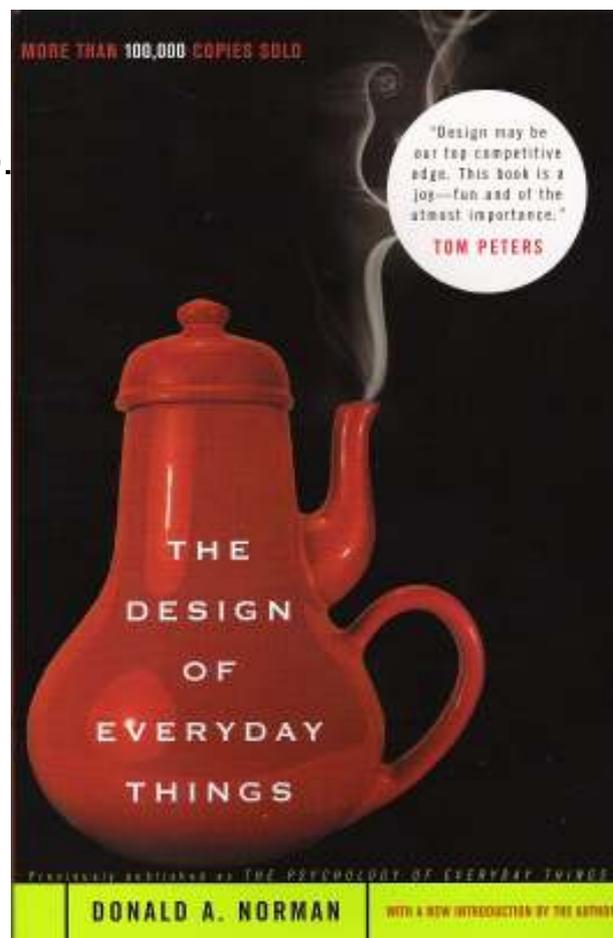
Mestrado em Sistemas Móveis – Universidade do Minho

27



## Soluções

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



Mestrado em Sistemas Móveis – Universidade do Minho

28



## Em conclusão...

- O que caracteriza um bom sistema interactivo? (heurísticas de Nielsen)
  - utiliza diálogo simples e natural;
  - fala a linguagem dos utilizadores;
  - minimiza a carga de memória;
  - é consistente;
  - fornece *feedback*.
  - fornece saídas claramente assinaladas.
  - fornece atalhos (*short-cuts*).
  - fornece boas mensagens de erro.
  - previne erros.
- e, já agora, que seja esteticamente agradável!



## Usability isn't skin deep



## 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 Centred 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 (a usabilidade).



## Processo de Engenharia de Usabilidade

Actividade	Informação produzida	Técnicas
<b>Conhecer os utilizadores</b>	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 escolha?
  - Qual é o seu nível de perícia?
    - Inexperientes, intermédios e peritos.
  - Utilizadores intermitentes ou continuados?
- Todos este 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 recear 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 o modo como as 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
<b>Conhecer as tarefas</b>	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)*



## 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?

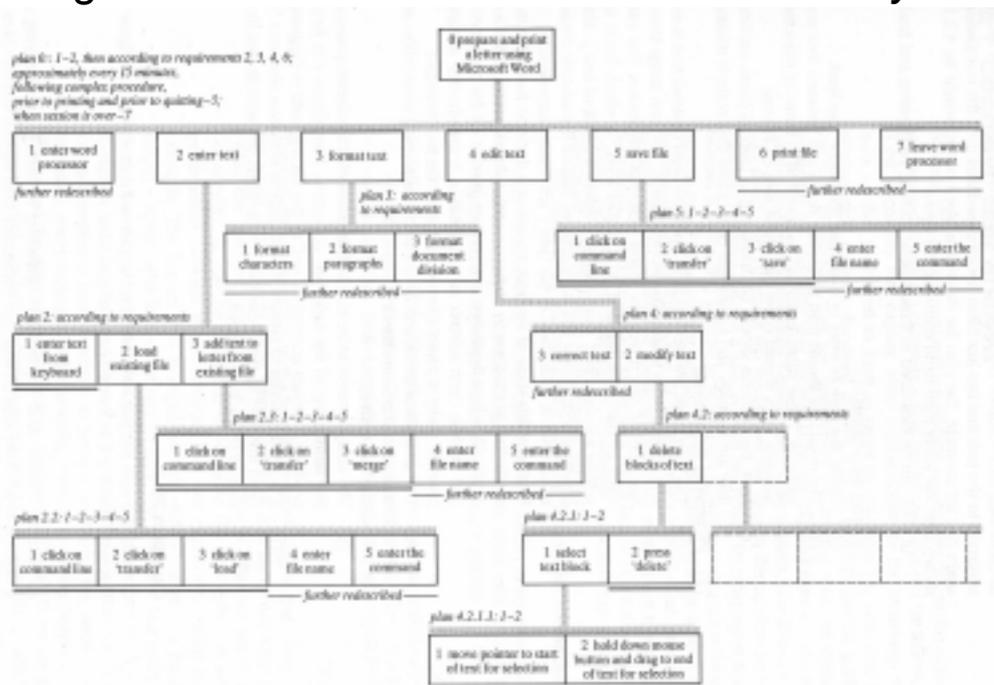
### **Composição**

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	Entrevistas, questionários e observação
<b>Capturar requisitos dos utilizadores</b>	Novo modelo de tarefas	Análise de tarefas
Definir objectivos de usabilidade	Requisitos dos utilizadores	Entrevistas, questionários e observação
Concepção do produto	Especificação de usabilidade	
Avaliação analítica	Concepção	<i>Guidelines</i>
Prototipagem	<i>Feedback</i> para a concepção	Heurísticas, <i>walkthroughs</i>
Avaliação com utilizadores	Protótipo para testes com os utilizadores	
Redesenho e avaliação com utilizadores	<i>Feedback</i> para redesenho	Entrevistas, questionários e observação
Avaliação final com utilizadores	Produto acabado	
	<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?
- Identificar **utilizadores**.
- Identificar **tarefas**.
- Definir **objectivos de usabilidade**.
- Outros requisitos Não Funcionais.



## 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?
- Identificar **utilizadores**.
- Identificar **tarefas**.
- 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
<b>Definir objectivos de usabilidade</b>	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

<b>Eficácia</b>	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
<b>Eficiência</b>	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
<b>Satisfação</b>	<i>Medição da satisfação utilizando questionários</i>



# Especificação de Usabilidade

<b>Especificação de Usabilidade</b>						
<i>Utilizadores: todos</i>						
<i>Pré-condições: medições a realizar após uma semana de utilização</i>						
		<b>Pior caso</b>	<b>Nível mínimo aceitável</b>	<b>Nível planeado</b>	<b>Melhor nível</b>	<b>Nível actual</b>
<b>Atributo</b>	<b>Medição</b>					
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

<b>Actividade</b>	<b>Informação produzida</b>	<b>Técnicas</b>
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	
<b>Concepção do produto</b>	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 Producto (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 Producto (II)

- **Modelação da Interacção**
  - 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 interacção?
  - *Interacção implícita, integração física-digital.*



## Concepção do Producto (III)

- **Guidelines**

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



## Concepção do Producto (IV)

- **Algumas Guidelines (www.wmmus.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 tratamento de erros
- Forneça *undo* e pontos de saída claramente assinalados
- 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. Mathew, 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 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>
<b>Avaliação analítica</b>	<i>Feedback</i> para a concepção	Heurísticas, <i>walkthroughs</i>
Prototipagem	Protótipo para testes com os utilizadores	
<b>Avaliação com utilizadores</b>	<i>Feedback</i> para redesenho	Entrevistas, questionários e observação
<b>Redesenho e avaliação com utilizadores</b>	Produto acabado	
<b>Avaliação final com utilizadores</b>	<i>Feedback</i> para o futuro	Entrevistas, questionários e observação

Adaptado de (Faulkner, 2000)



## 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 sumativa;
  - ou 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?

(O estranho caso da dactilógrafa!)



## 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*
    - ...
  - Métodos menos formais
    - Avaliação Heurística  
Método de inspecção *manual* baseado em *guidelines*.
    - *Cognitive Walkthroughs*  
Método de inspecção *manual* baseado num modelo de aprendizagem exploratória.
  - Qualidade dos resultados depende da *qualidade* de quem está a aplicar o método.



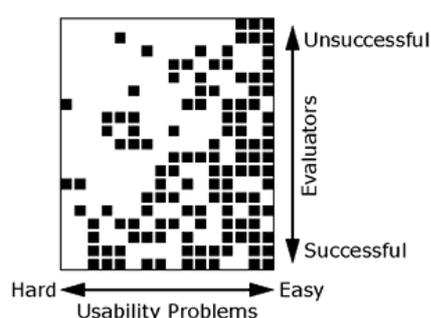
## Avaliação Heurística

- 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.
- 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).



## Avaliação Heurística (cont.)

- Procedimento:
  - Para cada avaliador de forma isolada (e repetidamente):
    - Inspeccionar o fluxo da interface de écran para écran;
    - Inspeccionar cada écran face as heurísticas.
  - Agregar os resultados.
- Necessidade de múltiplos avaliadores:



(Nielsen 92)



## 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.

Nova versão em [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)



## Faça você mesmo (I)

- Analisar a edição de sumários do FLY
- <http://www.di.uminho.pt/~jfc/ensino/ihcau/ivy>



## Heurísticas para Ecrãs Públicos

- Informação útil e relevante
- Utilização de visão periféria
- Adequação ao ambiente
- Desenho baseado em “informação suficiente”
- Consistência e intuitividade
- Transição simples para informação mais detalhada
- Visibilidade do estado
- Desenho esteticamente agradável

(Mankoff et al.)



## Outras Heurísticas(?) para Ecrãs Públicos

- A cor pode ser utilizada para facilitar compreensão da informação
- O layout deve reflectir o tipo de informação
- Utilização judiciosa de animações
- Utilizar *banners* apenas se estritamente necessário
- Mostrar a presença de informação mas não os detalhes
- A utilização de ciclos pode ser útil, se feita com cuidado
- Evitar a utilização de áudio
- Eliminar ou esconder controlos de configuração

(Somervell et al.)



## Cognitive Walkthrough

- Um método para analisar a usabilidade de sistemas baseada na teoria CE+ da *aprendizagem exploratória* de Lewis Polson:
  1. O utilizador formula um objectivo a ser atingido no sistema;
  2. O utilizador procura na interface as acções disponíveis;
  3. O utilizador selecciona a acção que lhe parece mais adequada face ao seu objectivo;
  4. O utilizador realiza a acção e avalia a resposta do sistema para avaliar se está a fazer progresso.
- 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 Walkthrough (cont.)

- Pré-requisitos:
  - a) Uma descrição de quem são os **utilizadores**;
  - b) Descrições detalhadas das principais **tarefas** a serem realizadas no sistema;
  - c) Um **protótipo** do sistema que permita simular a execução das tarefas.



## Cognitive Walkthrough (cont.)

- Procedimento – cada perito coloca-se no lugar do utilizador e procura executar uma dada tarefa enquanto vai respondendo a um conjunto pré-definido de 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?)
- Respostas negativas podem ser classificadas por severidade – por exemplo:
  - Tipo 1. o problema pode causar confusão ou demora na execução da tarefa.
  - Tipo 2. o problema pode impedir que o utilizador consiga executar a tarefa sem ajuda.
  - Tipo 3. o problema impede a execução da tarefa.
- Análise deve ser efectuada por 5-7 avaliadores e os resultados são agregados.



## Faça você mesmo (II)

- <http://www.historiadodia.pt>
- Ler a história do dia
  1. Clicar a imagem correspondente à história
  2. Clicar ícone |> para iniciar leitura da história
  3. Clicar os ícones <| e |> para navegar na história até ao fim
  4. Clicar na casa para sair
- Ouvir uma história do arquivo
  1. Clicar ícone “Arquivo”
  2. Clicar o mês desejado
  3. Clicar o período desejado
  4. Clicar a imagem correspondente à história
  5. Clicar ícone microfone para iniciar a narração da história
  6. Clicar os ícones <| e |> para navegar na história até ao fim
  7. Clicar na casa para sair



## Onde procurar mais informação...

- [www.usabilitynet.org](http://www.usabilitynet.org) - projecto financiado pela UE.
- [www.usabilitynews.com](http://www.usabilitynews.com) - British HCI Group.
- [www.upassoc.org](http://www.upassoc.org) - Usability Professional's Association.
- [www.usabilityfirst.com](http://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.



## Conclusão

- Conhecer o utilizador
- Conhecer a tarefa.
- Planear para a usabilidade.
  
- Procurou-se apresentar um enquadramento genérico e alguns exemplos de técnicas.
  - [www.ibm.com/easy](http://www.ibm.com/easy) - User engineering
  - [www.foruse.com](http://www.foruse.com) - Usage centered design
  - ...