

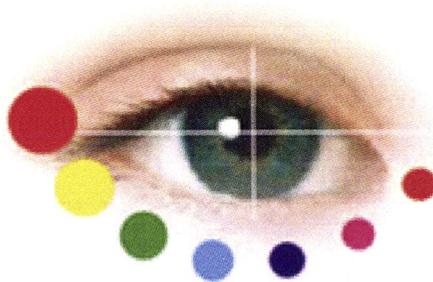
Universidade de Évora

Departamento de Gestão de Empresas

Mestrado em Organização e Sistemas de Informação

Dissertação de Mestrado

Interface Homem Máquina em Sistemas de Informação



Apresentada por

Lic^o Pedro Nuno Moreira da Silva

Orientador

Professor Doutor Carlos Zorrinho

Évora

2005

Esta dissertação não inclui as críticas e sugestões feitas pelo júri

Universidade de Évora
Departamento de Gestão de Empresas
Mestrado em Organização e Sistemas de Informação
Dissertação de Mestrado

ERRATA

**Interface Homem Máquina em Sistemas de
Informação**

Pedro Nuno Moreira Silva

Orientador:
Professor Doutor Carlos Zorrinho

Évora 2005

1. Escolha do tema

1.1. Problema

Quando surgiram os primeiros computadores, os utilizadores eram praticamente os seus construtores, já que manipulavam directamente o hardware não existindo propriamente uma interface que lhes facilitasse o trabalho (Winckler, 1999). Nos últimos anos a importância de produzir interfaces homem máquina tem assumido um papel vital. A interface tem vindo a ser reconhecida como sendo um dos aspectos mais relevantes para determinar a qualidade de um sistema.

A área de interface homem máquina tem como objectivo fornecer ferramentas e técnicas para produzir sistemas centrados nos utilizadores e nas suas necessidades. Dentro deste contexto, esta tese centra-se na produção ou modificação de sistemas explorando as ferramentas mais utilizadas de interface homem máquina, sugerindo um modelo de utilização para a produção de sistemas tendo como base as interfaces homem máquina.

1.2. Objectivos

A realização desta tese deverá ser entendida como um contributo para o desenvolvimento de eixos de pesquisa comuns no domínio da ergonomia cognitiva, em particular em torno da problemática da concepção e avaliação de interfaces homem-máquina.

O objectivo geral desta dissertação centra-se na definição de técnicas de verificação da Usabilidade em sistemas interactivos baseados em conhecimentos ergonómicos. Para que tal seja possível é necessário alcançar vários objectivos intermédios. Entre os principais, destacamos:

- Identificar / definir os tipos de interfaces com o utilizador para o qual o projecto deve ser direccionado.
- Identificar / definir as oportunidades e técnicas de verificação, que se prendem com as necessidades que os criadores / utilizadores têm durante o ciclo de vida do sistema interactivo.
- Identificar / definir abordagens pelas quais se possam verificar a interacção entre o homem e a máquina.

4. Ergonomia em Sistemas de Informação

Existem várias definições de ergonomia. No entanto, na sua essência, todas elas têm um factor em comum que se prende com o facto de ser um ramo da ciência e da tecnologia que inclui o conhecimento e as características comportamentais e biológicas do ser humano. Essas características, e tal como está descrita na definição oficial do Laboratório de Engenharia de *Usabilidade*(2003), devem ser aplicadas na especificação, design, avaliação, funcionamento e manutenção de produtos e sistemas, de forma a aumentar a segurança, eficiência e utilização satisfatória por indivíduos, grupos e organizações.

Enquanto conceito teórico, a Ergonomia (ou estudo dos factores humanos), é a descrita pela Associação Portuguesa de Ergonomia (2003) como sendo “a disciplina científica relacionada com a compreensão das interacções entre os seres humanos e os outros elementos de um sistema, é a profissão que aplica os princípios teóricos, dados e métodos pertinentes para conceber com vista a otimizar o bem-estar humano e o desempenho global do sistema.”

Dentro das suas competências incluem-se a Ergonomia Física, Cognitiva e Organizacional, sendo a ergonomia cognitiva a mais relevante para o nosso estudo, uma vez que está diz respeito aos processos mentais, como a percepção, a memória, o raciocínio e a resposta motora, componentes que afectam as interacções entre humanos e outros elementos de um sistema.

Os tópicos relevantes incluem a carga de trabalho mental, tomada de decisão, desempenho especializado, interacção homem-máquina, fiabilidade humana, stress do trabalho, e formação relacionadas com a concepção homem-máquina.

A analogia feita pelos investigadores da área de interface homem máquina, “entre mapas mentais dos indivíduos e a forma como estes se movimentam em ambientes físicos, concluindo que os utilizadores tendem a formar modelos mentais que se apresentam de modo mais simples do que a realidade.”(Marques Cardoso, 2003).

Desta forma, e segundo o Laboratório de Ergonomia e *Usabilidade* de Interfaces em Sistemas Homem – Tecnologia (2000), “a única e específica tecnologia da ergonomia é a tecnologia da interface homem - máquina. A ergonomia como ciência trata de desenvolver conhecimentos sobre as capacidades, limites e outras

insucesso das hipóteses do designer sobre a proposta de solução, tanto ao nível de funcionalidade como ao nível de interação.

Modelo de design em IHM

O modelo de design de interface foi dividido em três fases, tal como mostra a figura (8). Uma primeira fase que diz respeito ao conhecimento dos utilizadores e suas necessidades, uma segunda fase que trata do design do sistema, e uma terceira fase que é a implementação propriamente dita e o teste do produto.

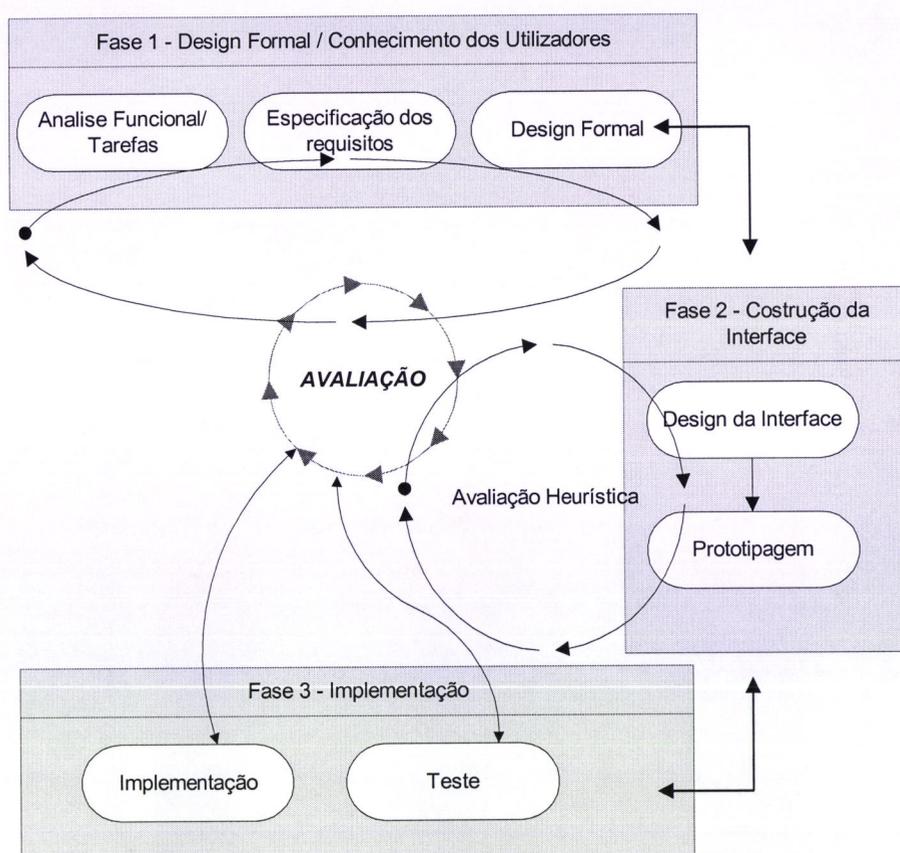


Figura 8 - Modelo de design IHM
(Adaptado de: Hartson, 1998)

Nesta nova abordagem, muito parecida com o modelo estrela atrás referido, a avaliação digamos que é a chave de todo o processo, já que ela está sempre presente e deve ser o motor de todo o desenvolvimento. A passagem de uma fase para a outra é sempre possível, mas é importante que tal não aconteça. O que é fundamental é que a participação dos utilizadores, na primeira e segunda fase seja muito forte, seja depois

- ***Implementar testes centrados nos utilizadores***

A única forma de confirmar que os objectivos foram atingidos e de identificar novas oportunidades para melhorar o desenho é recorrer aos testes centrados nos utilizadores. Estes testes têm as seguintes fases:

- Desenvolvimento de um plano de testes;
- Recolher e analisar os dados;
- Reportar os resultados e recomendações para alterações;
- Alterar até que o desenho cumpra os objectivos;
- Rastrear as alterações, manutenção;

Desta forma, o processo de desenvolvimento centrado no utilizador pode ser separado em duas fases importantíssimas. A primeira fase, que se designa de projecto, inclui o estudo detalhado do utilizador e das suas necessidades e da tecnologia que irá servir de suporte à construção da aplicação. A segunda, fase intitulada implementação, inclui a implementação propriamente dita, tendo em conta o estudo feito na primeira fase.

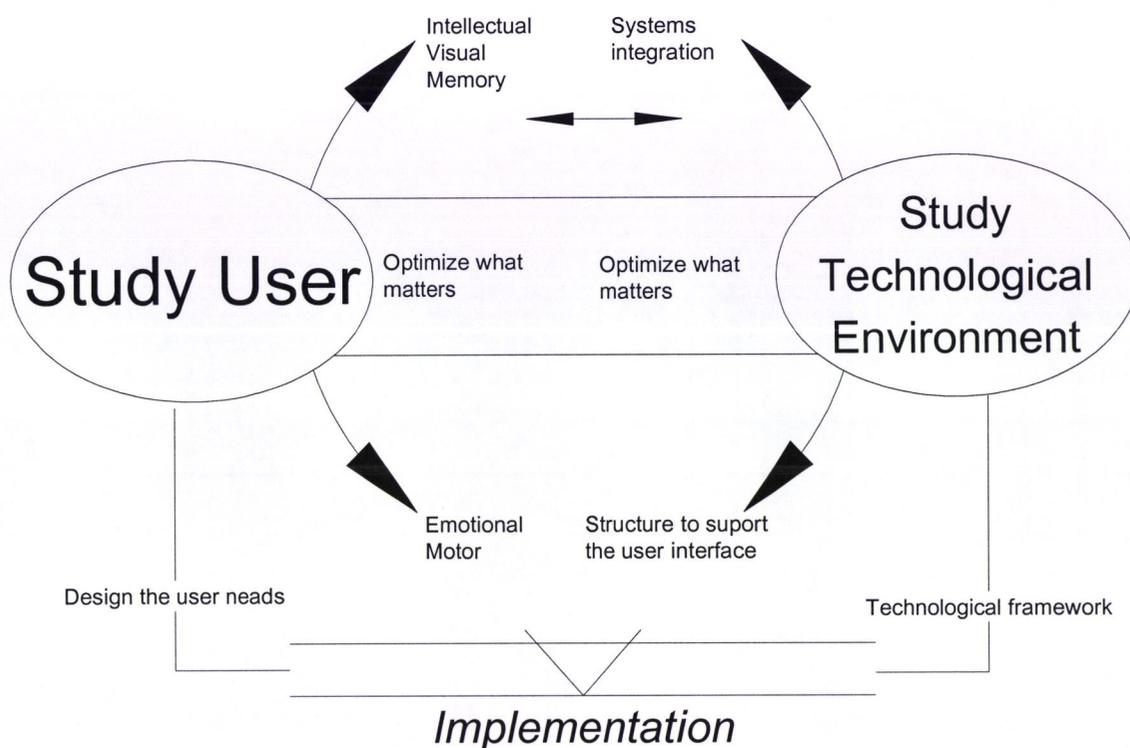


Figura 3 - Processo de DCU

(Adaptado de: Eric M. Schaffer, 1998, Human Factors International, Inc.)

Interface Homem Máquina em Sistemas de Informação

Dissertação apresentada à
Universidade Évora
para a obtenção do grau de
Mestre em Organização e Sistemas de Informação



162590

Orientador

Professor Doutor Carlos Zorrinho

Apresentada por

Lic^o Pedro Nuno Moreira da Silva

Évora

2005

Esta dissertação não inclui as críticas e sugestões feitas pelo júri

Agradecimentos

O trabalho agora apresentado apenas foi possível pelo empenho, dedicação e simpatia demonstrados por um conjunto de pessoas que, de forma altruísta, puderam e souberam acompanhar-me nesta laboriosa tarefa.

Não existindo o propósito de citar exaustivamente todos aqueles que contribuíram para a realização desta tese, agradeço em especial à minha família e ao Prof. Doutor Carlos Zorrinho, orientador da dissertação, pelos conhecimentos transmitidos e pela total disponibilizada demonstrada ao longo do trabalho. Aos meus colegas e amigos agradeço pelo apoio demonstrado.

A todos deixo, sem exceção, o meu profundo agradecimento.

Évora, 8 Abril de 2004

Resumo

A produção de software tem vindo ao longo dos anos a sofrer profundas alterações. A evolução é claramente sentida na crescente preocupação de produção de software que permita o controlo da qualidade dos programas e a satisfação dos clientes.

Um dos aspectos mais relevantes a levar em consideração é a qualidade da interacção entre o utilizador e os programas de computador. Para a grande maioria dos domínios de aplicação das tecnologias de informação, a comunicação entre o utilizador e o sistema tornou-se tão importante quanto o processamento efectuado pelo sistema.

Neste sentido, o desenho de sistemas de informação interactivos exige novos conhecimentos de ergonomia. De entre estas novas áreas de conhecimento salienta-se a ergonomia do software que, nos últimos anos, tem vindo a ser denominada de Usabilidade e que, por sua vez, deu origem ao conceito de Engenharia de Usabilidade. Os métodos e as técnicas propostos por esta engenharia são utilizados de forma a garantir uma Usabilidade óptima no desenho da interface com o utilizador a quando do desenvolvimento de um produto ou de um serviço, aumentando assim o valor desse mesmo produto para o cliente.

Neste contexto, o objecto desta investigação é o de encontrar eixos de pesquisa comuns no domínio da ergonomia cognitiva, em particular em torno da problemática da concepção e avaliação de interfaces homem-máquina.

Interface Human Machine in Information Systems

Abstract

The software production, has come to the long one of deep the years to suffer alterations. One from the clear signals of this type of alterations, is felt in the increasing concern of software production that allows the control of the quality of the programs and the satisfaction of the customers.

One of the aspects most important to have in account, is the quality of the interaction between the user and the machine.

For the great majority of dominions of application of the information technologies, the communication between the user and the system became so important how much the processing made for the system. In this sense, the drawing of interactive systems, demands new knowledge of ergonomics.

Between these new areas of salient knowledge it ergonomics of the software, that in the last years we hear to call usability, that in turn, gave to origin to the concept of Usability Engineering.

The methods and the techniques considered for this engineering are used of from, to guarantee an excellent usability in the drawing of the interface with the user. In this context the object of this work, is to find, common axles of research in the dominion of the conception and the evaluation of human-machine interfaces.

Índice

Agradecimentos	III
Resumo	IV
Abstract.....	V
1. Escolha do tema	11
1.1. Problema	11
1.2. Objectivos	11
1.3. Metodologia e organização da tese.....	12
1.4. Resultados esperados.....	13
2. Introdução.....	14
2.1. Objecto da Investigação	14
3. Sistemas de Informação interactivos	16
3.1. O significado do termo Interacção.....	17
4. Ergonomia em Sistemas de Informação	18
4.1. Interfaces	19
4.2. Porquê a mudança de paradigma?	22
5. Soluções Centradas no Utilizador (SCU).....	28
5.1. Conhecimento dos Utilizadores.....	29
5.2. Processo de Desenvolvimento Centrado no Utilizador.	31
6. Avaliação Heurística	36
6.1. Heurísticas	36
6.2. Testes com os utilizadores.....	41
7. Testes de Usabilidade	46
7.1. Os atributos da Usabilidade.....	47
8. Análise	51
8.1. Análise do utilizador	51
8.2. Análise do trabalho.....	52
8.3. Tarefa e Actividade	53
8.4. Análise do Ambiente	59
8.5. Relatório de Análise	59
Síntese.....	62
8.6. Síntese utilizando UML.....	64
8.7. Projecto da Interface.....	66
8.8. Resumo do Processo de Síntese.....	76
9. Avaliação.....	79
9.1. Classificação das Avaliações.....	80
9.2. Técnicas de Avaliação.....	80
9.3. Projecto de Avaliação.....	85
10. Modelo.....	88
10.1. Modelo de Avaliação Triangular de Interfaces (MATI).....	89
11. Conclusão	99

11.1. Perspectivas de investigação futura	100
12. Referências Bibliográficas	101
13. Sites de referência.....	105
14. Anexos	106

Índice de Figuras

Figura 1 – Interface com o utilizador	20
Figura 2 – Evolução histórica da produção de sistemas computacionais, adaptado de “User – Centred Solutions ”, Eric M. Schaffer, Human Factors International, pag 7.....	29
Figura 3 - Processo de DCU.....	33
Figura 4 – Curva demonstrativa da proporção dos problemas de Usabilidade encontradas na interface utilizando a Avaliação Heurística com vários avaliadores. Adaptado de Nielsen “Heristic Evaluation”, 4.....	44
Figura 5- Processo de negócio. Adaptado de Alfredo, 2003.....	57
Figura 6 - Etapa de reengenharia do trabalho do processo de Síntese.....	64
Figura 7 - <i>Star Model – Modelo Estrela</i> Adaptado de Hix, D. and Hartson, H.R.1993.....	74
Figura 8 - Modelo de design IHM.....	75
Figura 9 – Resumo do processo de Síntese	77
Figura 10 - Fases do ensaio de interação, adaptado de Cybis, 2003, p.111	83
Figura 11 – Plano de testes, adaptado de Cybis, 2003, p. 129.....	86
Figura 12 – Três realidades do processo de desenvolvimento.	88
Figura 13 – Primeira fase do MATI	90
Figura 14- Segunda fase do MATI.....	91
Figura 15 - Modelo de Avaliação Triangular de Interface	92
Figura 16 - Modelo em cascata, adaptado de António Miguel, 2003, pag 91	94
Figura 17 – Aplicação do MATI no modelo de desenvolvimento em Cascata	95

Índice de tabelas

Tabela 1- Escala de classificação da Avaliação Heurística, adaptado de Miguel Santinho, 2001	39
Tabela 2 – Características da Heurística.....	41
Tabela 3 – Tarefas como planos. Adaptado de Alfredo, 2003.	56
Tabela 4 - Exemplo segundo a abordagem HTA	58
Tabela 5 - As diferentes cores e seus significados, adaptado de Bruno Figueiredo, 2004	69
Tabela 6 - Significado das linhas e formas, adaptado de Barros(2003), p.58.....	71

Lista de abreviaturas

AT – Análise do trabalho

DCU – Desenvolvimento centrado nos utilizadores

ES – Ergonomia de software

GUI – Graphical User Interface

HCI – Human Computer Interaction

HTA – Hierarchical Task Analysis

IHM – Interfaces Homem Máquina

ISO – International Organization for Standardization

MATI – Modelo Análise Triangular de Interface

MDA – Method Analytique de Description des tâches

OSTA – Open Systems Task Analysis

OUI – Object User Interface

PARS – Xerox Palo Alto Reserch

SCU – Soluções centradas nos utilizadores

SSM – Soft System Methodology

UML – Unified Modelling Language

WUI – Web User Interface

WYSIWYG – What You See Is What You Get

1. Escolha do tema

1.1. Problema

Quando surgiram os primeiros computadores, os utilizadores eram praticamente os seus construtores, já que manipulavam directamente o hardware não existindo propriamente uma interface que lhes facilitasse o trabalho (Winckler, 1999). Nos últimos anos a importância de produzir interfaces homem máquina tem assumido um papel vital. A interface tem vindo a ser reconhecida como sendo um dos aspectos mais relevantes para determinar a qualidade de um sistema.

A área de interface homem máquina tem como objectivo fornecer ferramentas e técnicas para produzir sistemas centrados nos utilizadores e nas suas necessidades. Dentro deste contexto, esta tese centra-se na produção ou modificação de sistemas explorando as ferramentas mais utilizadas de interface homem máquina, sugerindo um modelo de utilização para a produção de sistemas tendo como base as interfaces homem máquina.

1.2. Objectivos

A realização desta tese deverá ser entendida como um contributo para o desenvolvimento de eixos de pesquisa comuns no domínio da ergonomia cognitiva, em particular em torno da problemática da concepção e avaliação de interfaces homem-máquina.

O objectivo geral desta dissertação centra-se na definição de técnicas de verificação da Usabilidade em sistemas iterativos baseados em conhecimentos ergonómicos. Para que tal seja possível é necessário alcançar vários objectivos intermédios. Entre os principais, destacamos:

- Identificar / definir os tipos de interfaces com o utilizador para o qual o projecto deve ser direccionado.
- Identificar / definir as oportunidades e técnicas de verificação, que se prendem com as necessidades que os criadores / utilizadores têm durante o ciclo de vida do sistema interactivo.
- Identificar / definir abordagens pelas quais se possam verificar a interacção entre o homem e a máquina.

1.3. Metodologia e organização da tese

Um dos factores que diferencia uma investigação científica de outros trabalhos não científicos é o suporte de uma metodologia e de um conjunto de métodos e procedimentos que asseguram a validade científica dos resultados obtidos. A especificação de uma metodologia científica, para além de assegurar que a investigação é passível de ser reproduzida, fornece ferramentas ao investigador para abordar uma dada problemática de uma forma regrada e dirigida.

A metodologia a usar para a realização deste estudo é a *Action Research* (Investigação/Acção) que procura responder às preocupações das pessoas numa problemática imediata (a acção) e, por outro lado, aprender com o processo e aumentar a base de conhecimento da comunidade científica (a investigação). Neste sentido, o presente trabalho está estruturado pelos capítulos seguintes:

Parte 1 – Sistemas de Informação interactivos e ergonómicos – realiza uma introdução aos sistemas interactivos e ergonómicos fazendo uma abordagem ao significado de interacção e da necessidade de ergonomia em sistemas de informação.

Parte 2 – Conhecimento dos utilizadores – refere-se à necessidade de conhecer os utilizadores quando se constroem sistemas de informação, nomeadamente soluções centradas nos utilizadores.

Parte 3 – Avaliação Heurística e Testes com os Utilizadores – apresentam-se formas de realização da Avaliação Heurística e de Testes com Utilizadores, procurando o seu significado e necessidade.

Parte 4 – Testes de Usabilidade – refere-se aos diferentes tipos de ferramentas para avaliar a Usabilidade de interfaces homem máquina.

Parte 5 – Processo de Análise – sugere uma primeira arquitectura para a construção de um sistema que passa por um processo de Análise.

Parte 6 – Processo de Síntese – Sugere uma segunda arquitectura para a construção de um sistema interactivo que passa por um processo de Síntese.

Parte 7 – Processo de Avaliação – Descreve as etapas realizadas no processo de avaliação e que tipo de testes devem ser incluídos nos processos anteriores.

Parte 8 – Proposta de Modelo de construção de IHM em SI – é proposto um modelo de construção de Sistemas de Informação tendo como base Interfaces Homem Máquina.

Parte 9 – Conclusão – São apresentadas conclusões da dissertação. São descritas considerações finais sobre o trabalho e sugeridas as propostas para estudos futuros.

A dissertação contém ainda um conjunto de anexos no final da mesma.

1.4. Resultados esperados

A presente dissertação pretende contribuir para melhorar a formação nestas matérias, tanto no quadro das licenciaturas na área de Informática como de formação avançada em Usabilidade em Sistemas de Informação, disponibilizando informação organizada numa área de conhecimentos recentemente criada.

Adicionalmente, o presente plano de estudos tem como objectivo fundamental o enquadramento para uma futura tese de doutoramento em Interfaces Homem Máquina em Sistemas de Informação.

No final do projecto, através das técnicas exploradas para o desenvolvimento de software interactivo, pretende-se criar documentação para que os produtores de software tenham acesso a estudos e formas de criar ou modificar sistemas interactivos e usáveis pelos utilizadores

2. Introdução

A produção de software tem vindo ao longo dos anos a sofrer profundas alterações. Um dos sinais claros deste tipo de alterações é sentida na crescente preocupação de produção de software que permite o controlo da qualidade dos programas e a satisfação dos clientes. Um dos aspectos mais relevantes a ter em conta é a qualidade da interacção entre o utilizador e os programas de computador.

Para a grande maioria dos domínios de aplicação das tecnologias de informação, a comunicação entre o utilizador e o sistema tornou-se tão importante quanto o processamento feito pelo sistema.

2.1. Objecto da Investigação

O desenho de sistemas de informação interactivos exige novos conhecimentos de ergonomia. De entre estas novas áreas de conhecimento salienta-se a ergonomia do software, que nos últimos anos ouvimos chamar de *Usabilidade* e que, por sua vez, deu origem ao conceito de engenharia de *Usabilidade*.

Os métodos e as técnicas propostos por esta engenharia são utilizados de forma a garantir uma *Usabilidade* óptima no desenho da interface com o utilizador a quando do desenvolvimento de um produto ou de um serviço, aumentando assim o valor desse mesmo produto para o cliente.

Neste contexto, o objecto desta investigação é o de encontrar eixos de pesquisa comuns no domínio da ergonomia cognitiva, em particular em torno da problemática da concepção e avaliação de interfaces homem-máquina. Para o efeito, será efectuado um levantamento do “estado da arte” neste domínio, com especial relevo para os factores que influenciam a construção ou manutenção de sistemas, utilizando interfaces homem máquina.

Por outro lado, pretende-se identificar os aspectos mais importantes que rodeiam a criação de sistemas de informação interactivos, e propor um modelo que sustente todos estes aspectos.

Parte 1

**Sistemas de Informação Interactivos e
Ergonómicos**

3. Sistemas de Informação interactivos

Os avanços tecnológicos colocaram à nossa disposição o computador. Este passou a fazer parte das nossas vidas e a funcionar como instrumento de trabalho, brinquedo, e mesmo máquina com a qual nos é permitido estabelecer as mais variadas comunicações.

Esta tese estuda a utilização do computador pessoal e as relações que com este podem ser estabelecidas para que os objectivos possam ser conseguidos ou determinadas tarefas executadas.

O computador é sem dúvida a maior invenção ocorrida na nossa era. Está longe de poder ser considerado semelhante a uma máquina de escrever, a um simples telefone ou a um ficheiro manual de registo de dados. Actualmente, e primeiro que tudo, o computador exhibe as suas características de uma forma cada vez mais transparente para o utilizador. (Laboratório E. *Usabilidade*, 2003).

Importa também perceber que o computador possui capacidades de transformação das diversas ferramentas de acordo com as necessidades dos utilizadores. Como refere Graça Simões: “Aviões há-os de todas as formas e podem ser descritos das mais diversas maneiras, mas não há qualquer tipo de dificuldade em dizer o que fazem: voam. Não existe nenhuma forma igualmente elegante de definir o computador pela sua função. Pode dizer-se que o computador ‘ computa ’ e a ciência da computação pode estabelecer um quadro conceptual de referência para definir o ‘ computável ’. Mas, ainda assim o que foi isolado como sendo ‘ o computador essencial ’ não apresenta analogias fáceis com outros aspectos do mundo (tal como um avião ou um pássaro), excepto, é claro, as analogias com a pessoa” (Simões, 1995:2).

Para finalizar, o computador possui ainda capacidades adoptivas e capacidades capazes de alterar o comportamento, adaptando-se a cada utilizador e aos seus objectivos.

De salientar que os utilizadores têm um papel fundamental no manuseamento da máquina. É o próprio utilizado que altera o comportamento da máquina, é ele que a controla e que lhe indica um conjunto de ordens, permitindo que o computador processe essas mesmas ordens e devolva os resultados pretendidos. Este processo é feito cada vez mais interactivamente e de forma progressivamente mais simples e

eficiente. Este processo é realizado, como não poderia deixar de ser, recorrendo à interface - utilizador.

3.1. O significado do termo Interação

O termo interação, e no que diz respeito à IHM, tem como principal objectivo o estudo quer do desenho do computador, quer a forma como este é utilizado. A investigação nesta área tem como função principal tornar a utilização dos computadores o mais próximo possível do utilizador comum. Para que tal seja conseguido é necessário que estes sejam fáceis de utilizar e sejam “transparentes” e agradáveis na utilização.

O domínio de investigação em Interface Homem Máquina (Human Computer Interaction HCI, na literatura anglo-saxónica), e a que chamaremos ao longo desta tese de IHM, tem como objectivo o estudo entre a interação entre duas entidades, uma humana e outra a máquina, e visa encontrar eixos de pesquisa comuns no domínio da ergonomia cognitiva em torno da problemática da concepção e avaliação de Interfaces Homem – Máquina.

Os estudos no domínio da IHM visam, desta forma, construir e desenvolver teorias, modelos e princípios, que permitam, por um lado, o desenho de sistemas de informação interactivos, no que diz respeito à ligação destes com o utilizador, e, por outro, permitir perceber a utilização que é feita destes sistemas e respectivos reflexos sobre a actividade humana.

De entre estas novas áreas de investigação salienta-se a Ergonomia do Software e a Engenharia de *Usabilidade*. Os métodos e as técnicas propostos são utilizados de forma a garantir uma ergonomia óptima no desenho da interface com o utilizador no momento da projecção e desenvolvimento do sistema de informação.

4. Ergonomia em Sistemas de Informação

Existem várias definições de ergonomia. No entanto, na sua essência, todas elas têm um factor em comum que se prende com o facto de ser um ramo da ciência e da tecnologia que inclui o conhecimento e as características comportamentais e biológicas do ser humano. Essas características, e tal como está descrita na definição oficial do Laboratório de Engenharia de *Usabilidade*(2003), devem ser aplicadas na especificação, design, avaliação, funcionamento e manutenção de produtos e sistemas, de forma a aumentar a segurança, eficiência e utilização satisfatória por indivíduos, grupos e organizações.

Enquanto conceito teórico, a Ergonomia (ou estudo dos factores humanos), é a descrita pela Associação Portuguesa de Ergonomia (2003) como sendo “a disciplina científica relacionada com a compreensão das interacções entre os seres humanos e os outros elementos de um sistema, é a profissão que aplica os princípios teóricos, dados e métodos pertinentes para conceber com vista a otimizar o bem-estar humano e o desempenho global do sistema.”

Dentro das suas competências incluem-se a Ergonomia Física, Cognitiva e Organizacional, sendo a ergonomia cognitiva a mais relevante para o nosso estudo, uma vez que está diz respeito aos processos mentais, como a percepção, a memória, o raciocínio e a resposta motora, componentes que afectam as interacções entre humanos e outros elementos de um sistema.

Os tópicos relevantes incluem a carga de trabalho mental, tomada de decisão, desempenho especializado, interacção homem-máquina, fiabilidade humana, stress do trabalho, e formação relacionadas com a concepção homem-máquina.

A analogia feita pelos investigadores da área de interface homem máquina, “entre mapas mentais dos indevidos e a forma como estes se movimentam em ambientes físicos, concluindo que os utilizadores tendem a formar modelos mentais que se apresentam de modo mais simples do que a realidade.”(Marques Cardoso, 2003).

Desta forma, e segundo o Laboratório de Ergonomia e *Usabilidade* de Interfaces em Sistemas Homem – Tecnologia (2000), "a única e específica tecnologia da ergonomia é a tecnologia da interface homem - máquina. A ergonomia como ciência trata de desenvolver conhecimentos sobre as capacidades, limites e outras

características do desempenho humano e que se relacionam com o projecto de interfaces, entre indivíduos e outros componentes do sistema. Como prática, a ergonomia compreende a aplicação da tecnologia da interface homem - máquina ao projecto ou modificações de sistemas para aumentar a segurança, conforto e eficiência do sistema e da qualidade de vida.”

A ergonomia de um sistema de informação ajuda a diminuir as distâncias que existem entre os utilizadores de um dado sistema informatizado e a máquina. Essa ajuda é sentida mais vivamente na facilidade de compreensão e de desempenho na utilização dos sistemas informáticos, servindo de garante para possíveis frustrações que podem ocorrer durante a utilização do software.

4.1. Interfaces

“A interface de uma aplicação corresponde entre trinta a cinquenta por cento do código de uma aplicação”

Myers & Rosson (1992)

A interface é uma parte integrante de qualquer sistema computacional e determina a forma como as pessoas interagem e controlam o sistema. Cada vez mais a vida humana está ligada à utilização de sistemas computacionais e electrónicos. Esta é uma realidade que ocorre tanto a nível organizacional como a nível pessoal. Nos nossos dias, o desenvolvimento e o projecto de interfaces são, sem sombra de dúvidas, a parte mais dispendiosa da maioria das aplicações informáticas. Tal como refere Myers & Rosson (1992): “A interface de uma aplicação corresponde entre trinta a cinquenta por cento do código de uma aplicação.

A facilidade de utilização destes dispositivos deve ser encarada como sendo uma parte essencial do seu projecto, uma vez que, pelo simples facto de um sistema possuir uma interface difícil de compreender, este não irá ser utilizado na sua perfeição, levando a um elevado desperdício de tempo e a que se gaste muito dinheiro em formação.

Tal como está descrito no relatório de investigação POCTI/Com43208 (2001), interface corresponde às ligações entre o sistema e o utilizador que permitem

estabelecer uma comunicação eficaz e que têm como objectivo promover a participação e a interacção, resultando na satisfação do utilizador. A interface deverá ser desenvolvida tendo como base as características cognitivas do indivíduo: memória, percepção e representações.

Existem variadíssimas definições de interface. No entanto, de uma forma mais simplista, podemos definir interface como sendo a forma como o utilizador interage com a máquina (ver fig.1).

Imaginemos, por exemplo a Internet. A navegação dentro de um site é um dos elementos da interface, outro elemento é a própria página, já que é através dela que comunicamos com a máquina.

Contudo, a interface não trata exclusivamente da interacção entre o homem e a máquina. Por exemplo no caso de um site, a interface é a forma como este comunica com o utilizador, quer a comunicação seja estática ou dinâmica.

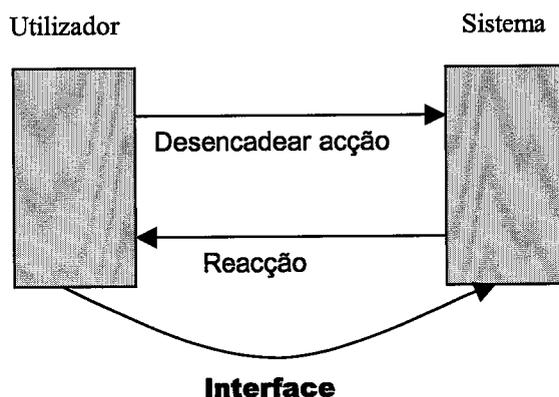


Figura 1 – Interface com o utilizador

Segundo Nilsen (1994) a maioria das interfaces existentes apresentam problemas de construção, nomeadamente no que diz respeito à arquitectura de informação, organização de elementos visuais, interactividade e funcionalidade. Estes problemas originam problemas na aprendizagem e na utilização correcta da aplicação, impossibilitando que se retire o máximo partido das potencialidades que esta pode oferecer.

A construção de interfaces tem vindo a ser fortemente influenciada pelo aumento da complexidade dos sistemas e pela crescente preocupação na utilização de normas que melhorem a qualidade do software produzido. (ISO/IEC 9126-1, 2000).

No início, a criação de interfaces era feita utilizando caracteres. Actualmente, a maioria das aplicações centram-se nas interfaces gráficas (GUI- Graphical User Interface). A primeira interface gráfico GUI foi desenvolvida pela Xerox Palo Alto Reserch (PARS) para o sistema 8010 Star. Esta interface era baseado em janelas e o conteúdo era manipulado através de ícones, pondo de parte o texto, apresentando também o mecanismo de clicar e arrastar, utilizando o rato, o que permitia a manipulação de objectos da tela e formatação do texto conhecida como WYSIWYG (What You See Is What You Get). Outras interfaces surgiram desde então, como é o caso das interfaces centradas nos objectos (OUI- Object User Interface) e, paralelamente com a explosão do fenómeno Internet, surgiram os WUI (Web User Interface).

4.1.1. Importância da Interface com o utilizador

A necessidade cada vez mais crescente de produzir melhores interfaces não se limita apenas ao nível do utilizador. No mundo empresarial esta necessidade é crescente. No passado, e numa perspectiva industrial, as empresas centravam a sua principal atenção no custo do hardware. Com o decréscimo dos custos associados ao hardware, a preocupação passou a centrar-se no custo de software, nomeadamente no que diz respeito ao desenvolvimento de novo software.

Nos nossos dias, a realidade já é outra. As empresas estão a centrar a sua atenção no custo associado ao personware, que se refere aos custos associados à formação e utilização diária dos sistemas por parte dos seus funcionários. Todavia, a preocupação em investir cada vez mais nas interfaces não se limita ao desenvolvimento de novos sistemas computacionais.

Nos últimos anos, em Portugal, a importância das aplicações integradas tem sido gigantesca. Este tipo de aplicações, para que se consiga tirar um maior nível de

rendimento, tem de ser parametrizadas por especialistas quer ao nível funcional, quer ao nível das interfaces, de forma a satisfazerem as necessidades dos utilizadores.

O custo do personware, ou nas soluções centradas no utilizador (SCU), é actualmente uma das principais preocupações das empresas no que diz respeito ao custo de sistemas interactivos, uma vez que este pode possibilitar um manancial de ferramentas que incluem tanto os custos de formação dos utilizadores, como os custos provenientes da utilização diária do software (incluindo as perdas na produtividade causadas por dificuldades de utilização).

4.2. Porquê a mudança de paradigma?

A mudança de paradigma faz-se sentir porque os programas existentes dedicam grande parte das linhas de código à aparência que a aplicação irá ter, esquecendo recorrentemente um factor não menos importante que é a funcionalidade. Então pode surgir a seguinte questão: *Quais serão os constituintes de uma boa interface?* Na resposta a esta questão não bastará apenas referir que uma boa interface é aquela que é “amigável”. Na verdade, a grande maioria das pessoas (utilizadores de uma forma geral) não pretende encontrar um amigo no seu computador, pretende sim realizar o seu trabalho da forma mais simples e agradável possível.

4.2.1. Dificuldades encontradas no projecto de interfaces.

Os benefícios causados pela utilização das normas e critérios da ES são indiscutíveis. No entanto, os problemas que levam ao projecto de uma interface para um sistema de informação ainda não estão resolvidos. As principais dificuldades que se encontram quando se está a projectar um sistemas são as seguintes (Brad A. Myers,1993):

- *Dificuldade em entender as tarefas e os utilizadores.* A necessidade de adequar a interface ao modo com que será utilizado requer uma compreensão prévia dos utilizadores e das suas habilidades e expectativas. O levantamento deste tipo de

informação não é fácil, sobretudo quando os programadores têm dificuldades de imaginar as necessidades e condições de trabalho dos utilizadores.

- *Complexidade inerente as tarefas e aplicações.* As aplicações a serem criadas envolvem geralmente situações de difícil modelação, ou porque as tarefas necessárias são na sua essência muito complicadas, ou porque a aplicação que se propõem a desenvolver têm que resolver uma gama muito extensa de problemas.

- *A variedade de aspectos e requisitos diferentes.* Além das limitações inerentes a qualquer projecto, a construção de interfaces tem que ser feita obedecendo a determinados padrões, design gráfico, documentação, internacionalização e performance, entre outras. Estas condições fazem com que se torne ainda mais complicado o desenvolvimento da interface.

- *Teorias e métodos insuficientes.* Embora existam muitas metodologias para a criação de uma boa interface, o que torna impossível a criação de um método único e abrangente, a maior parte dos estudos disponíveis revela que a habilidade e treino dos projectistas é um factor importantíssimo para a sua qualidade.

- *Dificuldade em realizar um projecto interactivo.* O ideal será que a nossa interface possa ser refinada ciclicamente. Todavia, a maioria das alterações acarretam uma diminuição da *Usabilidade*¹ do nosso sistema. Neste sentido, é muito difícil saber quando uma interface está bem elaborada. Por outro lado, é também muito difícil obter resultados partindo dos utilizadores primários, até porque na maioria dos casos estes não iram ser os compradores nem os responsáveis.

4.2.2. Dificuldade em implementar interfaces.

Como podemos constatar a projecção de uma interface não é tarefa fácil. Mas a sua implementação também o não será. Tal como diz Brad A. Meyers (1993), existem razões importantes que explicam porque é que o software relacionado com a implementação de interfaces está entre os mais difíceis de ser implementado.

Essas razões prendem-se com o seguinte facto:

Necessidade de se fazer um projecto interactivo – Implementar uma interface interactivamente significa que as etapas do projecto estão perfeitamente interligadas com a implementação e teste. A vantagem directa deste tipo de projectos reside no facto de a produção de software tradicional ser inadequada e o processo utilizado mais difícil de implementar e documentar.

Programação reactiva – A produção de software para uma interface deve ser feita para que o utilizador possa exercer controle sobre a aplicação e não deixar que seja a aplicação a fazê-lo. Este tipo de software acarreta uma maior dificuldade na modelação e organização do código.

Multiprocessamento – Na nossa interface podem existir diversas tarefas que possam ser executadas simultaneamente, sendo que o tempo de execução dessas tarefas varia. É, portanto, fundamental que o nosso sistema dê respostas instantaneamente. Desta forma, os problemas relacionados com multiprocessamento, sincronização, controle de concordância e consistência das informações, sejam tidos em conta quando se implementa uma interface.

Programação em tempo real – Existem limitações importantes tanto a nível do tempo de resposta como no redesenho das informações que serão apresentadas pela interface. Estas características não são, na maioria dos sistemas, tidas em conta.

Necessidades de robustez – todas as entradas dos utilizadores devem ser atendidas satisfatoriamente. Este facto difere muito da “robustez costumeira” que se deseja num software. As interacções feitas pelos utilizadores com o nosso sistema podem ser feitas das mais diversas maneiras, o que torna muito difícil prever todos os tipos de interacções. As necessidades de cancelar e desfazer as acções dos utilizadores são um factor adicional à implementação da nossa interface.

¹ Usabilidade: conjunto de características todas elas centradas no utilizador.

Dificuldade em sistematizar testes – os sistemas que exercem manipulação directa sobre a interface são, na sua grande maioria, pouco úteis para efectuar testes, já que estes não simulam de forma correcta as actividades exercidas pelos utilizadores. Neste sentido, alterações na interface implicam mudanças nos testes. Além deste facto, grande parte da qualidade da interface não pode ser avaliada pelo sistema, necessitando também da apreciação de humanos.

Falta de suporte por linguagens de programação – As linguagens de programação mais antigas não possuem primitivas internas para comunicação com interfaces mais complexas. As necessidades de multi-programação e a reactividade do software requerem suporte pelas linguagens de programação, o que muitas das vezes não acontece.

Complexidade das ferramentas – Quando as linguagens de programação não conseguem dar resposta às necessidades da interface cria-se uma necessidade de procurar ferramentas que resolvam esta situação. De uma forma geral, estas ferramentas são incompatíveis e requerem muitas horas de treino por parte dos programadores.

Dificuldade de modularização – Uma das maiores dificuldades relacionadas com a criação da interface está no facto de não se conseguir separar a forma da interface com o conteúdo a ser apresentado e, na maioria das vezes, mudanças na interface implicam mudanças na funcionalidade da aplicação.

Voltando novamente à Ergonomia, e tal como referimos anteriormente, esta relaciona-se com a interacção entre situações de trabalho e tecnológicas com o homem, sendo que a ergonomia envolve, neste processo, ciências como a Anatomia, a Filosofia e a Psicologia, com a finalidade de potenciar as capacidades humanas e a manutenção do bem-estar e a defesa da saúde. Na verdade, a ideia base prende-se com o facto de se tentar adaptar o trabalho à pessoa que o executa de forma ajustada.

A utilização da Psicologia serve de adaptação cognitiva situacional na forma como a informação é processada e como é feito o processo da tomada de decisão.

A optimização do Sistema de Informação é, desta forma, o nosso objectivo primordial. Segundo o Laboratório de Ergonomia e *Usabilidade* de Sistemas Homem - Máquina, a única e específica tecnologia da ergonomia é a tecnologia da interface homem máquina, ou seja a ergonomia desenvolve conhecimentos sobre as capacidades, limites e outras características de desempenho humano, que se relacionam com o interface entre indivíduos e outros elementos do sistema. Desta forma, a tarefa consiste na avaliação Ergonómica do hardware ligada ao design de produtos e à sua utilização e, principalmente, à ergonomia do software, ligada à interface com sistemas e à utilização de programas.



Parte 2

Conhecimento dos Utilizadores

“O mundo inteiro é um palco, onde todos os homens e todas as mulheres são meros participantes: Eles têm suas saídas e suas entradas; e cada um por sua vez, desempenha muitos papéis (...)”

Shakespeare, Adaptado por Yourdon, 1999, p. 49

5. Soluções Centradas no Utilizador (SCU)

UCS, is both a technique and a philosophy that puts the user's needs ahead of anything else.
Mark Pearrow(2000)

Os projectos centrados nos utilizadores têm como finalidade fazer uma aproximação que sirva de suporte para o processo completo de desenvolvimento de aplicações com actividades centradas nos utilizadores, tornando essas aplicações fáceis de utilizar e proporcionando a estes algum valor acrescentado.

Tal como refere Mark Pearrow (2000) as SCU são, simultaneamente, técnicas que põe à frente de tudo as necessidades e expectativas dos utilizadores. As pesquisas feitas nas organizações indicam que a maioria dos projectos falham porque as exigências e as necessidades dos utilizadores não são levadas em consideração. Por outro lado, a manutenção de um sistema que se encontre a funcionar sem levar em consideração estas necessidades pode custar cem vezes mais do que um sistema que foi projectado e desenvolvido segundo as exigências dos utilizadores.

Infelizmente, e como já referimos, o que é melhor para o utilizador nem sempre é natural para o designer e, pelo contrário, o que é fácil para o utilizador é, na maioria dos casos, difícil para o designer ou para o programador do projecto. É pois fundamental que o programador e o designer conheçam as necessidades específicas do utilizador. Para que esse conhecimento seja possível é necessário compreender o comportamento humano e conhecer as características da classe de utilizadores que vão utilizar o sistema.

Actualmente, as linguagens de programação são muito poderosas e os motores de base de dados muito potentes. Aliados a estes factos, temos os grandes desenvolvimentos a nível de conectividade e das redes de computadores que nos permitem criar sistemas mais rápidos e com maiores capacidades de armazenamento. Com estas inovações surge um novo paradigma: construção de sistemas centrados no utilizador. (fig. 2)

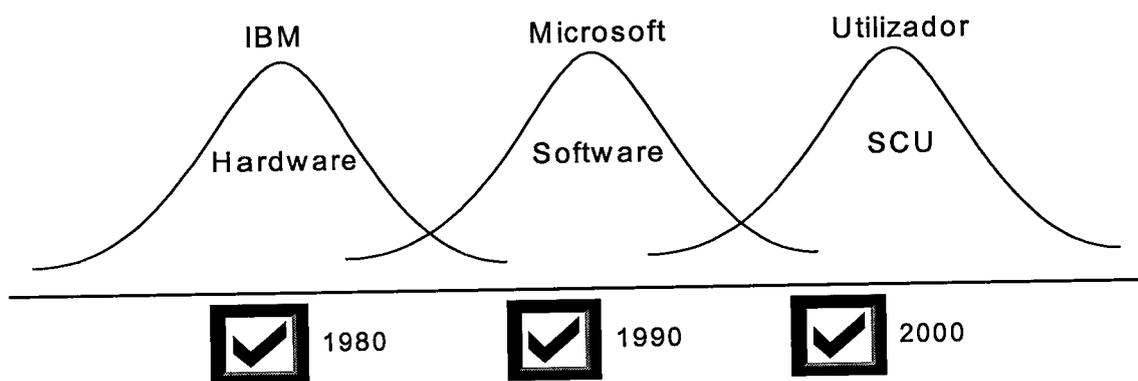


Figura 2 – Evolução histórica da produção de sistemas computacionais, adaptado de “User – Centred Solutions”, Eric M. Schaffer, Human Factors International, pag 7.

5.1. Conhecimento dos Utilizadores

Idealmente, as actividades SCU devem incluir entrevistas com os utilizadores, observação directa do seu trabalho, realização de uma análise do perfil e das necessidades dos utilizadores, proceder a uma análise de tarefas e, como não poderia deixar de ser, analisar o fluxo de informação. Se estas actividades forem bem desenvolvidas é dado como certo o sucesso do nosso sistema.

É também fundamental que os utilizadores sejam envolvidos no projecto, uma vez que estes conhecem como ninguém o seu trabalho. Estes são uma preciosa ajuda para o designer e para o analista, permitindo que se tenha um conhecimento razoável do negócio em que o sistema irá actuar. Desta forma, o entendimento do universo e o meio envolvente onde estes se inserem passa a ser uma tarefa mais fácil e mais próxima da realidade. Não adianta estar a construir um sistema de última geração e com diversos recursos se a interface é difícil de ser compreendida e a sua utilização difícil de ser apreendida.

Os utilizadores, ao serem envolvidos na construção do sistema, ficam motivados, o que permite recolher informações importantíssimas para a equipa de projecto. Informações estas que nos permitem, por exemplo, saber que tarefas necessitam os utilizadores de efectuar e com que frequência e sob que condições essas tarefas são realizadas.

Também se torna importante para os utilizadores que o sistema possua as palavras e conceitos utilizados por estes. É um detalhe simples, mas que pode fazer uma grande diferença na utilização do sistema.

A atenção deve estar voltada para a preservação da consistência entre a realidade e o virtual. Não adianta utilizar a mesma nomenclatura do universo dos utilizadores e depois forçar um significado diferente ao conhecido para realizar as necessidades do sistema.

Também é importante imaginar os passos e reacções que os utilizadores irão ter, catalogando todas as acções inesperadas e informações incorrectas que possam ser escritas no nosso sistema. É fundamental procurar sempre antecipar os erros e mal entendidos que o utilizador possa vir a ter em relação às funcionalidades e os elementos da nossa interface.

5.1.1. Princípio

Quando se está a projectar uma interface para interacção sobre um determinado tipo de software, existe um princípio que deve ser levado em consideração, que é o de deixar sempre o sentimento por parte do utilizador que detém o controlo sobre o software. Este princípio, tal como diz Eric Schaffer (2000), acarreta três implicações fundamentais. A primeira tem a ver com o facto de ser o utilizador a iniciar uma acção e não o computador ou o software. O utilizador deve possuir um papel activo e não reactivo. A segunda implicação diz respeito aos utilizadores. Como vimos estes têm preferências e capacidades de utilização do software diferentes uns dos outros, logo devemos ter o cuidado de possibilitar a personalização de alguns aspectos da interface. A terceira implicação relaciona-se com a interactividade do software e com a capacidade de resposta que este deve ter para com o utilizador. Devemos evitar tempos de processamento muito elevados e, se tal não for possível, devemos fazer com que a evolução do processamento seja visível e resulte de uma escolha do utilizador. Deve-se ainda permitir que a interrupção da escolha possa ser feita a qualquer momento pelo utilizador.

5.1.2. Regras a seguir do DCU

Durante o do processo de desenvolvimento de soluções centradas nos utilizadores existem quatro regras fundamentais que devem ser levadas em consideração. A primeira destas regras diz respeito à alocação apropriada das funções entre o sistema e o utilizador. Devem ser clarificados os aspectos referentes as tarefas que são suportadas pelos utilizadores e pelo sistema. Outra regra a ter em consideração é, como já referimos, o envolvimento activo dos utilizadores. A utilização de pessoas com conhecimentos reais do contexto em que a aplicação vai ser utilizada permite uma melhor aceitação e comprometimento dos utilizadores com o sistema. A terceira regra a ter em conta é a criação de equipas multi-disciplinares que irão potenciar o desenvolvimento activo de várias pessoas com diferentes perspectivas e experiências. Com um leque tão alargado de pessoas com motivações e interesses diferentes podem surgir mais ideias e soluções inovadoras. No entanto, podem também causar alguns problemas de comunicação e desenvolvimento do produto. Para evitar este tipo de complicações importa definir desde logo os objectivos do nosso sistema. A última regra diz respeito à interacção das soluções de desenho. É necessário existir um feedback contínuo com os utilizadores. Utilizando técnicas de prototipagem, por exemplo, é possível que eles participem activamente na construção das interfaces necessárias ao sistema e que estas se ajustem às suas necessidades.

5.2. Processo de Desenvolvimento Centrado no Utilizador.

O processo de desenvolvimento de soluções centradas nos utilizadores passa por várias actividades (etapas) que não podem deixar ser tomadas em consideração. Nuno Jardim Nunes (2003) descreve essas actividades da seguinte forma:

- ***Perceber e especificar o contexto de utilização.***

A qualidade do sistema (incluindo a segurança) depende muito do contexto em que o sistema vai ser utilizado. Os aspectos que devem ser conhecidos sobre o contexto incluem:

- o As características dos utilizadores;

- As tarefas que estes executam;
- O ambiente em que os utilizadores trabalham;

- ***Especificar os Requisitos² Organizacionais e dos Utilizadores***

Os requisitos centrados nos utilizadores devem ser especificados ao mesmo tempo que os requisitos técnicos³ e funcionais⁴. Estes incluem:

- Identificação do grupo dos utilizadores;
- Identificação dos objectivos do desenho;
- Identificação das prioridades apropriadas para os diferentes requisitos;
- Identificação dos diferentes benchmarks⁵ sobre os quais o sistema pode ser testado;
- Evidência da aceitação dos requisitos pelos diferentes parceiros;
- Conhecimento dos diferentes requisitos estatutários⁶ e legais.

- ***Desenhos e protótipos do sistema***

- Simular as diferentes alternativas de desenho utilizando diferentes técnicas de prototipificação desde as fases iniciais de desenvolvimento;
- A prototipagem aumenta entre a equipa de desenvolvimento e os utilizadores finais, funcionando como uma alternativa “barata” para explorar alternativas de desenho;

² *Requisitos*: Expressa as características e restrições do produto de software do ponto de vista da satisfação das necessidades do utilizador.

³ *Requisitos técnicos*: são definidos a partir dos funcionais, ou seja para cada componente do projecto, o gestor deversa ser capaz de eleger as tecnologias mais indicadas, tanto no que diz respeito ás capacidades de cada tecnologia bem como as suas restrições, avaliando a mais apropriada.

⁴ *Requisitos funcionais*: definem o que cada componente do projecto deversa ter, como ele irá responder e sob que condições. Para a sua obtenção é necessário uma comunicação, sem obstruções com profundidade e clareza, com os que detêm o conhecimento.

⁵ *Benchmarks*: permite comparar o resultado com os objectivos estabelecidos, indicando as áreas que necessitam de mais atenção.

⁶ *Requisitos estatutários e legais*: requisitos existentes na organização que tem de ser mantidos. Ex: Regulamento interno.

- ***Implementar testes centrados nos utilizadores***

A única forma de confirmar que os objectivos foram atingidos e de identificar novas oportunidades para melhorar o desenho é recorrer aos testes centrados nos utilizadores. Estes testes têm as seguintes fases:

- o Desenvolvimento de um plano de testes;
- o Recolher e analisar os dados;
- o Reportar os resultados e recomendações para alterações;
- o Alterar até que o desenho cumpra os objectivos;
- o Rastrear as alterações, manutenção;

Desta forma, o processo de desenvolvimento centrado no utilizador pode ser separado em duas fases importantíssimas. A primeira fase, que se designa de projecto, inclui o estudo detalhado do utilizador e das suas necessidades e da tecnologia que irá servir de suporte à construção da aplicação. A segunda, fase intitulada implementação, inclui a implementação propriamente dita, tendo em conta o estudo feito na primeira fase.

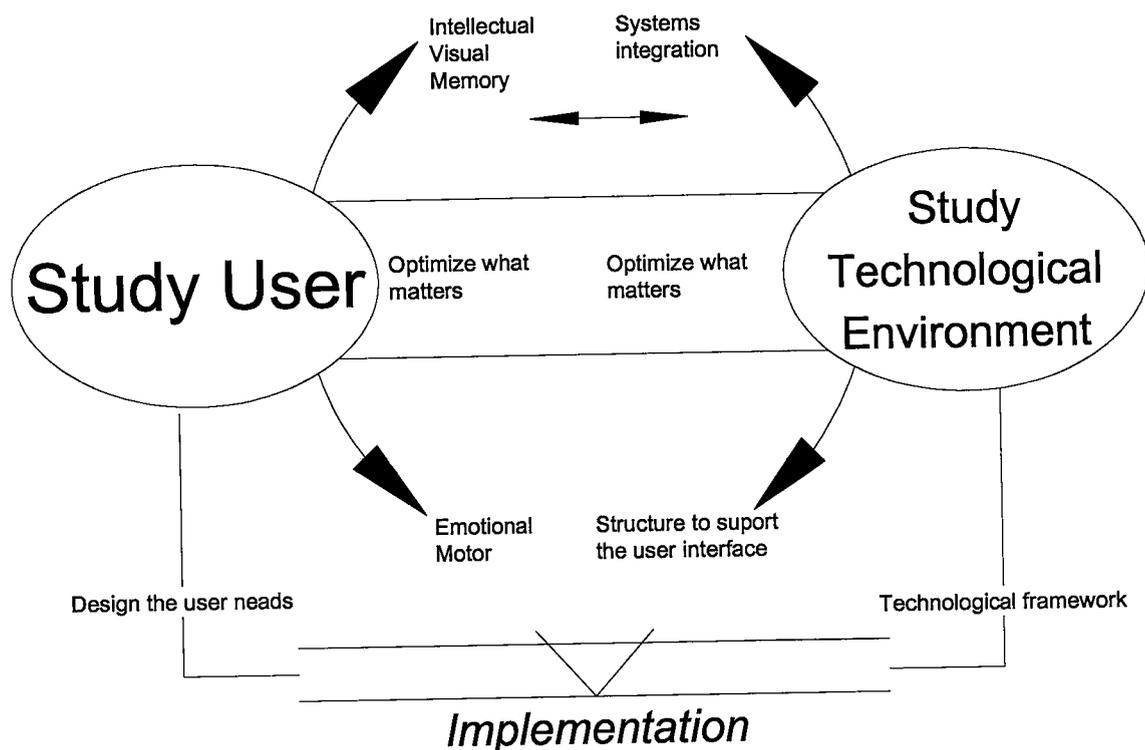


Figura 3 - Processo de DCU

Na figura 3 está esquematizada a fase de projecto. Nesta fase, para além de se fazer o estudo do utilizador tendo em conta os aspectos visuais, intelectuais, emocionais motores e de memória próprios de cada utilizador, assim como de se realizar o estudo da tecnologia de forma a integrar o sistema que irá servir de suporte à interface, também tem que ser criado o design da aplicação tendo em conta as necessidades tecnológicas e dos utilizadores. Depois deste trabalho feito e de criada a estrutura da interface passamos à segunda fase. A segunda fase (implementação) não é mais do que a implementação propriamente dita da interface, tendo em conta o estudo feito na fase anterior e a utilização nessa implementação de standards de interface. De salientar ainda que nesta fase, e ao longo das várias fases que compõem a implementação, terão de ser realizados constantemente testes de *Usabilidade* de forma a garantir o sucesso da aplicação e a perfeita aceitação por parte dos utilizadores. No que diz respeito aos testes de *Usabilidade*, estes irão ser tratados na parte quatro desta tese.

Parte 3

Avaliação Heurística e Testes aos Utilizadores

6. Avaliação Heurística

A avaliação heurística é um dos métodos mais importantes de análise da *Usabilidade*. Trata-se de um método de inspeção sistemática da interface do utilizador em relação à *Usabilidade*. Existem várias técnicas de avaliar a *Usabilidade* de uma interface. Entre as existentes, as mais utilizadas são, sem sombra de dúvida, a avaliação heurística e a dos testes com utilizadores. A avaliação heurística é uma das formas mais económicas e práticas de avaliar a *Usabilidade* numa interface, permitindo detectar problemas na fase inicial de desenvolvimento da interface. Desta forma, evitam-se erros desnecessários encontrados depois da interface estar concluída, com resultados óbvios na redução de custos de manutenção. Esta avaliação deve ser feita por uma equipa de especialistas que podem ser engenheiros da *Usabilidade*, designers, utilizadores ou programadores. Este tipo de avaliação poderá ser feita com os colegas de trabalho, reduzindo os custos da avaliação, mas nunca por uma única pessoa, uma vez que esta não poderá levantar todas as questões possíveis para as heurísticas.

6.1. Heurísticas

Em traços gerais, a avaliação heurística é feita através de um avaliador (equipa) que interage com a interface e julga a sua adequação comparando-a com os princípios de *Usabilidade* reconhecidos através das heurísticas. Jacob Nielsen (2003) sugere 10 tipos de heurísticas que devem estar presentes em qualquer tipo de avaliação. Essas heurísticas são:

1- Falar a Linguagem dos Utilizadores

Os diálogos devem ser expressos por palavras ou conceitos familiares aos utilizadores do sistema, ou seja o nosso sistema deve estar preparado para falar a linguagem do utilizador e não o utilizador falar a linguagem do sistema.

- Eu entendo a terminologia do sistema?

2- Diálogos Simples e Naturais

A informação contida nos diálogos deve ser relevante e não deve conter informação desnecessária, uma vez que esta poderá diminuir a visibilidade relativa.

- Eu percebo rapidamente a informação transmitida?

3- Minimizar a sobrecarga de memória do utilizador

O software deve permitir que o utilizador execute uma determinada sequência de operações sem que seja necessário recorrer à memória para realizar um ou outro comando específico de uma dada tarefa.

- Eu consigo reconhecer os elementos que constituem a interface? O design está nítido e apelativo?

4- Consistência

A consistência é um dos princípios básicos da *Usabilidade*. Se um utilizador estiver consciente que, para realizar uma dada tarefa, tem que executar uma sequência de passos bem definida e que esta sequência se repete em interfaces diferentes aumentamos o grau de confiança na utilização da interface.

- Não necessito de aprender novos truques?

5- Feedback

O sistema deve manter constantemente o utilizador informado sobre as suas acções em tempo útil.

- Sei o que o sistema está a fazer?

6- Navegação

Regularmente os utilizadores escolhem funções erradas. É necessário que o sistema permita uma saída rápida dessa situação, permitindo que utilizador adquira confiança e aprenda por exploração.

- Eu encontro o caminho em qualquer lugar onde me encontre?

7- Atalhos

O sistema deve permitir formas rápidas de realizar tarefas para utilizadores experientes.

- Consigo fazer aquilo que necessito de forma rápida?

8- Boas mensagens de Erro

As mensagens de erro devem ser descritas numa linguagem clara e precisa. Estas devem ser construtivas e levarem a soluções.

- Este erro é fácil de corrigir?

9- Prevenir erros

Melhor que ter boas mensagens de erro é existir um sistema que à partida prevê a não ocorrência do erro.

- O sistema não me deixa cometer erros?

10- Ajuda e Suporte

O ideal é que o nosso sistema não necessite de recorrer à ajuda. Mas esta deve existir e deve ser de fácil acesso e ser focalizada nas tarefas a executar pelo utilizador.

- Eu obtenho ajuda quando necessito?

Durante a avaliação heurística é pedido ao grupo de avaliadores que faça uma avaliação da interface e faça uma comparação com as heurísticas propostas por Nielsen. Os comentários e resultados obtidos serão de vital importância para a construção final da nossa interface.

Para além das heurísticas propostas por Nielsen, podem também ser criadas questões relacionadas com cada uma destas heurísticas. Estas questões podem ser levantadas por qualquer pessoa e permitem fazer uma simulação do comportamento do utilizador perante o sistema.

Como não nos é possível detectar todos os problemas através das heurísticas, os testes com os utilizadores permitem validar os testes de heurística e corrigir outros erros que não foram detectados. Os testes aos utilizadores, contrariamente às heurísticas que são feitas com base no design do produto que estamos a desenvolver, estes necessitam, pelo menos, da existência de um protótipo onde se possam fazer os

testes, reduzindo-se custos de uma alteração futura mais profunda no programa. Este tipo de testes serão estudados mais adiante.

6.1.1. Justificativa

Robert Bailey, da Human Factors International, defende que a avaliação heurística é uma forma barata de avaliar a *Usabilidade* numa interface, sendo que muitos dos problemas encontrados por este método não são na realidade problemas e que parte deles derivam de os avaliadores manterem a mesma lista de heurísticas ultrapassada. Perante os argumentos apresentados, Jakob Nielsen, defensor e principal impulsionador da avaliação heurística, responde que é importante distinguir problemas maiores de problemas menores em *Usabilidade*, sendo que a avaliação heurística é um método excelente para detectar este tipo de problemas, ainda que os problemas maiores sejam os primeiros a serem encontrados. (Miguel Santinho, 2001)

Quanto aos problemas menores, estes são classificados por Nielsen numa escala de gravidade resultante de 3 factores. A partir dessa correlação determina-se então a gravidade do problema. A escala de classificação, bem como a escala de gravidade do problema, é descrita na tabela seguinte.

<p>1- Frequência com que ocorre o problema</p> <p>1- Raramente 2- Quase sempre 3- Sempre</p>	<p>Gravidade do problema</p> <ul style="list-style-type: none"><input type="checkbox"/> 0 - Não é (ou não há acordo entre os avaliadores) de todo um problema de usabilidade.<input type="checkbox"/> 1 - Apenas um problema cosmético, não necessita de ser corrigido a menos que se disponha de tempo.<input type="checkbox"/> 2 - Problema menor: deve ser dada baixa prioridade à resolução deste problema.<input type="checkbox"/> 3 - Problema maior: deve ser dada alta prioridade à resolução deste problema.<input type="checkbox"/> 4 - Catástrofe: a correcção deste problema é imperativa.
<p>2- O impacto do problema e a facilidade para o utilizador o resolver</p> <p>1- Fácil 2- Difícil</p>	
<p>3- A persistência do problema</p> <p>1- É um problema que os utilizadores podem resolver após a primeira ocorrência. 2- É um problema que ocorre repetidamente</p>	

Tabela 1- Escala de classificação da Avaliação Heurística. Adaptado de Miguel Santinho (2001)

As heurísticas propostas acarretam necessariamente características próprias que têm de ser avaliadas. Por exemplo, quando avaliamos a heurística numero 4 (Consistência) proposta por Nielsen, esta acarreta consigo um conjunto de características que devem ser validadas, nomeadamente:

- Verificar se ícones iguais têm acções iguais;
- Ícones universalmente conhecidos têm o mesmo significado;
- Coerência visual (tipos de letra e utilização de cores consistente);
- Em formulários, as etiquetas estão próximas das caixas de resposta;
- Etc.

Como podemos constata, cada heurística é composta por um conjunto de características que devem ser avaliadas. Estas características irão ser exploradas nesta tese quando definirmos o modelo de avaliação da nossa interface. As diferentes características têm, naturalmente, graus de gravidade⁷ associados. Miguel Santinho (2001) propõe uma escala de avaliação do grau de gravidade (0 a 4) para cada característica da heurística. Para determinar o índice de gravidade de cada uma das heurísticas é proposta a seguinte formula:

$$igH = \frac{\sum_{i=1}^{Th} gr_i}{\frac{\sum_{i=1}^{Th} g \max - \sum_{i=1}^{gr=NA} g \max}{Th}}$$

Onde:

igH = Índice da gravidade de heurística.

gr = Gravidade registada de heurística.

g max = Gravidade máxima de cada heurística.

NA = Heurística não aplicável.

Th = Total das Heurísticas (características).

Se considerarmos a heurística 9 (Prevenção de erros), que tem quatro características a validar, os níveis máximos de gravidade são representados na tabela seguinte:

Heurística	9 – Prevenção de Erros		
		Gravidade máxima	Gravidade registada
Característica 1	São evitados erros de sistema	4	3
Característica 2	São evitados erros imprevistos	4	2
Característica 3	Todos os links têm destino válido	3	NA
Característica 4	As páginas não têm links para elas próprias	2	1

Tabela 2 – Características da Heurística

Se aplicarmos a formula anterior de forma a determinar a gravidade desta heurística:

$$igH = \frac{3+2+1}{\frac{13-3}{4}} = 2,4$$

Chegamos à conclusão que o índice de gravidade da Heurística 9 é 2, ou seja representa um problema menor ao qual deve ser dada baixa prioridade.

6.2. Testes com os utilizadores

Como referimos anteriormente, os testes com os utilizadores, além de validarem as heurísticas atrás descritas, permitem também corrigir outro tipo de erros que não foram ainda detectados. Para que os testes decorram da melhor forma possível é necessário criar condições específicas, ou seja deve ser criado um laboratório de testes equipado com diversos postos de trabalho que simulem na perfeição o ambiente real de execução.

Para cada posto de trabalho, Nielsen (1994) sugere que deverá existir um responsável pela observação que terá como função a recolha das notas de todas as acções e movimentos efectuados pelo utilizador. A posição deste deverá ser feita cautelosamente para que não interfira na utilização normal das interfaces por parte dos utilizadores, caso contrario influenciará os resultados.

⁷ Grau de gravidade: índice pelo qual deve ser avaliada a gravidade do nosso problema. Os valores estão, normalmente, tabelados e a resolução do problema deve variar de acordo com a sua gravidade.

As avaliações devem ainda ser gravadas, de forma a permitir uma análise posterior mais detalhada por parte dos avaliadores. O utilizador deverá testar a interface tendo por base um conjunto de acções que foram definidas como princípios heurísticos. Para cada tarefa a avaliar serão tomados em consideração factos como: tempo de realização da tarefa, movimentos efectuados com o rato, hesitações ao realizar as tarefas, etc. Os utilizadores irão executar um conjunto de tarefas que serão analisadas pelos avaliadores e que darão lugar a um estudo estatístico de todos os dados recolhidos. Este estudo permite iniciar o processo de rectificação dos problemas detectados.

Segundo Jared Spool (1997), os utilizadores devem responder as questões com graus de complexidade distintos, permitindo desta forma recolher as reacções a diferentes solicitações. Desta forma, o autor propõe quatro tipos de questões:

1. “**Simple facts:** the simplest type of question is a single fact retrieval, a question for which there is only one correct answer.”
2. “**Comparison of facts:** the user must research two or three facts, then compare them to arrive at the answer.”
3. “**Judgment:** require the user to decide whether he or she has found enough information to give a satisfactory answer.”
4. “**Comparison of Judgment:** involving both comparisons and judgments.”

Spool, Jared, “Web Site Usability”, 45

O primeiro tipo diz respeito a uma resposta simples, ou seja a resposta correcta é apenas uma e não oferece qualquer dúvida. A segunda resposta terá de ser feita utilizando uma comparação simples, em que a escolha só poderá ser uma e está presente na interface que estamos a utilizar. A terceira questão é uma questão complexa e tem que estar relacionada com o contexto em que se está inserido. Por fim, a resposta à última questão, ou último grau, tem a ver com a capacidade de resposta relacionada com os vários tipos de interface presentes na nossa aplicação.

Para cada tarefa efectuada deve ser calculado o índice de sucesso⁸ de cada tarefa. Miguel Santinho propõe a seguinte fórmula:

$$isT = \frac{3 * \sum C + 2 * \sum E + \sum NR}{nu} \text{ Onde:}$$

C, corresponde ao total de tarefas realizadas com sucesso;
E, corresponde ao total de tarefas realizadas com insucesso;
NR, corresponde ao total de tarefas não realizadas;
nu, número de utilizadores;

O índice de sucesso de cada tarefa varia numa escala de 1 a 3, sendo o 1 o mínimo e 3 o máximo.

6.2.1. Avaliadores

Como vimos anteriormente, a avaliação heurística é feita por um grupo de avaliadores que faz uma avaliação da interface e compara com as heurísticas propostas por Nielsen. Os comentários e resultados obtidos serão de vital importância para a construção final da nossa interface. Mas surge um problema logo à partida para quem quer realizar uma avaliação heurística de uma interface, que se prende com o facto de definir o número de avaliadores a utilizar. Segundo estudos feitos por Nielsen (2004), verifica-se que o crescimento da probabilidade de encontrar problemas de *Usabilidade* é exponencial e que o valor ideal se situa entre 10 e 15 avaliadores, fig.: 4

⁸ Índice de sucesso: rácio que nos dá a taxa de sucesso de realização de uma dada tarefa

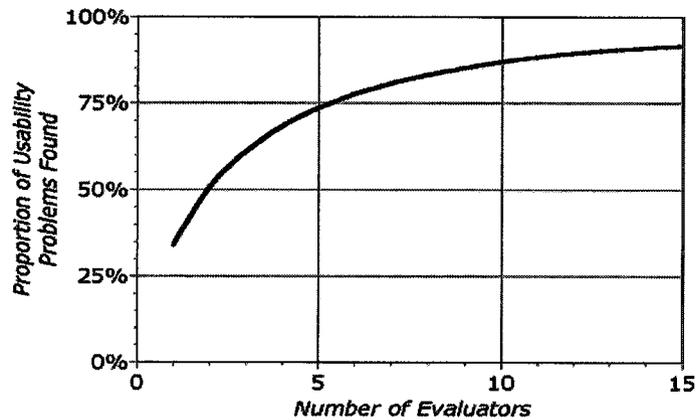


Figura 4 – Curva demonstrativa da proporção dos problemas de *Usabilidade* encontradas na interface utilizando a Avaliação Heurística com vários avaliadores. Adaptado de Nielsen “Heristic Evaluation”, 4.

Analisando a curva do gráfico encontrada pelos estudos efectuados por Nielsen podemos constatar que, utilizando apenas um avaliador, os problemas encontrados não são muito significativos. Com cinco avaliadores já conseguimos encontrar cerca de 75% dos problemas. Neste sentido, tal como sugere o autor, é recomendável utilizar cinco avaliadores. Na impossibilidade de utilização de cinco utilizadores nunca se devem utilizar menos de três. Mais avaliadores só devem ser utilizados quando se verificar que a *Usabilidade* no nosso sistema é um factor crítico para o seu sucesso. Através desta Análise é possível estabelecer uma relação de custos e eficácia, permitindo encontrar o número de observadores indicado à dimensão do projecto em que se está a trabalhar e ao orçamento previsto para a realização dos testes. Desta forma, segundo Cybis (2004), podemos considerar que para identificar a maior parte dos problemas ergonómicos das interfaces é necessário utilizar mais de 3 avaliadores com perfis e competências diferentes. Porque, tal como referimos, os resultados dependem directamente das pessoas que realizam as avaliações, mais concretamente das experiências e conhecimentos que os avaliadores possuem e os aplicam na avaliação da interface.

Parte 4

Testes de Usabilidade

7. Testes de Usabilidade

Para além da avaliação heurística e dos testes com utilizadores, os testes de *Usabilidade* constituem uma avaliação rigorosa do nosso sistema. Estes visam medir, quantificar e comparar o nosso sistema de forma a identificar problemas de *Usabilidade*. Estes testes devem ser efectuados durante o processo de desenvolvimento de software para identificar problemas de *Usabilidade* de software já existente e proceder à sua correcção.

As medidas obtidas com estes testes têm de ser relacionadas com *eficiência* (medido o grau de produtividade do utilizador), *eficácia* (medido a rapidez com que o utilizador executa o seu trabalho, tendo em conta os recursos disponíveis) e *satisfação* (medindo o grau de satisfação ou frustração na utilização do produto). (Teofanos M., Laskoski S., 2004). Desta forma, podemos dizer que um teste de *Usabilidade* visa medir e quantificar a *Usabilidade* de uma determinada interface, medindo, por exemplo, o tempo que um utilizador leva a completar uma tarefa, o número de enganos que este teve para a efectuar, o número de utilizadores que cometeram os mesmos erros e a quantificação de quantos utilizadores cometeram o mesmo erro por minuto ou por tarefa.

Os Testes de *Usabilidade* funcionam como a Avaliação Heurística, ou seja baseiam-se na observação directa do utilizador em interacção com a interface em estudo. O responsável pelo teste deve apenas observar (exposto ou não) como o utilizador executa as tarefas, pedindo a este que explique os seus pensamentos em voz alta e registando o seu comportamento. Este responsável deve ainda preparar um roteiro de acções que o utilizador deve efectuar.

Existem muitas formas de realizar os testes de *Usabilidade*, existindo mesmo software específico que recolhe, por exemplo, informação do movimento efectuado pelos olhos do utilizador e das expressões e atitudes que este tem ao utilizar a interface. Outra forma de realizar os testes é designado *Design Participado* (*Participatory Design*⁹), no qual se realiza um brainstorming¹⁰ no qual participam os

⁹ *Participatory Design* : área de pesquisa que tem como premissa a ideia de que os utilizadores do sistema desempenham um papel crítico no projecto. O design participado pode ser entendido como uma abordagem de projecto centrado no utilizador. Porém, as suas origens mais modernas são atribuídas à

utilizadores, os designers, e os programadores. Em conjunto tenta-se chegar a uma solução.

7.1. Os atributos da Usabilidade

A *Usabilidade* é definida na norma ISO 9241 como a forma que um produto pode ser utilizado por utilizadores específicos de forma a atingir objectivos específicos com eficácia, eficiência e satisfação, num contexto de utilização específico.

A implementação de testes de *Usabilidade* vai inevitavelmente tornar o nosso sistema mais caro de implementar. Todavia, tal como refere Jacob Nielsen, o retorno do investimento feito é extremamente lucrativo quando se desenvolvem interfaces de acordo com os utilizadores. Isso pode ser feito a partir de uma perspectiva de engenharia ligada à ideia de optimização, ou seja maximizando os resultados e minimizando os recursos. Desta forma, surge o conceito de Engenharia de *Usabilidade* que não é mais do que o esforço que as organizações têm feito para o desenvolvimento da *Usabilidade* dos sistemas interactivos.

Esta engenharia necessita inevitavelmente de atributos e componentes que permitem uma aproximação sistemática da *Usabilidade* como uma disciplina de engenharia. Logo, a aplicação de princípios científicos à construção de produtos e os testes e as avaliações aos componentes permitem produzir interfaces homem máquina com *Usabilidade*. Os atributos e métodos recomendados para os testes de *Usabilidade* são os seguintes (Marques Cardoso, 2003):

- É importante que os utilizadores, as tarefas e o ambiente utilizados no teste sejam representativos da utilização que se pretende.

ocidentalização do chamado projecto cooperativo ou modelo escandinavo. Esta abordagem, embora se preocupe com a participação dos utilizadores, dá principal ênfase às relações de equipa para permitir uma efectiva participação.

¹⁰ Brainstorming: forma colectiva de geração de novas ideias através da contribuição e participação de um grupo. Este conceito tem por base a hipótese de um grupo gerar mais ideias do que um indivíduo. Pode ser um método importante na inovação, permitindo o desenvolvimento de pensamentos criativos e ideias promissoras.

- ❑ Seleccionar as tarefas e grupo de utilizadores mais importantes para serem testados.
- ❑ O número de utilizadores seleccionados deve ser representativo. Normalmente, três a cinco utilizadores são suficientes para identificar problemas.
- ❑ Construir tarefas e escrever instruções para que possam ser facilmente apreendidas pelos utilizadores.
- ❑ As sessões deverão ser devidamente planeadas: fornecer instruções, passar o teste, responder a um questionário e fazer uma nova entrevista após o teste.
- ❑ Convidar investigadores para que observem as sessões. Se tal não for possível, gravar as sessões em vídeo para que possam ser vistas posteriormente.
- ❑ Normalmente, são contactados dois administradores, um para interagir com o utilizador e outro para anotar problemas ou falar com os observadores.
- ❑ Caso seja possível, deve utilizar-se uma sala de teste ligada por vídeo a outra onde os utilizadores serão observados.
- ❑ No caso se serem necessárias medidas de *Usabilidade*, deve-se observar o utilizador sem que se faça qualquer comentário.
- ❑ Se as medidas não forem necessárias pedir-se-á ao utilizador que faça comentários relativamente aos conteúdos de cada écran e o motivo pelo qual fez determinadas escolhas.

Como vimos, a melhor forma de avaliar a *Usabilidade* de um sistema é observar utilizadores reais e acompanhá-los na execução de tarefas. Este é o método mais económico, mais simples e fidedigno. Bastando, para tal, cinco utilizadores a executar pequenos testes orientados.

No final dos testes, os consultores deverão elaborar um **relatório detalhado** contendo uma lista de tarefas, de acordo com as necessidades dos utilizadores. Este relatório deve conter uma breve *descrição das metas propostas e do que foi feito para a sua concretização*, deve também conter uma *análise do comportamento dos*

utilizadores: como é que estes se comportaram, se sorriam, se faziam cara de sofrimento, se faziam muitas perguntas, etc. É importante que o relatório contenha um *resumo das entrevistas* feitas durante o teste, registando quais as principais opiniões, perguntas, dúvidas, reclamações e sugestões dadas pelos participantes após o teste. Como não poderia deixar de ser o relatório deveria listar os *problemas de Usabilidade* encontrados, sendo cada problema tratado de forma isolada com descrição dos critérios heurísticos que este viola, o grau de severidade e o custo da sua resolução ou não. Para finalizar, o relatório deve conter ainda um *plano de correcção* mp qual serão propostas correcções e custo-efeito para cada problema encontrado. De salientar que cada solução dada no relatório deve sempre ser feita tendo em conta a relação de custo-benefício para os utilizadores da aplicação. Os problemas encontrados com grau de severidade baixo devem ser ignorados e deve ser feito um plano de intervenção para que os problemas que contenham um grau de severidade elevado (factores considerados críticos) possam ser solucionados de imediato.

Parte 5

**Construção (modificação) da Interface
Processo de Análise**

*“O homem é um animal utilizador de ferramentas...Sem ferramentas ele não é nada,
com ferramentas ele é tudo”*

*Thomas Carlyle,
Sartor Resartus, volume I, Capit.4*

8. Análise

Tal como referimos nos capítulos anteriores, tanto na avaliação heurística como nos testes com os utilizadores e nos testes de *Usabilidade*, o utilizador é a “peça” fundamental para o sucesso do nosso sistema. Neste capítulo iremos referir aspectos que devem ser tomados em consideração quando estamos a analisar um utilizador ou um grupo de utilizadores. Outro aspecto não menos importante diz respeito à análise do trabalho, ou seja, o estudo das tarefas e funções efectuadas pelos utilizadores no seu dia a dia. A perspectiva de análise não é mais do que o esforço feito para a compreensão da realidade e o contexto operacional de um sistema.

8.1. Análise do utilizador

Independentemente da abordagem ou a estratégia que iremos utilizar para o desenvolvimento do nosso sistema, a primeira coisa que deveremos fazer é definir as características dos utilizadores, ou seja identificar os actores na linguagem UML¹¹, (pessoas que interagem directamente com o sistema), também conhecidos segundo alguns autores como a população alvo. Os utilizadores de um sistema são normalmente divididos em dois grupos:

Utilizadores directos – utilizadores que trabalham directamente e de forma assídua com o sistema.

Utilizadores indirectos – utilizadores que utilizam o sistema mas não tão assiduamente como os anteriores, como por exemplo gerentes, supervisores, responsáveis pelo sistema, etc.

Para a nossa análise vamos tomar em consideração os utilizadores directos, uma vez que são estes que irão utilizar o sistema de forma assídua. Logo, torna-se necessário identificar não só o tipo de utilizadores, como as suas características. Para

¹¹ UML: “*Unified Modelling Language*” utilizada no desenvolvimento de sistemas. Abrange sempre qualquer característica de um sistema nos seus diagramas, sendo aplicada nas várias fases de desenvolvimento (desde a especificação da Análise dos requisitos até à finalização, com a fase de testes). O objectivo é modelar o sistema em termos de diagramas orientados a objectos de forma a criar consistência, facilitar a comunicação com outras aplicações, simplicidade em ser actualizado e compreensível.

que estas características sejam conhecidas é necessário recorrer a um estudo estatístico (Modelo I em anexo) que nos dará informação sobre: categoria, faixa etária, experiência profissional, experiência de trabalho que executa, experiência na utilização de tecnologia, experiência em sistemas similares, frequência de utilização do sistema e motivação com que executa as tarefas. A informação recolhida com este modelo permite, por exemplo, saber se a população é jovem ou não, informação esta relevante para a construção da interface, até porque, normalmente, populações jovens necessitam de um nível de exigência e aparência diferente de uma população menos jovem. O mesmo se verifica entre utilizadores frequentes e utilizadores que utilizam raramente o sistema. Esta análise permite também saber a motivação e a o grau de aceitação do nosso sistema, uma vez que pessoas motivadas e com expectativas exigem mais da nosso sistema do que pessoas desmotivadas e sem perspectivas futuras. O conhecimento destas e outras características são pertinentes para o desenvolvimento da interface do nosso sistema e devem ser relacionadas e tomadas em consideração quando estamos a desenvolver a interface.

8.2. Análise do trabalho

Se a análise do utilizador é uma fase crucial do nosso modelo, a análise do trabalho também o é. Nunca poderemos construir ou modificar um sistema automatizado, por mais simples que ele seja, sem conhecer primeiro o trabalho que ele deve realizar. Tal como refere Walter Cybis, do Laboratório de *Usabilidade* de Informática, “ a análise do trabalho para a especificação de um novo sistema é justificada pela abordagem – conhecer para modificar.” Além do conhecimento real do trabalho executado pelos utilizadores, o analista fica com uma panorâmica geral dos problemas e complexidades que irá encontrar. A análise do trabalho torna-se também importante uma vez que poderá servir de ponto de partida para trabalhos futuros em sistemas semelhantes, bem como para testar novas alternativas e funcionalidades do sistema

Desta forma, a análise do trabalho ou AT, segundo Cybis, prevê actividades de colecta de informação, entrevistas e observação de duas etapas análise: análise de tarefas e análise de actividades. Segundo o mesmo autor, para encerrar o processo de

análise, tem que ser feita uma terceira análise: análise do ambiente. Esta análise permite retirar informações importantes para a construção da nossa interface, como por exemplo o tipo de iluminação e ruído existentes, o tipo de tecnologia disponível, o tipo de ferramentas existentes e que servirão de apoio à nossa tecnologia, etc.

8.3. Tarefa e Actividade

Antes de se fazer a descrição da análise de tarefas e da análise de actividades é importante distinguir actividade de tarefa. Desta forma, considera-se tarefa como sendo o trabalho que as pessoas devem realizar seguindo ordens de uma chefia, de um colega ou de ele mesmo. Por actividade entende-se como sendo a forma como o trabalho é efectivamente realizado ou, por outro lado, o modo como a pessoa ou conjunto de pessoas realizam a tarefa.

8.3.1. Análise de Tarefas

Existem muitas e variadas formas de realizar a análise de tarefas. Mas, e tal como diz Luciano Roque(2001), todas estas formas visam a análise e documentação da informação relevante sobre as tarefas desempenhadas pelos potenciais utilizadores do artefacto a desenvolver e que poderão ser alvo de redefinição pela introdução do artefacto influenciar as opções a tomar para o seu design.

O conteúdo da análise de tarefas deve ser descrito em termos de metas, objectivos, procedimentos e regras, tendo o analista de recolher a informação através de entrevistas, da análise de circulação dos documentos e da informação, bem como ser feita uma análise da organização do trabalho, das ligações deste com os serviços e das características dos postos de trabalho.

O analista, ao fazer a análise de tarefas, deve evitar a descrição do funcionamento esperado do sistema, centrando-se apenas na descrição de tarefas que constituem a envolvente de utilização desse mesmo sistema. Desta forma, deve centrar a sua atenção nas actividades dos utilizadores (Licínio Roque, 2001/2002).

8.3.2. Análise de Actividades

A análise das actividades tem como objectivo estudar a forma como o sistema é trabalhado ou, dito de outro modo, a forma como as tarefas são efectuadas pelos trabalhadores. Este tipo de análise, tal como a análise de tarefas, deve ser feita “*in loco*”, utilizando sessões de trabalho reais. Quando tal não for possível devem ser criados simulacros, onde se possa aproximar tanto quanto possível da realidade e tirar daí a maior quantidade de informação possível. As observações a tirar das interacções criadas entre os operadores do sistema devem ser organizadas de forma a encontrar três factores distintos:

- Situações de normalidade;
- Situações de aprendizagem;
- Situações de incidentes.

A análise deste tipo de situações irá revelar aspectos importantes para a construção da nossa interface e do nosso sistema. Aspectos como as operações realizadas, o seu encadeamento, a sua frequência e o aparecimento de incidentes ou situações fora da normalidade devem ser recolhidas e tratadas nesta fase. Também deve ser recolhido informação referente à circulação e utilização da informação para que se encontrem lacunas ou possíveis más utilizações deste recurso. Devem ser eliminadas informações inúteis que poderão induzir em erro o processo e ser criados mecanismos para que a informação necessária chegue correctamente. Também nesta fase temos a oportunidade de retirar informação relativa às denominações e termos utilizados pelos utilizadores indo de encontro com a heurística 1, falada no capítulo 6 desta tese.

Tal como diz Walter Cybis, do Laboratório de *Usabilidade* Informática, as diferenças existentes entre o que é previsto e o que se verifica na prática, tanto no que se refere ao funcionamento como à operação do sistema, está certamente na origem de erros e incidentes ou de custosas estratégias de acomodação (jeitinho) desenvolvidas pelos operadores. (Cybis: Maio, 2003, p.13).

8.3.3. Modelação

Existem diversas técnicas para a modelação de tarefas. A *Hierarchical Task Analysis* (HTA) é uma das formas mais conhecidas de análise da tarefa. A estrutura básica da Análise Hierárquica de Tarefas é uma representação gráfica da decomposição de uma determinada tarefa em várias subtarefas. As informações sobre a tarefa são obtidas por meio de conversas com os utilizadores e descrição do trabalho a realizar. O objectivo principal é descrever a tarefa em termos de hierarquia de operações e planos.

O *Méthode Analytique de Description des tâches* (MAD), sugerido por Scapin (1989) e (1990), é uma variação de HTA. Ele permite decompor as tarefas em subtarefas. O conceito de tarefa é representado por um objecto genérico chamado de objeto-tarefa que é definido por um conjunto de elementos. Além disso, cada objeto-tarefa (folha da árvore hierárquica da decomposição) é identificado por um nome e um número. Assim, cada tarefa será representada pelas seguintes características:

1. Identificação da tarefa (número e nome);
2. Elementos da tarefa (finalidade, estágio inicial, pré-condições, corpo da tarefa, pós-condições, estágio final);
3. Atributos da tarefa.

Essas técnicas são geralmente usadas no desenvolvimento de metáforas de interfaces de sistemas em geral e são particularmente importantes na adequação da interface às tarefas dos utilizadores (Alfredo, 2003).

Normalmente, nas organizações as pessoas estão envolvidas com um ou mais processos de trabalho. Nesses processos as tarefas que são executadas estão interrelacionadas entre si, onde um ou mais actores estão envolvidos. O resultado da execução das tarefas origina o produto ou serviço pretendido. Vamos considerar, para a nossa Análise Hierárquica de Tarefas, que cada tarefa é uma parte do processo executada do início ao fim por um mesmo actor e que esta pode ser dividida em actividades, ou subtarefas, onde uma actividade é caracterizada por um actor que a

executa, pelos objectos utilizados para a sua execução, pela duração e pela frequência com que esta é executada.

As tarefas podem ser agrupadas em planos (Alfredo, 2003), que podem ser:

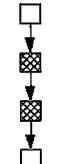
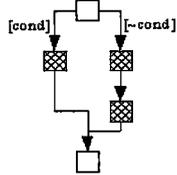
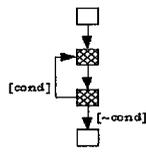
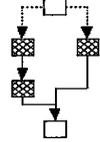
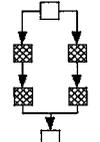
<p>1. Sequenciais: as actividades são executadas umas após as outras;</p>	
<p>2. Condicionais: certas actividades podem ou não ser executadas sob certas condições;</p>	
<p>3. Cíclicos: um grupo de actividades deve ser executada repetidamente sob uma certa condição;</p>	
<p>4. Concorrentes: actividades que podem ser executadas ao mesmo tempo;</p>	
<p>5. Alternativas: selecciona um grupo de tarefas para executar;</p>	

Tabela 3 – Tarefas como planos. Adaptado de Alfredo, 2003.

Desta forma, a HTA compreende uma lista de tarefas e actividades que são agrupadas em tarefas de nível mais elevado e decompostas em tarefas ou acções de nível mais baixo, de forma a descrever as actividades dos utilizadores em frases simples mas completas onde se podem recolher elementos como:

- Quem faz o quê?
- Quando o Faz?
- Que objectos intervêm na acção?
- Que outros actores estão envolvidos?
- Com que efeitos.

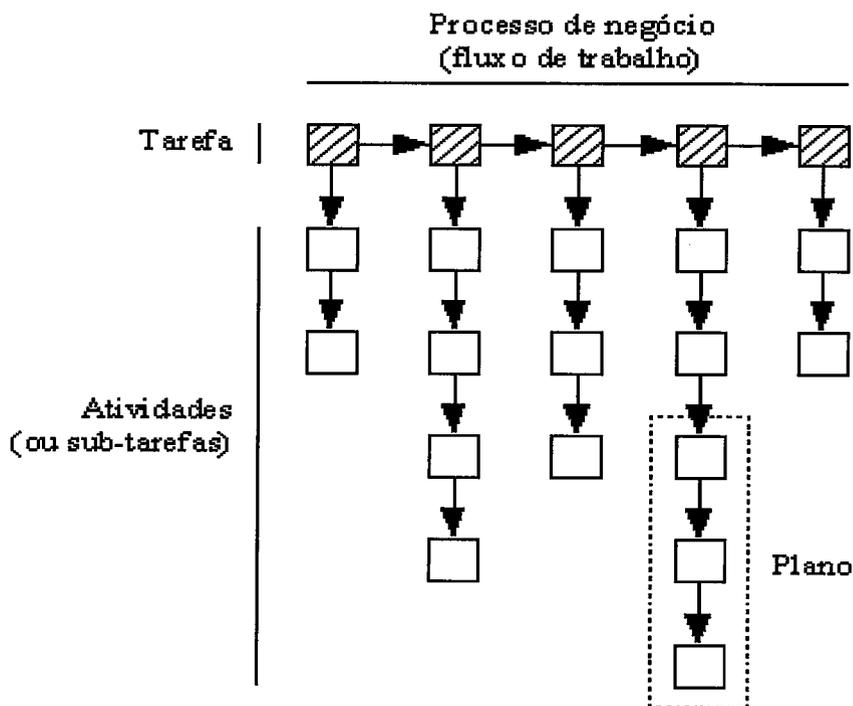


Figura 5- Processo de negócio. Adaptado de Alfredo, 2003.

Segundo Licínio Roque (2001), esta subdivisão pode prolongar-se por sub-níveis, enquanto se julgar relevante para a recolha de informação ou para o design a realizar. Normalmente, paramos quando atingimos o nível das operações elementares, em que já descreveríamos as interações com o artefacto.

Vejamos o exemplo da tarefa – “limpar uma casa” – segundo a abordagem HTA.

A tarefa mais geral toma o número 0.	0: Como limpar uma casa
Agora vamos decompor a tarefa em actividades mais simples.	0: Como limpar uma casa 1: ir buscar o aspirador 2: colocar as peças do aspirador 3: limpar as divisões 4: limpar o aspirador e retirar as peças
Nalguns casos as subtarefas (actividades) ainda podem ser decompostas.	0: Como limpar uma casa 1: ir buscar o aspirador 2: colocar as peças do aspirador 3: limpar as divisões 3.1: limpar a sala 3.2: limpar a cozinha 3.3: limpar os quartos 4: limpar o aspirador e retirar as peças

Definir planos de execução	<p>0: Como limpar uma casa 1: obter o aspirador de pó 2: colocar as peças do aspirador 3: limpar os aposentos 3.1: limpar a sala 3.2: limpar a cozinha 3.3: limpar os quartos 4: limpar o aspirador e retirar as peças</p> <p>Plano 0: 1-2-3-4 Quando o aspirador estiver cheio, faça 4</p> <p>Plano 1: faça 3.2 todo dia Faça 3.3 três vezes por semana Faça 3.1 quando houver visitas</p>
----------------------------	--

Tabela 4 - Exemplo segundo a abordagem HTA

Como referimos anteriormente, existem várias formas de modelação de tarefas. Nesta tese propõem-se esta ferramenta de modelação. Esta estrutura será uma preciosa ajuda para o processo de síntese que iremos desenvolver no capítulo seguinte.

Muitos projectos de desenvolvimento de sistemas começam com uma fase de Análise, tal como nos propomos, em que um grande número de informações é transmitido aos que desenvolvem sistemas por parte de especialistas em aplicação. Contudo, depois de algum tempo, a confusão instala-se devido à quantidade de informação que o analista tem de tratar. Neste sentido, é fundamental que a informação que chega ao analista seja de acordo com as necessidades reais dos sistemas, onde o envolvimento dos trabalhadores é fundamental e o recurso a ferramentas da moderna engenharia de software e ferramentas automatizadas no computador exercem um impacto favorável para a compreensão e desenvolvimento do nosso sistema computacional. Desta forma, a modelação da informação, tal como dizem Sally Shlaer e Stephen Mellor, providencia um remédio poderoso e específico contra a origem de falhas do projecto, uma vez que o modelo representa o mundo real em estudo, assim como identifica, classifica e resume o que está no problema e organiza a informação numa estrutura formal. Segundo os mesmos autores, a modelação é uma ferramenta pensante usada para ajudar na formalização do conhecimento (Shlaer / Mellor, 1990, p.9).

8.4. Análise do Ambiente

Para qualquer analista de sistemas, a tarefa mais difícil na especificação de um sistema é, muitas vezes, a determinação do que faz parte do sistema e o que não faz parte. Qualquer sistema que se desenvolva, não importa a sua grandeza e ambição, fará parte de um sistema ainda maior (Yordon, 1990, p.409). Depois de se encontrar a parte integrante do sistema, torna-se importante definir as interfaces entre o sistema e o ambiente que o rodeia. Necessitamos de definir que tipo de informações e interações são feitas pelo ambiente que o rodeia. Durante a visita à realidade do trabalho deverão ser recolhidos dados referentes ao contexto de trabalho, e o ambiente que nele é realizado. A atenção deve estar centrada em dados relativos ao tipo de iluminação existente, que ruído é gerado pelo processo, o estado da tecnologia existente, que tipo de ferramentas de apoio devem ser criadas, que tipo de utilizadores utilizam o sistema, que tipo de layout deve ser feito, etc.

A repercussão destes e outros factores sobre a síntese da interface será de vital importância para o sucesso do nosso sistema. Imaginemos que necessitamos de fazer o controlo de entradas e saídas de pessoas num dado local. Uma das soluções será a utilização de um aparelho de registo de entradas ou saídas sonoras. No entanto, o aparelho será fortemente condicionado pelo nível de ruído ambiente, ou seja, se o ruído do processo for muito elevado teremos de procurar outra solução. Este exemplo pode parecer um pouco simplista, mas penso que ilustra bem a necessidade deste tipo de análise.

8.5. Relatório de Análise

No final da Análise deve ser feito um relatório que transcreva a descrição do utilizador e do trabalho por este é realizado. Este relatório deve conter várias informações, tais como: a análise do utilizador a sua identificação e caracterização, a análise do trabalho, nomeadamente a identificação de tarefas e actividade e sua caracterização, e ainda, como não poderia deixar de ser, a análise do ambiente e da tecnologia disponível e das possibilidades e restrições das tecnologias a utilizar. O relatório deve também conter um diagnóstico das situações problemáticas e as

possíveis soluções baseadas em recomendações para a concepção da interface. Quanto à análise tecnológica, esta deve focar o ambiente tecnológico onde a síntese da interface será realizada. Desta forma, deve conter o estilo geral da interface, as bibliotecas e recursos a serem implementados e as ferramentas que irão servir de apoio à implementação. Este tipo de Análise irá facilitar a próxima fase de síntese e será uma preciosa ajuda para a criação de interfaces com *Usabilidade*.

Parte 6

**Construção (modificação) da Interface
Processo de Síntese**

“Deve ser observada uma simplicidade espartana. Nada poderá ser feito apenas porque contribui para a beleza, conveniência, conforto ou prestígio.”

Yourdon, 1999, p.465

Síntese

Depois de ter sido feita a análise do utilizador, do trabalho que este executa e do ambiente que o rodeia e influencia, torna-se agora necessário sintetizar essas mesmas análises para se dar início à construção (modificação) da nossa interface.

Propomos neste capítulo desenvolver o processo de síntese tomando como base o trabalho desenvolvido no capítulo anterior. O processo aqui descrito é fruto de uma sequência de etapas que não podemos deixar de fazer referência. A primeira etapa diz respeito à especificação do contexto de utilização, ou seja vamos encontrar todas as pessoas que irão ser envolvidas no projecto. Estas pessoas devem ser esclarecidas sobre o contexto de utilização da nova interface.

A norma 9241-11 define como contexto de utilização os utilizadores, as tarefas, o equipamento (hardware, software e documentos) e os ambientes físicos e sociais susceptíveis de influenciar a *Usabilidade* de um produto dentro de um sistema de trabalho. O documento de especificação do contexto de utilização é o ponto de partida para o projecto da futura interface com o utilizador, fazendo com que o primeiro protótipo da nossa interface não apareça do nada mas por outro lado seja sustentado e possa evoluir de acordo com um contexto e uma realidade existentes. Esta especificação irá ainda criar condições para a realização de testes de Usabilidade e permitem ao analista definir os estilos e componentes a utilizar, bem como definir os processos a incluir na interface.

A segunda etapa a que iremos fazer referência é a definição de uma nova estrutura de trabalho. Segundo a norma atrás descrita, as tarefas fazem parte da especificação do contexto de utilização. Desta forma, e para que possamos especificar a estrutura da futura tarefa, é necessário realizar um processo de reengenharia¹² do trabalho (Wlter Cybis, Maio 2003, p.92). Neste processo de reengenharia as decisões sobre quem faz o quê devem ser baseadas nas características cognitivas dos dois agentes envolvidos no sistema: o homem e o computador.

¹² Reengenharia: processo de melhoria organizacional que dê prioridade as acções críticas, ou seja aquelas que podem afectar mais o desempenho da organização como um todo.

Outro aspecto importante refere-se ao controle das actividades e ao grau de automatização que pode ser conferido à execução da tarefa interactiva. Como se sabe, o ser humano e a máquina têm capacidades diferentes. Enquanto a máquina consegue velocidades e precisões elevadas de processamento de operações lógicas e algorítmicas, assim como uma enorme capacidade de armazenamento, o ser humano, por outro lado, consegue fazer o reconhecimento de padrões de comportamentos e efectuar juízos de valor sobre problemas encontrados, fazendo associações com experiências passadas e similares, e possui também uma capacidade versátil para enfrentar soluções inéditas. Nesse sentido, é importante sublinhar a relação inversamente proporcional entre o ser humano e a máquina e tentar encontrar o equilíbrio entre um e o outro.

Como ponto de partida para encontrar o equilíbrio vamos utilizar o relatório de Análise que definimos no capítulo anterior. Este relatório comporta informação não só dos utilizadores como da própria estrutura do trabalho actual e dos problemas encontrados. Estas informações são uma preciosa ajuda para o processo de definição da nova estrutura de trabalho (reengenharia), onde mais uma vez o envolvimento e a participação dos utilizadores têm que ser uma constante. Neste processo importa definir as tarefas problemáticas e encontrar várias maneiras de resolução. Para tal, é necessário intervir junto dos utilizadores e recolher essa mesma informação. Encontradas essas tarefas, torna-se importante definir prioridades, sendo estas definidas segundo dois parâmetros: o primeiro diz respeito à viabilidade técnica, e o segundo à vontade e desejos dos utilizadores. Depois de encontradas, vamos definir e representar a nova estrutura das tarefas, que serão representativas não só das necessidades e desejos dos utilizadores, mas também uma forma viável e eficaz de resolução do problema por parte da componente técnica. Torna-se então necessário redefinir as tarefas de forma a interligar funções em função dos utilizadores e da tecnologia. O processo acima referido como reengenharia pode ser descrito da seguinte forma:

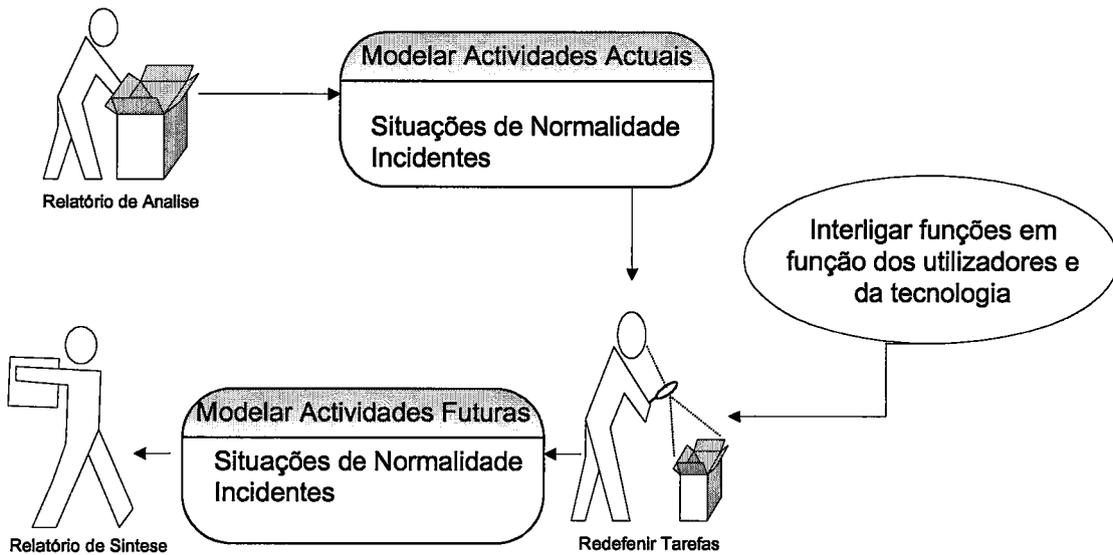


Figura 6 - Etapa de reengenharia do trabalho do processo de Síntese

Esta etapa é crucial para o bom funcionamento do nosso sistema. O relatório de Análise fornece-nos informação relativamente aos utilizadores e as tarefas que estes executam. Importa agora pegar nesse trabalho e fazer a modelação dessas actividades. A modelação das actividades actuais, ou seja, as actividades que são realizadas normalmente pelos utilizadores, já foi realizada na Análise através da *Modelação de Tarefas (HTA)*. Pegando nesse trabalho, evoluímos para uma modelação orientada a objectos, para que possamos interligar funções em função dos utilizadores e da tecnologia, no sentido de evoluir para a modelação das actividades futuras (fig 6).

8.6. Síntese utilizando UML

O grande problema do desenvolvimento de sistemas utilizando a orientação a objectos na fase de Análise é a de não existir uma notação padronizada e realmente eficaz que abranja qualquer tipo de aplicação a desenvolver. Cada simbologia existente tem na sua génese conceitos próprios e utiliza gráficos e nomenclaturas próprias, resultando daí uma grande confusão para aqueles que necessitam de utilizar a programação orientada a objectos no desenvolvimento dos sistemas.

A UML é muito mais que a padronização de uma notação. É também o desenvolvimento de novos conceitos. Desta forma, a compreensão da UML não é apenas a aprendizagem de uma simbologia, mas também significa a aprendizagem da modelação orientada a objectos. Desta forma, vamos considerar como necessária a modelação orientada a objectos para a criação do nosso sistema de informação.

Tal como defendem, entre outros, Coad, Yourdon e Pressman, a orientação a objectos é uma tecnologia para a produção de modelos que especificam o domínio do problema de um sistema. Segundo eles, quando os sistemas são construídos nesta base, estes são flexíveis a mudanças, possuem estruturas bem conhecidas e proporcionam oportunidades de criar e implementar componentes totalmente reutilizáveis. A orientação a objectos requer um método que integre o processo de desenvolvimento e a linguagem de modelação com a construção de técnicas e ferramentas adequadas.

8.6.1. Utilização da UML

A unificação dos métodos e a criação de novos padrões é uma tentativa de padronizar a modelação orientada a objectos para que qualquer sistema possa ser modelado com consistência, facilitando a comunicação com outras aplicações, simplificando a utilização e facilitando a compreensão.

A UML é usada no desenvolvimento dos mais diversos tipos de sistemas, abrange sempre qualquer característica de um sistema em algum dos seus diagramas e pode ser aplicada em qualquer fase do desenvolvimento de um sistema, desde a fase de análise até à fase de testes. O objectivo do UML é descrever um sistema utilizando diagramas orientados a objectos. Para a nossa fase de síntese iremos utilizar os diagramas de Modelação do Comportamento¹³ do UML. Estes diagramas dividem-se em três tipos diferentes: (1) Diagramas de Estados, (2) Diagramas de Actividades e (3) Diagramas de Interação, sendo os últimos divididos ainda em Diagramas de Sequência e Diagramas de Colaboração. (Booch, 1994, 1998); (Silva, 2001).

¹³ Diagrama de Modelação do Comportamento – diagramas que permitem especificar a dinâmica ou comportamento de um sistema segundo a abordagem orientada a objectos

Diagramas de Estados – Exibem os estados do sistema. Estes diagramas são formados por estados, transições eventos e actividades. Permitem ter uma visão dinâmica do sistema. (Booch,1998).

Diagramas de Actividades – É um tipo especial de diagrama de estados, exibindo o fluxo de uma actividade para outra no sistema. (Booch,1998).

Diagramas de Interação – Exibem uma interacção, sendo compostos por um conjunto de objectos e dos seus relacionamentos, incluindo as mensagens que podem ser trocadas entre eles. Estes diagramas dão uma visão dinâmica do sistema. (Booch,1998)

Diagrama de Sequência – é um diagrama de interacção cuja ênfase está na ordenação temporal das mensagens. (Booch,1998)

Diagramas de Colaboração – é um diagrama de interacção cuja ênfase está na organização estrutural dos objectos que enviam e recebem mensagens. (Booch,1998)

Depois de modeladas as actividades actuais recorrendo aos diagramas mencionados é agora necessário fazer a interligação de funções. O que possa parecer funcional para o utilizador pode não o ser para a tecnologia. Neste sentido, torna-se necessário rever a modelação e refaze-la de forma a satisfazer o desejo dos utilizadores e a viabilidade técnica. A última etapa do processo de síntese tem a ver com o projecto da interface.

8.7. Projecto da Interface

O projecto da interface de um sistema afecta mais que qualquer outra coisa o modo como o utilizador irá usufruir de um produto. Desta forma, é importante promover um bom desenho da interface com o utilizador com consistência visual e funcional.

Tal como diz Cybis (2003), as actividades de projecto de interface têm como objectivo definir formas de apoiar a realização da futura estrutura de trabalho no contexto de uso defendido para o sistema do sistema, relacionando com o tipo de utilizadores as tarefas e o seu ambiente técnico, organizacional e social.

Tal como referimos na primeira parte desta tese, as melhorias trazidas pelas plataformas Windows e pela evolução das linguagens de programação proporcionam uma evolução de uma interface centrada na aplicação para uma interface centrada na informação. Desta forma, é necessário repensar a interface do software, dos componentes básicos, das suas operações e das propriedades que se lhes aplicam, para que os utilizadores interajam com a informação sem se preocuparem com a aplicação. Para se realizar o projecto da interface não basta ter uma boa interface gráfica, até porque, tal como vimos, existem outros factores que têm de ser tidos em consideração. O projecto da aplicação, a Análise do trabalho do ambiente e do próprio utilizador, a criação de protótipos e avaliação da facilidade de utilização são factores importantíssimos para um bom projecto da interface.

Uma interface bem projectada deve ser baseado em princípios, num processo de desenvolvimento centrado no utilizador e nas tarefas que estes executam. As tarefas que estes executam estão modeladas e o processo de desenho centrado nos utilizadores também já foi descrito na segunda parte desta tese.

8.7.1. Design de Interfaces

O termo Design é de origem Inglesa e, no seu sentido mais amplo, significa criar, conceber, dar forma a algo. É a parte constituinte da criação ou redesenho de um determinado objecto ou produto, tendo em conta factores sociais, económicos e estéticos reflectidos no projecto ou no desenvolvimento deste.

Nas duas partes anteriores foram definidos questões relativas à organização da estrutura da informação, especificação do contexto da utilização e novas formas realização do trabalho. Estas fases são bastante importantes, pois estabelecem a funcionalidade e o trabalho a ser desenvolvido. Porém, ainda não existem imagens visuais do novo sistema. Por outras palavras, o motor do nosso carro está, nesta fase, pronto, mas ninguém irá comprar o automóvel só pela qualidade do motor. Ele tem de

ser agradável também à vista e bem aceite pelos utilizadores. Tal como diz Figueiredo (2004, p.49), o design determina a forma, e a forma deve seguir a função. A adequação dos objectos à sua função de uma forma esteticamente agradável é a base do design. Segundo o autor referido, a função primordial do designer é estabelecer hierarquias visuais. Desta forma, a importância do designer num projecto de interfaces é importantíssima e assume cada vez mais maior proporção, visto a interface ser composta na sua grande parte de elementos visuais. A partir deste ponto, nasce a necessidade de utilizar fundamentos sobre factores humanos e de design na elaboração de interfaces homem máquina. Um desses factores relaciona-se com os *elementos visuais*. Segundo Cybis (1997), os elementos visuais mais significativos para as questões ergonómicas são: a cor, a linha, a fonte (a que iremos chamar de tipo de letra nesta tese) e o arranjo (a que iremos chamar de composição nesta tese).

A Cor e a sua utilização em IHM

Segundo Marcus (1987) a cor é um factor relevante na comunicação, já que ela possui capacidades de influenciar as pessoas, proporcionando reacções nos sentimentos, nas emoções e no intelecto do indivíduo. Desta forma, a utilização das cores pode fazer com que se alcancem propósitos específicos. As cores, quer estejam presentes em imagens, quer em tabelas, ou mesmo em textos, são um dos elementos essenciais para o sucesso de uma composição. Consoante a cor utilizada, assim se estabelece o ambiente apropriado ao tipo de interface a ser criado.

Cada cor possui o seu significado (Figueiredo, 2004). Muitas vezes, o significado varia consoante a cultura. No quadro seguinte estão expostos os significados das principais cores. Consoante os impactos causados, as cores devem ser utilizadas de forma apropriada na construção da interface, pois o uso incorrecto pode proporcionar resultados indesejáveis.

Cor	Significado	Observação
Amarelo	Optimismo, esperança, filosofia, desonestidade, covardia, traição	Cor sagrada para os Hindus.
Azul	Confiança, conservadorismo, segurança, tecnologia, limpeza, ordem	Usado em empresas de seguros, bancos, simboliza confiança.
Branco	Pureza, limpeza, precisão, inocência, esterilidade, morte	Associado ao casamento em culturas Ocidentais, e à morte nas Orientais.
Castanho	Terra, Confiança, conforto, durabilidade	Pesar, melancolia, resistência, vigor.
Cinzentos	Intelecto, futurismo, modéstia, tristeza, decadência	É a cor que o olho reconhece mais facilmente.
Laranja	Energia, equilíbrio, força, luminosidade, dureza, prazer, euforia, tentação, advertência	Simboliza o flamejar do fogo. Estimula o apetite.
Preto	Poder, sensualidade, morte, tristeza, mistério, medo, elegância	É expressivo e angustiante ao mesmo tempo. É alegre quando combinado com certas cores. Por vezes tem conotação de Nobreza.
Púrpura	Espiritualidade, mistério, realeza, transformação, crueldade, arrogância.	Simboliza a dignidade real.
Verde	Natureza, saúde, boa sorte, inveja, renovação	Cor repousante. Não funciona bem em todo mundo. Na China e na França, embalagens verdes não são bem vistas. Tem sucesso no Médio Oriente.
Vermelho	Poder, energia, calor, paixão, amor, agressão, perigo, sangue, guerra	Altera o seu significado em presença de outras cores. Simboliza aproximação, encontro.

Tabela 5 - As diferentes cores e seus significados, adaptado de Bruno Figueiredo, 2004

Tal como diz Barros (2003), a boa utilização das cores em interfaces homem máquina pode aumentar significativamente o tempo de resposta do utilizador. Segundo a autora, existem algumas recomendações que devem ser seguidas na utilização das cores:

- Seleccionar correctamente as cores pois elas influenciam na legibilidade da interface;
- Agrupar com a mesma cor elementos da tela que possuam a mesma relação;

- Não utilizar mais de cinco cores diferentes. Sabe-se que, actualmente, a tecnologia tem a seu favor placas de alta resolução. No entanto, a mente humana só é capaz de discriminar, em média, 7,5 milhões de cores.
- Avaliar o meio cultural e profissional na codificação cromática.

Cybis (1997) recomenda a utilização de cores saturadas para fundos e contornos de objectos (com a excepção do verde e vermelho para contornar os objectos). Para apresentação tipográfica de primeiro plano deve-se evitar a utilização do azul, do magenta e do rosa para os caracteres apresentados em primeiro plano.

As Linhas e a sua utilização em IHM

Tal como as cores, as linhas e as formas têm os seus significados próprios (conforme mostra a tabela 6). A cada forma e a cada linha corresponde um estado de espírito muito próprio. Quando falamos de linhas e formas, normalmente associamos à noção de movimento e sentido (direcção). Segundo Barros (2003), a linha é resultado da acção de forças sobre um ponto, em que se pode diferenciar duas possibilidades fundamentais. A primeira é a actuação de uma força que origina uma linha recta, e a segunda é a actuação de duas forças que pode originar a criação de linhas quebradas ou curvas. Grande parte dos objectos criados de interacção tem nos seus limites linhas e formas que são bastante significativas na grandeza e leveza destes objectos. Desta forma, e uma vez que o projecto de interfaces gráficos é uma composição, sugere-se a utilização de linhas de acordo com o significado exposto na tabela (6).

O tipo de letra e a sua utilização em IHM

Outro dos elementos visuais a ter em consideração tem a ver com o tipo de letra. Esta deve ser legível, agradável e de fácil compreensão. Com a evolução das artes gráficas surgiram novas preocupações em relação à disposição de caracteres. Segundo Cybis (1997), as principais características pertinentes à percepção de fontes

utilizadas em textos e rótulos das telas dos computadores são a *serifa*¹⁴ e o espaçamento entre caracteres.

Linhas				Formas		
						
Vertical	Horizontal	Diagonal	Curva	Círculo	Quadrado	Triângulo
Masculino	Tranquilidade	Movimento	Feminino	Protecção	Tranquilidade	Acção
Homem Erecto	Continuação	Desarmonia	Abraço	Perfeição	Esmero	Conflito
Dignidade	Estabilidade	Crescimento	Não agride	Simetria	Estabilidade	Tensão
Aspirações	Paz	Direcção	Continuidade	Paz	Escuro	Equilíbrio
Estabilidade	Calma		Sensualidade	Infinito	Honestidade	Força
Firmeza	Vagar		Graciosidade	Formalidade	Confiança	Vida
Predomínio			Delicadeza	Movimento	Preto	Estabilidade
Grandeza				Azul	Vermelho	Amarelo
Interrupção				Concentração	Firmeza	Elevação
Segurança				Infinidade		
Sofisticação						

Tabela 6 - Significado das linhas e formas, adaptado de Barros(2003), p.58

Cybis (1997) recomenda a utilização de fontes sem serifa em títulos e rótulos curtos. Já as fontes com serifa são recomendadas para a utilização em textos, para que os utilizadores reconheçam de uma forma rápida os caracteres. Quanto ao espaçamento entre caracteres, este deve ser dispostos de forma contínua e proporcional.

Os recursos, em termos de estilo, devem ser utilizados com cautela, para discriminar ou destacar elementos ou partes de um texto. Incluem-se nestes recursos as caixas, o negrito o itálico e o sublinhado. O tamanho dos caracteres deve ser apropriado. É recomendada a utilização do tamanho 18 a 24 para transparências e de 10 a 12 para textos normais. O tamanho também deve permanecer inalterado na mudança de fontes.

¹⁴ Serifa : fontes com serifa são caracterizadas por uma terminação saliente dos caracteres, por exemplo o “Times New Romam”. Sem serifa são percebidas com leveza, dificultam a leitura, por exemplo o “Arial”.

A composição e a sua utilização em IHM

O último elemento visual a que iremos fazer referência é a Composição. A composição corresponde à forma como os itens de informação se encontram dispersos na interface. A composição é também muitas vezes chamada de Arranjo ou Layout. Cybis (2003) recomenda que para se conseguir um arranjo adequado devemos:

- Definir um grid¹⁵ para o layout das telas. Deve ser defendido alinhamentos para os elementos conforme as linhas e colunas do grid.
- Também devem ser defendidas zonas de trabalho, agrupando os elementos interrelacionados. Deve ser criada uma estrutura a partir das relações entre os elementos, onde se coloca em evidência o que é mais importante no grupo. Esse destaque deve ser feito o mais colorido possível e o mais à esquerda. Depois devemos distribuir da esquerda para a direita em função da importância e da cronologia.
- Devemos dar equilíbrio às telas, distribuindo os elementos de forma balanceada. Evitar espaços vazios ou preenchidos em excesso com componentes.
- Manter consistência entre os layouts das telas.

8.7.2. Avaliação de Interfaces

Segundo Barros (2003), a avaliação de interfaces é um passo importante do processo de design. É através da avaliação que se consegue estimar o sucesso ou insucesso da solução que o designer está a propor, tanto em termos de funcionalidade como de interação. Tal como refere Nielsen (1993), quanto mais bem informado estiver o designer de interfaces sobre os utilizadores melhor será a qualidade dos seus produtos.

Neste sentido, a avaliação é um factor importantíssimo, não só no desenvolvimento do design da interface mas durante todo o processo de construção do sistema. Este tema da avaliação irá ser tratado mais em pormenor na parte 7 da nossa tese, onde irão ser demonstrados métodos e técnicas a serem seguidos para a avaliação da interface / sistema.

¹⁵ Grid: grelha que serve de suporte à construção do *layout das telas*.

8.7.3. Modelo para o design de Interfaces Homem Máquina

O modelo de design aqui apresentado é fruto do estudo de vários modelos que servem de auxílio à construção e design de interfaces. Este modelo é fortemente influenciado pela Avaliação Heurística descrita na 3ª parte desta tese. Mas também serviram de referência modelos tradicionais, tais como o modelo em *Cascata* e em *Espiral*, o *SSM (Soft System Methodology)*, o *OSTA (Open Systems Task Analysis)* e o *Modelo Estrela*, entre outros. Iremos agora apresentar um breve resumo dos métodos que influenciaram a escolha no desenvolvimento do modelo proposto.

Nos referidos métodos de desenvolvimento clássicos de software, os engenheiros de software utilizam uma abordagem *top-down* baseada na decomposição funcional. Este método, mais conhecido por modelo em *Cascata (Royce, 1970)*, propõe uma abordagem sistemática, linear e sequencial ao desenvolvimento de software. Este método inicia-se com a análise do sistema e progride passando pela análise de requisitos projecto, implementação e testes. No entanto, e tal como diz Preece (1994), sistemas mais complexos necessitam de um desenvolvimento mais iterativo, dando origem ao ciclo de vida em *Espiral (Boehm, 1988)*, modelo evolutivo que, ao conjugar a natureza iterativa da prototipagem com a característica sistemática do modelo sequencial, fornece o potencial para o desenvolvimento expedito de versões incrementais do software (António Manuel, 2003, pa.101).

No *Modelo em Estrela (Hix, D; Hartson, H.R., 1993)*, contrariamente ao que acontecia nos métodos tradicionais, é contemplada a interacção entre o homem e a máquina e faz mesmo parte do ciclo de vida dos sistemas. Este método é muito parecido com o método em *Cascata*, incluindo duas novas etapas que não aparecem no modelo em cascata: a prototipação e a avaliação. Este método proposto por Hix pressupõe a avaliação contínua e interacção. A avaliação é uma peça fundamental neste método. Segundo ele, não é necessário especificar todos os requisitos antes de começarmos a trabalhar no projecto do sistema. Podemos iniciar com um protótipo rápido dos ecrãs do sistema e depois vamos incrementando esses mesmos ecrãs com novos requisitos detectados. De acordo com este modelo, o ciclo de vida de desenvolvimento pode ser iniciado em qualquer das fases indicadas pelas setas de

entrada na figura (7). Cada fase pode ser seguida por qualquer outra, conforme mostram as setas duplas entre estados. Este modelo tem como principal preocupação a participação constante dos utilizadores tanto na construção como na avaliação constantes feitas no término de cada etapa.

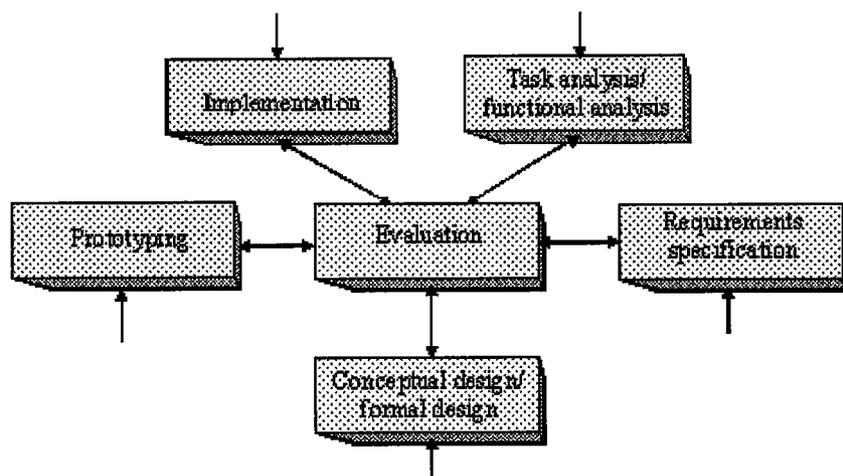


Figura 7 - *Star Model – Modelo Estrela* Adaptado de Hix, D. and Hartson, H.R.1993

No modelo de *Engenharia de Usabilidade*, os elementos básicos são o teste empírico e a construção de protótipos, combinados com o design iterativo. Numa primeira fase, é extremamente difícil construir uma interface totalmente adaptado ao utilizador, uma vez que é necessário testar e planejar alterações utilizando o design iterativo, o que torna muito difícil a mudança em sistemas já implementados. Desta forma, as mudanças são possíveis apenas no estado de construção do protótipo, pois mudanças num sistema já implementado são difíceis e onerosas para os programadores.

No Modelo baseado em *Avaliação Heurística* é utilizada a Avaliação Heurística descrita na 3ª. Parte desta tese, onde são feitos testes às interfaces e feitas as avaliações para que as interfaces se adequem as necessidades dos utilizadores. Às modificações introduzidas no nosso modelo, contrariamente ao modelo original onde as avaliações eram feitas apenas por uma equipa de avaliadores, aqui também incluímos uma avaliação feita pelos utilizadores.

Tal como diz Hartson (1998), a avaliação da interface é um passo importante para o processo de design, afinal é através dela que se consegue estimar o sucesso ou

insucesso das hipóteses do designer sobre a proposta de solução, tanto ao nível de funcionalidade como ao nível de interação.

Modelo de design em IHM

O modelo de design de interface foi dividido em três fases, tal como mostra a figura (8). Uma primeira fase que diz respeito ao conhecimento dos utilizadores e suas necessidades, uma segunda fase que trata do design do sistema, e uma terceira fase que é a implementação propriamente dita e o teste do produto.

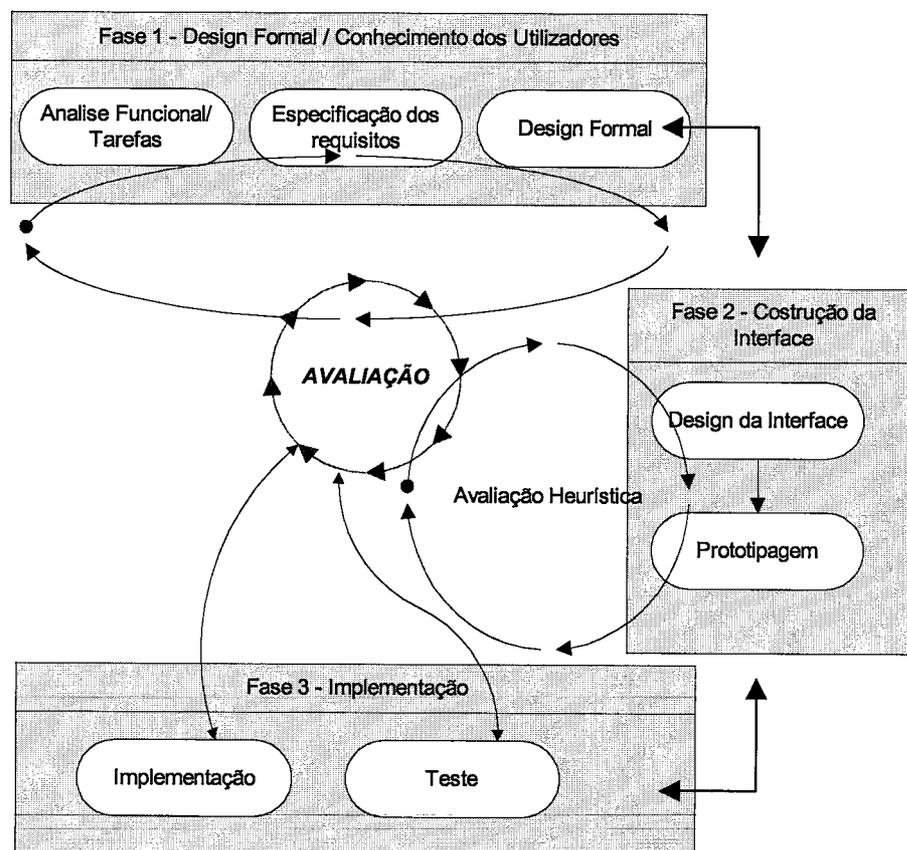


Figura 8 - Modelo de design IHM

Nesta nova abordagem, muito parecida com o modelo estrela atrás referido, a avaliação digamos que é a chave de todo o processo, já que ela está sempre presente e deve ser o motor de todo o desenvolvimento. A passagem de uma fase para a outra é sempre possível, mas é importante que tal não aconteça. O que é fundamental é que a participação dos utilizadores, na primeira e segunda fase seja muito forte, seja depois

minimizada na terceira fase, onde serão realizados os testes e avaliações pela equipa responsável pelas avaliações (os avaliadores).

Como podemos ver, foi dada grande importância ao processo de avaliação no nosso modelo. Tal como diz Hartson (1998), a avaliação da interface é um passo importante do processo de design, já que é através dela que se consegue estimar o sucesso ou insucesso da nossa interface sobre o sistema que pretendemos criar ou reformular. Antes de um novo produto de design industrial sair para o mercado, este deve ser testado quanto à sua eficácia e *Usabilidade*. Os métodos que devem ser utilizados são a avaliação heurística e os testes com utilizadores. As questões inerentes ao processo de avaliação irão ser tratadas na parte seguinte desta tese.

8.8. Resumo do Processo de Síntese

O processo de síntese fica completo com o projecto da interface que deve ser feito de acordo com as técnicas e modelos atrás referidos. Tal como descrevemos no início deste capítulo, o processo de síntese passa por três fases (fig 9). Uma primeira fase de especificação do contexto de utilização, onde iremos ter contacto com o tipo de utilizadores que iram utilizar o sistema, as tarefas que executam, o equipamento existente, os ambientes físicos e sociais que irão ser influenciados. Esta fase só é possível, como vimos, com o envolvimento de todos os intervenientes no processo. Este estudo é o ponto de partida para a fase de projecto da interface. No entanto, para que se possa passar para o projecto propriamente dito, é necessário especificar a estrutura das tarefas futuras, sendo necessário existir a Reengenharia do trabalho. Neste processo de reengenharia, as decisões sobre quem faz o quê devem ser baseadas nas características cognitivas dos dois agentes intervenientes no processo, o homem e a máquina.

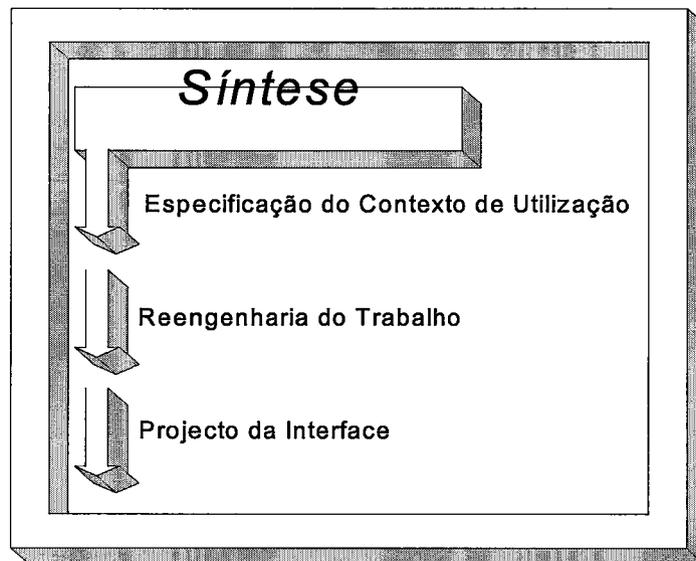


Figura 9 – Resumo do processo de Síntese

Por fim, podemos passar à fase de projecto da interface. Pretende-se com esta fase que sejam defendidas formas de implementar o futuro contexto de utilização do sistema, relacionando o tipo de utilizadores, as tarefas e o ambiente. Aspectos como, definição do estilo e funcionalidades do sistema, definição da estrutura das interações, cenários de utilização, diálogos e transições, tipo de apresentação e diálogo a implementar e objectos de interacção a criar, devem ser tidas em consideração. Depois desta fase damos por concluído o processo de síntese e podemos passar à implementação e testes do sistema.

A fase de implementação não irá ser tratada nesta tese, uma vez que depois de definidos as formas e os conteúdos a incluir no sistema bastará apenas à equipa de programadores programar de acordo com o que foi definido. A fase de testes, na medida em que tem interesse numa abordagem em interface homem máquina, irá ser tratada na fase seguinte, a que chamaremos Avaliação.

Parte 7

**Construção (modificação) da Interface
Processo de Avaliação**

9. Avaliação

Quando uma determinada empresa estabelece uma nova loja num determinado local estuda a receptividade dos seus produtos nesse novo mercado. Ela avalia a sua concorrência, a visibilidade do local e, se não vender os seus produtos, ela tende sempre a reorganizar a sua montra de forma mais atractiva para poder cativar possíveis clientes. Da mesma forma, a construção de um sistema, seja ele novo ou reformulado, deve ser avaliado quanto à eficácia da sua interface com os utilizadores através da interface que a aplicação oferece. O processo de avaliação de interfaces torna-se, desta forma, um factor importantíssimo na construção de sistemas próximos dos humanos (interfaces homem máquina). Podemos mesmo dizer que a avaliação direcciona e influencia todo o processo de design, criando um produto útil e utilizável.

O objectivo da avaliação serve, dito em poucas palavras, para avaliar a funcionalidade do sistema, avaliar o efeito do sistema perante o utilizador e, de uma forma geral, identificar problemas específicos do sistema. Como objectivos gerais, define-se que o processo de avaliação serve para: validar a eficácia da interacção homem máquina face à efectiva realização das tarefas por parte dos utilizadores; verificar a eficiência desta interacção face aos recursos empregues (tempo, quantidade de incidentes, passos desnecessários, procura da ajuda, etc.); obter indícios da satisfação ou insatisfação que ela possa trazer ao utilizador. (Cybis,2003).

Segundo Barros (2003), quando se inicia um processo de avaliação de interfaces é importante esclarecer que, mesmo com a junção de vários métodos e uma avaliação complexa e exaustiva, ainda existe um certo grau de incerteza. Desta forma, todo o ciclo de vida da interface deve conter um acompanhamento de avaliações periódicas. Assim, justifica-se a importância por nós dada ao processo de Avaliação no nosso modelo de construção de interfaces em IHM, onde a avaliação é o objectivo central no processo de desenvolvimento. Tal como vimos no capítulo anterior, a avaliação permanece em contacto permanente com as várias fases do processo de desenvolvimento.

9.1. Classificação das Avaliações

Segundo Preece (1994), as avaliações de uma interface podem ser classificadas de dois tipos: *formativas*, quando são feitas durante o processo de design da interface, ou *somativas*, feitas quando o produto já está terminado. Quanto à primeira opção, a avaliação formativa é bastante mais benéfica, porque permite identificar e resolver problemas de interacção antes que a aplicação tenha sido implementada, até porque, como se sabe, alterações feitas à posterior acarretam um conjunto de problemas muitas das vezes difíceis de resolver.

Por outro lado Barros (2003) classifica as avaliações em dois grupos. O primeiro diz respeito à avaliação segundo *inspecção de Usabilidade*. Este tipo de avaliação tem a particularidade de poder ser feita em qualquer fase do desenvolvimento do sistema, mas não inclui a presença dos utilizadores. O segundo grupo tem a ver com a avaliação recorrendo a *testes de Usabilidade*. Estes tipos de testes já foram explorados na 4ª parte desta tese.

Cybis (1994) classifica as Avaliações em três tipos: *prospectivas*, procurando a opinião do utilizador; as *preditivas/analíticas*, que visam prever erros na construção da interface sem a colaboração do utilizador; e, por fim, as *objectivas/empíricas*, que procuram problemas de interface com a observação directa do utilizador a interagir com a interface.

De uma forma geral, a grande diferença entre os vários tipos de classificação das Avaliações diz respeito à presença ou não dos utilizadores. Actualmente, são utilizadas várias técnicas de Avaliação de interfaces em IHM. Seguidamente iremos descrever as mais utilizadas.

9.2. Técnicas de Avaliação

Como vimos, Cybis classifica as avaliações em três tipos ou em três técnicas: técnicas Prospectivas, técnicas Preditivas e técnicas Objectivas. De uma forma geral, o objectivo de todas estas técnicas é o de encontrar problemas de *Usabilidade* em interfaces homem máquina e contribuir para a sua eliminação.

9.2.1. Técnicas prospectivas

Este tipo de técnica utiliza como recurso para a avaliação as entrevistas/questionários com os utilizadores de forma a avaliar a satisfação ou insatisfação em relação ao sistema. O recurso aos questionários é uma técnica muito útil na avaliação da interacção, pois através dela podemos retirar informações como perfil dos utilizadores, necessidades e sugestões. Deveremos ter especial cuidado na elaboração dos questionários, uma vez que é através desta ferramenta que o avaliador irá obter as informações necessárias para a construção da interface.

9.2.2. Técnicas preditivas

Este tipo de técnicas, também chamadas por Cybis (2003) de técnicas de diagnóstico, são técnicas que dispensam a participação dos utilizadores e baseiam-se em inspecções e verificações de versões intermédias ou acabadas de software interactivo feitas pelos gestores de projecto ou especialistas em *Usabilidade*. O autor classifica-as de três formas diferentes: avaliações Analíticas, avaliações Heurísticas e avaliações por *Checklists*.

- ***Avaliação Analítica***

Este tipo de avaliação deve ser realizada nas primeiras fases de desenvolvimento da interface e, normalmente, envolvem a decomposição hierárquica da estrutura das tarefas de forma a verificar as interacções propostas.

- ***Avaliação Heurística***

Este tipo de avaliação, abordada na parte 3 desta tese, representa um julgamento de valor sobre as qualidades ergonómicas das interfaces homem máquina. Este tipo de avaliação pode ser utilizada durante todo o ciclo de desenvolvimento do sistema. As regras da avaliação heurística conduzem à descoberta, invenção e resolução de problemas para a concepção de sistemas. A avaliação heurística é baseada num conjunto de regras que visam descrever prioridades comuns em interfaces utilizáveis. (ver parte 3 da tese).

- ***Avaliações por Checklists***

A inspecção ergonómica por *Checklists* é uma técnica que se baseia em listas de verificação através das quais os profissionais, não necessariamente especialistas em ergonomia, como por exemplo programadores e analistas, diagnosticam rapidamente problemas gerais e repetitivos das interfaces (Jeffries et al, 1991). As listas de verificação devem ser bem elaboradas, uma vez que esta técnica de avaliação, ao contrário das avaliações heurísticas, centra-se na qualidade das ferramentas e não nos avaliadores, sendo a ferramenta responsável pela identificação dos problemas de *Usabilidade* existentes na interface.

- ***Avaliação por Inspeções Cognitivas***

Esta técnica verifica os processos cognitivos do utilizador enquanto este interage com o sistema pela primeira vez. (Polson,1985). Pretende-se com esta avaliação verificar se o software proporciona ao utilizador facilidade de utilização. O sucesso desta avaliação baseia-se, fundamentalmente, no conhecimento do avaliador sobre o processo cognitivo do utilizador enquanto este interage com a interface. Para que esta avaliação seja bem efectuada, o avaliador deve ter um conhecimento prévio do utilizador perante a tarefa, qual a sua relação com os sistemas informáticos e os caminhos necessários a percorrer para realizar a tarefa. Após este conhecimento, o avaliador deve realizar as tarefas tendo em conta os passos que o utilizador faz para realizar a tarefa.

9.2.3. Técnicas objectivas

Este tipo de técnicas contam com a participação directa dos utilizadores e dizem respeito aos ensaios de interacção e sessões com sistema espões. Este tipo de técnicas já foram abordadas nesta tese no ponto 6.2. Merece ainda a nossa atenção as sessões com sistemas espões e os ensaios de interacção.

- ***Sistemas espões***

Os sistemas espões têm como finalidade a observação directa dos utilizadores, utilizando um utilitário que permanece residente na máquina do utilizador junto à aplicação de teste. A finalidade é registar as acções realizadas pelo utilizador numa situação real de trabalho. Este tipo de aplicações tem a vantagem de permitir que o

utilizador, mesmo sabendo que está a ser testado, não fique inibido. Em contrapartida, com este tipo de testes estamos limitados às ferramentas de espionagem existentes.

- **Ensaio de interacção**

Um ensaio de interacção consiste numa simulação da utilização de um sistema onde participam pessoas representativas da situação real. O objectivo deste tipo de ensaio é que essas pessoas realizem tarefas típicas como as que executam numa situação real de trabalho. A montagem de um ensaio de interacção, segundo Walter Abreu Cybis, passa por três fases importantes (figura 10). Uma primeira fase chamada de Análise preliminar, onde se toma conhecimento do software a utilizar e é feito um pré – diagnostico ergonómico. Uma segunda fase chamada Definição dos cenários e da amostra, onde se pretende recolher informação acerca do utilizador e das tarefas em função do utilizador. E, por fim, uma última Análise do processo de ensaio da interacção, chamada de realização de ensaios, onde se realizam os ensaios propriamente ditos.

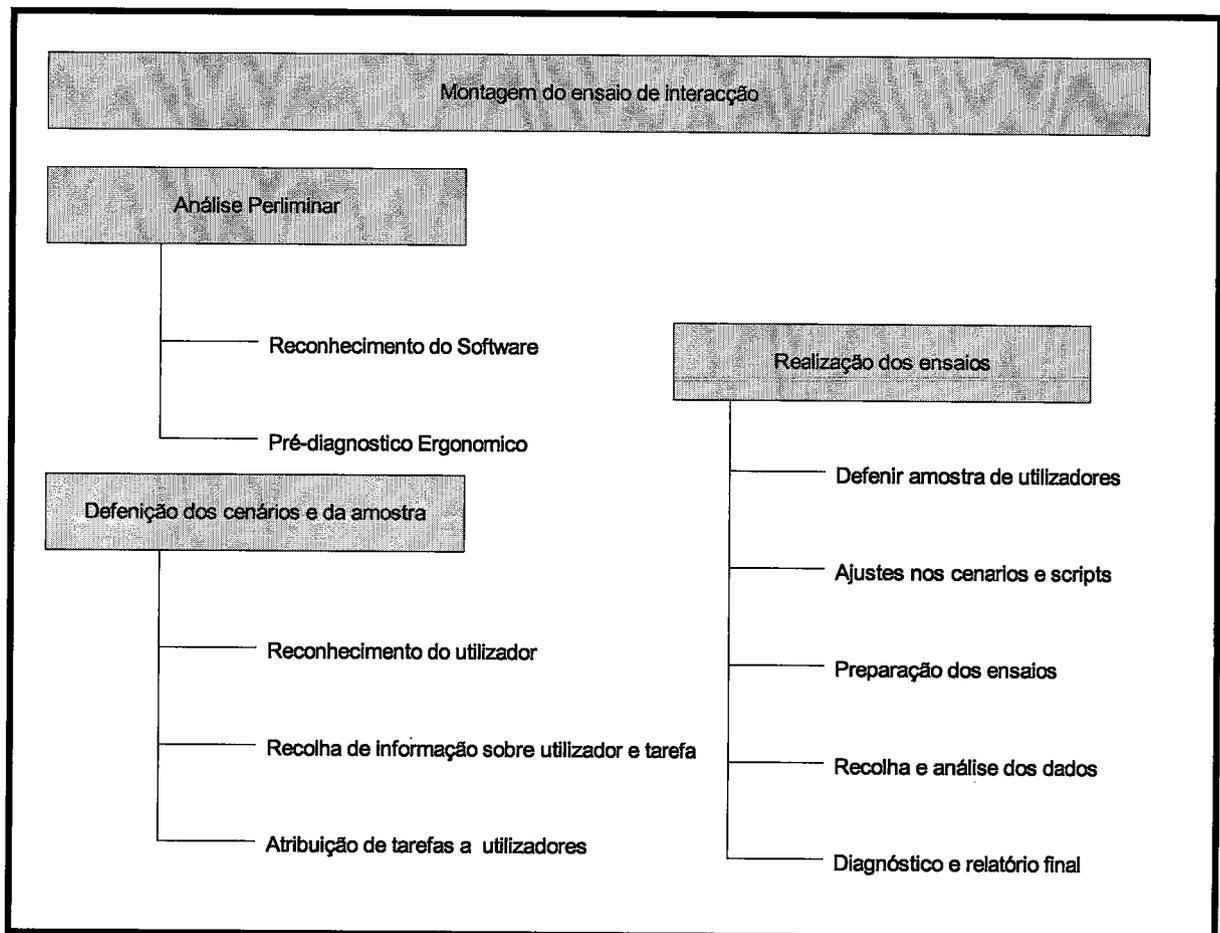


Figura 10 - Fases do ensaio de interacção, adaptado de Cybis, 2003, p.111

Na primeira fase, o reconhecimento do software deve ser feito recorrendo a entrevistas com os responsáveis pelo desenvolvimento do software. Para tal, propomos a utilização do modelo II em anexo nesta tese. Este tipo de análise destina-se a compreender o ciclo de desenvolvimento pelo qual passou o software (Cybis, 2003, p122). A partir desta informação estamos aptos para realizar o pré-diagnóstico, que não é mais do que a utilização de uma avaliação heurística ou a utilização de checklists para a inspecção ergonómica e que resulta num conjunto de hipótese sobre problemas de *Usabilidade* que serão posteriormente testados durante os ensaios de interacção.

Quanto à segunda fase do processo de ensaio da interacção, tal como descrevemos, pretende-se recolher informação acerca do utilizador e das tarefas em função de cada utilizador. Como primeira tarefa iremos fazer o reconhecimento dos utilizadores, ou melhor iremos verificar se os utilizadores possuem o perfil descrito pelos projectistas do sistema. Nesta fase devemos também seleccionar um conjunto de utilizadores que irão realizar os testes de software. Devemos também, nesta fase, recolher informação acerca das tarefas em função dos utilizadores. Para tal, devemos, mais uma vez, recorrer aos questionários, de forma a retirar informação como: recursos disponíveis; contexto em que se insere a tarefa; nível de formação dos utilizadores; frequência de utilização de uma dada funcionalidade, etc.

Quanto à escolha da amostra de utilizadores para a realização dos testes, embora possa ser feita na segunda fase do processo de ensaio da interacção, como referimos, ela aparece no nosso modelo na terceira fase, ou seja na realização dos ensaios. Existem alguns cuidados a tomar a quando da escolha da amostra dos utilizadores. Estes devem ser *utilizadores directos do software*, ou seja, aqueles que necessitem realmente do software para realizarem as suas tarefas. O grupo também deve conter *utilizadores novatos* e *utilizadores experientes* nas tarefas, uma vez que os utilizadores novatos permitem recolher informação sobre facilidade de aprendizagem e simplicidade de utilização. Por outro lado, os utilizadores experientes permitem retirar informação sobre a organização das funções e repartição das informações.

Depois de escolhidos os utilizadores partimos para os ensaios propriamente ditos. A preparação dos ensaios também requer alguns cuidados. A escolha do local de realização dos testes, o equipamento necessário, a escolha de técnicas de diálogo e a escolha de estratégias de intervenção em caso de impasse devem nesta fase ser definidas.

Quanto às situações de impasse, ou seja quando o utilizador não sabe que atitude deve tomar perante a interface para realizar a tarefa, devemos deixar que este tente resolver sozinho a tarefa e, muito importante, não tomar atitudes que possam inibir o utilizador na continuação do ensaio. Se o utilizador permanecer muito nervoso ou constrangido devemos interromper o ensaio.

Preferencialmente, o lugar de ensaio deve ser realizado no local real de trabalho de cada utilizador. No caso de não ser possível, devemos utilizar um laboratório preparado para o efeito. O tempo de duração do teste não deve demorar mais de uma hora. Além do utilizador, deve participar no teste 1 ou 2 ergonomistas observadores e um responsável pelo funcionamento dos equipamentos.

Os resultados obtidos devem ser relatados e comentados num caderno de encargos que irá ser entregue aos directores de projecto, onde constam as situações inesperadas verificadas, os incidentes e erros registados e os comentários sobre quais devem ser a prioridade de resolução dos problemas.

9.3. Projecto de Avaliação

Como referimos, o processo de avaliação direcciona e influencia todo o processo de desenvolvimento de produtos interactivos. Desta forma, o sucesso no desenvolvimento ou manutenção de um sistema interactivo está directamente relacionado com a verificação e validação da sua *Usabilidade*. Desta forma, tal como refere Alan Dix (1993), Cybis (2003), podemos evitar desperdícios de tempo e de ideias ao desenvolver aplicações que não sigam a linha de orientação defendida pela

Usabilidade de um sistema. Desta forma, o processo e Avaliação de uma interface homem máquina deve ser organizado segundo uma metodologia de projecto.

Como vimos no capítulo anterior, a *Usabilidade* de um sistema associa as características dos vários tipos de utilizadores, as tarefas, os equipamentos e o ambiente de trabalho do sistema. Desta forma, no desenvolvimento de um sistema interactivo, um avaliador de *Usabilidade*, ou melhor o responsável pela elaboração dos testes de *Usabilidade* do nosso sistema, terá como tarefa inicial a elaboração de um plano de testes de *Usabilidade*, no qual desencadeará uma sequência de testes e avaliações em cada uma das partes de desenvolvimento do produto.

9.3.1. Plano de testes

O desenvolvimento de um plano de testes é a chave para o sucesso de um sistema de informação. Ele deve ser feito em cada etapa do processo de desenvolvimento e, na sua génese, tem como finalidade combinar vários tipos de técnicas para testar o projecto e a implementação do sistema. Ele deve prever para cada versão intermediária do sistema, mesmo que não passe de um conjunto de ideias, qual o tipo de teste apropriado quais as condições de teste e qual o resultado admitido (Cybis,2003,p129). O mesmo autor propõe o seguinte plano de testes:

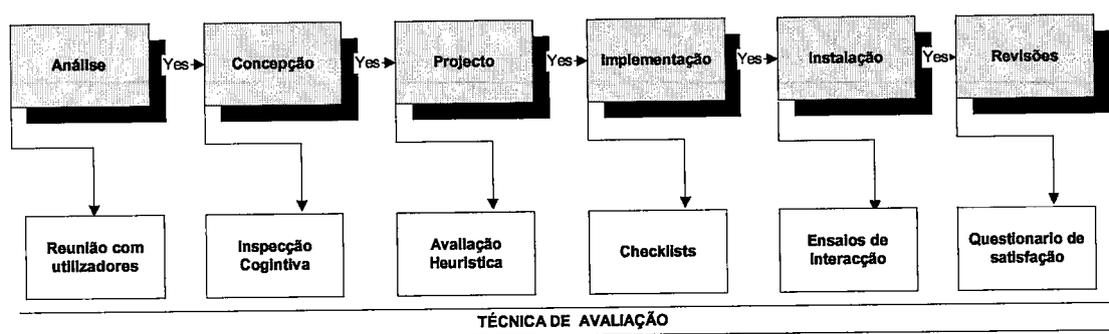


Figura 11 – Plano de testes, adaptado de Cybis, 2003, p. 129.

Para cada uma das etapas de desenvolvimento do sistema é realizada uma técnica específica de avaliação. Como podemos ver, e tal como fizemos referência na parte anterior desta tese, a avaliação deve ser feita continuamente e durante todo o processo de desenvolvimento.

Parte 8

**Proposta Modelo de Construção da Interface
Homem Máquina em Sistemas de Informação**

10. Modelo

Como vimos a construção de interfaces homem máquina é feita através de um conjunto de actividades todas elas centradas nos utilizadores. Dependendo do paradigma de desenvolvimento do sistema de informação elas podem estar encadeadas de diversas formas. Ao longo desta tese elas foram divididas em três fases distintas: (1) Análise, (2) Síntese e (3) Avaliação. Cada uma destas fases incorpora necessariamente um conjunto de tarefa e acções que não podem deixar de ser tidas em consideração para quem quer projectar e implementar um sistema de informação tendo em conta os factores de *Usabilidade* e de Interface Homem Máquina.

Estas três realidades, que genericamente condicionam e orientam a actividade de desenvolvimento de interfaces homem máquina, podem ser representadas tal como mostra a figura seguinte utilizando uma forma triangular.



Figura 12 – Três realidades do processo de desenvolvimento.

Actualmente, existem muitos modelos e teorias de desenvolvimento de sistemas de informação. No entanto, tal como diz Lauesen (2005), na prática os programadores raramente usam um modelo com exactidão. Normalmente, estes modificam os modelos existentes ou criam os seus próprios modelos.

Desta forma, a esta construção, a que demos o nome de *Modelo de Avaliação Triangular de Interfaces (MATI)*, não tem como objectivo a criação de um modelo que sirva de base para o processo de desenvolvimento do sistema de informação, mas sim a criação de uma estrutura que possa ser encaixada em qualquer processo de desenvolvimento de software.

Alguns dos aspectos mais importantes que rodeiam a criação de sistemas de informação interactivos têm vindo a ser discutidos nos primeiros capítulos desta tese. Contudo, parece oportuno nesta fase propor um modelo que sustente todos estes aspectos e que passamos de imediato a tecer algumas considerações.

10.1. Modelo de Avaliação Triangular de Interfaces (MATI)

O objectivo de um modelo do processo de desenvolvimento é proporcionar ao projecto uma estrutura que reduza os riscos. Um projecto sem estrutura é impossível de gerir. Não pode ser estimado, não pode ter marcos definidos, o seu progresso não pode ser monitorizado e não podem ser feitas quaisquer promessas ao cliente sobre o seu custo ou a sua qualidade (Miguel, 2003, p89).

Não é pretensão nossa fazer uma abordagem a estes modelos, uma vez que existe actualmente muita informação disponível acerca de modelos do processo de desenvolvimento. O nosso modelo tem como objectivo o de encontrar oportunidades e técnicas de verificação durante o processo de desenvolvimento, independentemente do modelo seguido pelos projectistas. Estas oportunidades estão relacionadas a ver com a construção de interfaces homem máquina em sistemas de informação.

No desenvolvimento de interacção, o uso da interacção tem um motivo adicional, relacionado com a imprevisibilidade do comportamento humano (Gould & Lewis, 1985). Desta forma, o desenvolvimento de uma interface com o utilizador, sobretudo a componente de interacção, tem de ser essencial e inerentemente interactivo, isto é, tem de ser um processo que se corrige a si próprio. Existe, desta forma, uma necessidade de alternar entre processos de análise, onde estão reflectidos os pontos de vista do sistema e caminham na direcção do utilizador, e um processo de síntese onde estão reflectidos as necessidades dos utilizadores e a acompanhar estes processos um processo de avaliação que irá avaliar esse algo que foi criado.

Desta forma, o modelo proposto está dividido em duas fases. Uma primeira fase chamada de realidade e contexto operacional do sistema e uma segunda fase de especificação, projecto, e implementação. Estas duas fases serão sempre

acompanhadas pelo processo de avaliação que irá servir de ferramenta de avaliação através da utilização das várias técnicas descritas no capítulo anterior.

10.1.1. Primeira fase do MATI - Análise

A primeira fase do nosso modelo é composta por quatro etapas tal como mostra a figura:

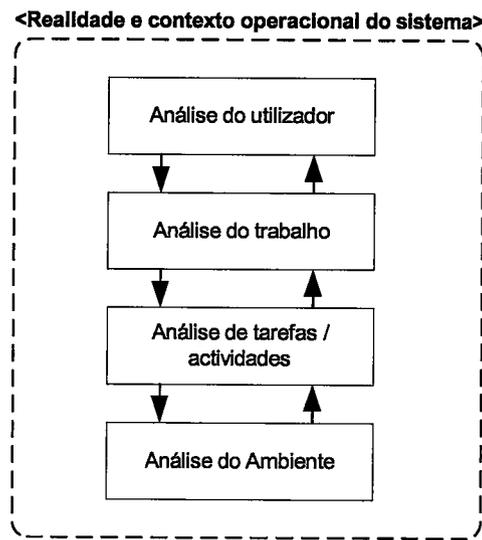


Figura 13 – Primeira fase do MATI

Esta é a fase de Análise. Nela é retirada informação relativa aos utilizadores, tipo de população, nível de exigências, motivação e expectativas dos utilizadores perante o sistema. É retirada ainda informação sobre o trabalho real a efectuar, problemas com que nos vamos deparar, funcionalidades que o nosso sistema irá ter e alternativas possíveis. Também deve acompanhar este estudo o conhecimento do ambiente em que se irá inserir o nosso sistema (Ver parte 8 desta tese).

Ferramentas:

As ferramentas a utilizar no que diz respeito ao conhecimento dos utilizadores estão indicadas no modelo I em anexo nesta tese. Para a Análise do trabalho, além das entrevistas com os trabalhadores, modelação das tarefas utilizando a técnica HTA e a Análise do ambiente utilizamos o modelo II em anexo. Como técnicas de avaliação devem ser utilizadas reuniões com os utilizadores e inspecções cognitivas.

10.1.2. Segunda fase do MATI – Síntese

Esta fase é composta por três fases, como se demonstra na figura seguinte:

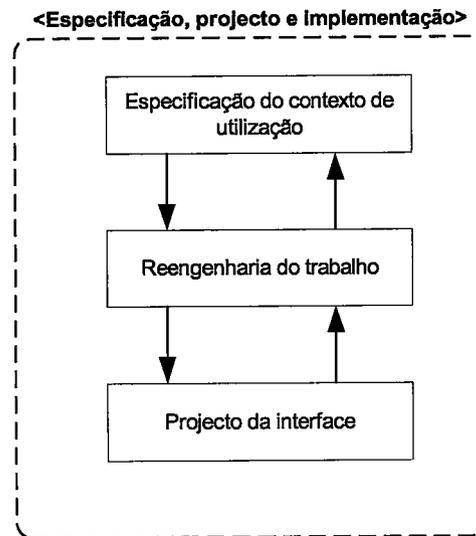


Figura 14- Segunda fase do MATI

Uma primeira relaciona-se com a especificação do contexto de utilização, na qual iremos ter contacto com o tipo de utilizadores que irão utilizar o sistema, as tarefas que executam, o equipamento existente e os ambientes físicos e sociais que irão ser influenciados. Esta fase só é possível, como vimos na parte 9 desta tese, com o envolvimento de todos os intervenientes no processo. Este estudo é o ponto de partida para a fase de projecto da interface. No entanto, para que se possa passar para o projecto propriamente dito, é necessário especificar a estrutura das tarefas futuras, sendo necessária a existência da Reengenharia do trabalho.

Ferramentas:

Modelação das actividades futuras utiliza-se os diagramas de UML orientados a objectos. Quanto ao projecto da interface, este deve ser feito segundo o desenho centrado nos utilizadores descrito na segunda parte desta tese. Como técnicas de avaliação, a avaliação heurística e os testes de *Usabilidade* devem ser feitas na fase de especificação do contexto de utilização e no próprio projecto da interface. Na fase de projecto, tal como vimos, também é importante fazer os testes e ensaios de interacção.

10.1.3. Modelo MATI

Estas duas fases do projecto, como vimos, são sempre acompanhadas de uma terceira fase tão ou mais importante que as duas anteriores. Esta fase, a que demos o nome de avaliação, ocupa a base do nosso triângulo. Tal como vimos no capítulo anterior, este processo direcciona e influencia todo o processo de desenvolvimento de produtos interactivos. Neste sentido, propõem-se o seguinte modelo de avaliação:

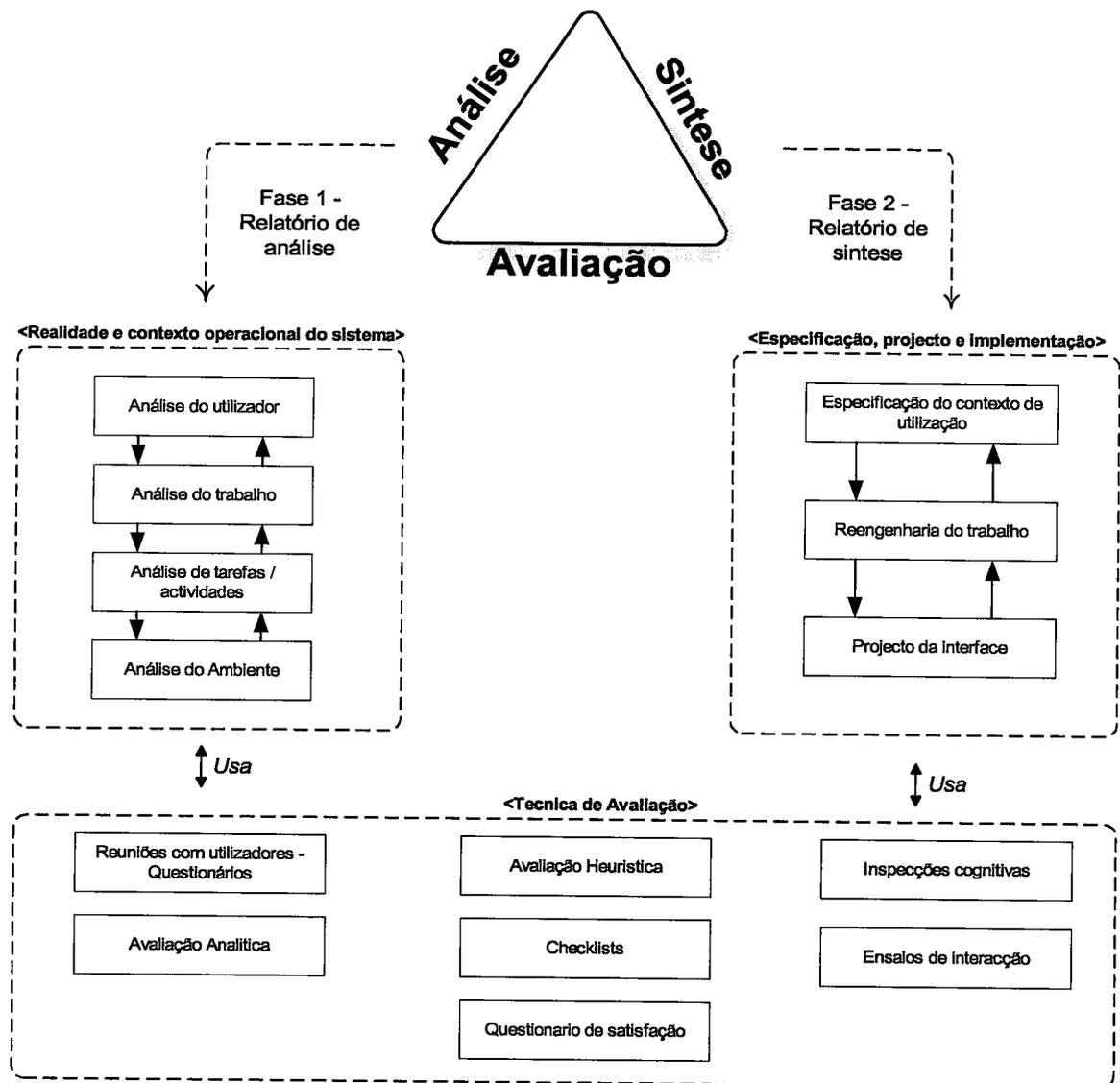


Figura 15 - Modelo de Avaliação Triangular de Interface

Este modelo, para além de poder ser utilizado em qualquer modelo de processo de desenvolvimento de um sistema de informação, permite ajustar-se à ideia de

prototipagem rápida, que tem como vantagem permitir a avaliação numa fase mais precoce do processo de desenvolvimento. A utilização de modelos centrados na avaliação, ou seja, os resultados de cada actividade são avaliados antes de se prosseguir para a actividade seguinte, faz com que os produtos sejam ajustados de acordo com as necessidades dos utilizadores e do trabalho propriamente dito.

O desenvolvimento de interfaces homem máquina é apenas uma parte de um conjunto maior de processos de desenvolvimento de um sistema interactivo. Desta forma, torna-se necessário aplicar a cada uma das áreas de desenvolvimento o processo de avaliação tendo em conta os utilizadores, já que estes são parte integrante do sistema. Considerar também a tecnologia, já que a linguagem de programação a utilizar pode por problemas e restrições ao processo de design e, por fim, ter em atenção o próprio ambiente em que irá ser instalado o próprio sistema.

10.1.4. Aplicação do MATI no modelo em Cascata

O modelo em Cascata é um dos muitos modelos utilizados no processo de desenvolvimento de sistemas. Este método, que foi apresentado em 1970 por Winnston Royce, propõe uma abordagem sistemática, linear e sequencial ao desenvolvimento de software. Segundo este modelo, e tal como se demonstra na figura seguinte, o desenvolvimento de software é constituído por um conjunto de actividades executadas de forma sequencial e sistemática (Miguel, 2003, pag.91) que se iniciam ao nível do sistema de informação (requisitos do sistema) e prosseguem com as fases de requisitos de software, análise, desenho dos programas, codificação, testes e operação (onde se processa o suporte).

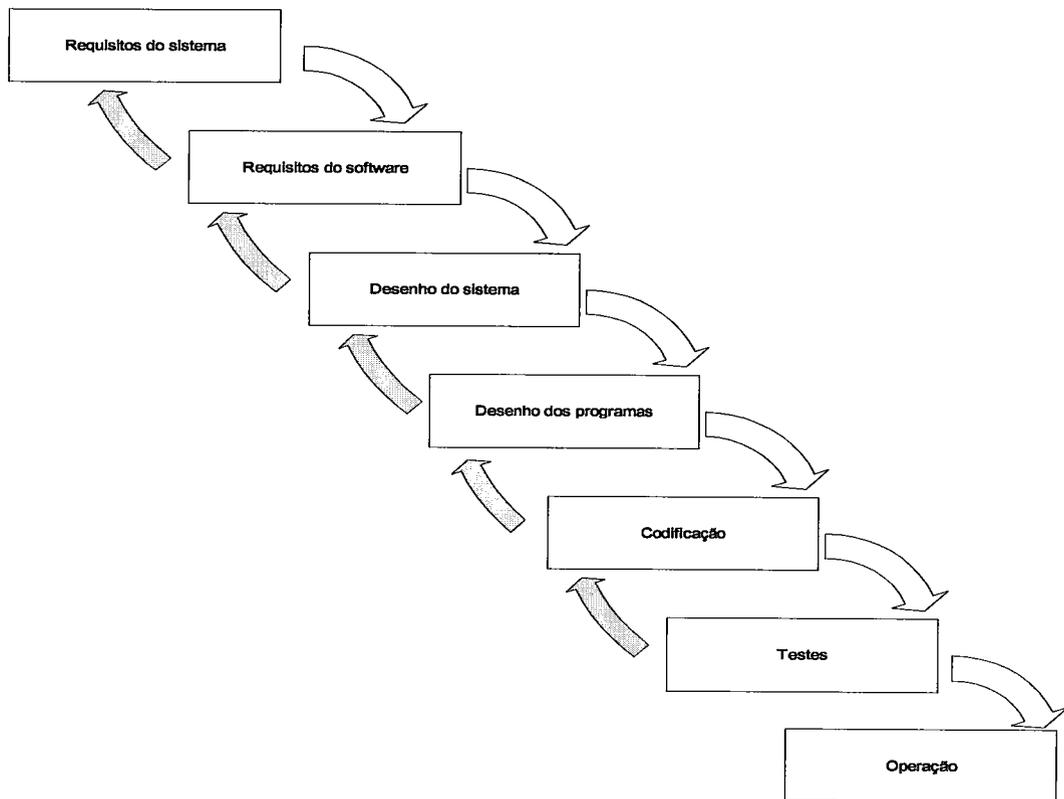


Figura 16 - Modelo em cascata, adaptado de António Miguel, 2003, pag 91

Este modelo é dos mais utilizados na Engenharia de Software. No entanto, segundo António Miguel, é alvo de algumas críticas:

1. Não reconhece explicitamente a necessidade de revisão, isto é, a necessidade de retornar a fases anteriores e refazer processos à luz da informação obtida durante o desenvolvimento.
2. Durante o desenvolvimento têm de ser tomadas muitas decisões. Essas decisões podem conduzir a um certo número de caminhos e resultados possíveis
3. É muitas vezes difícil ao cliente/utilizador expressar, de forma explícita, todos os seus requisitos. Por ser sequencial e linear, tem dificuldade em acomodar a incerteza existente no início dos projectos.
4. O utilizador tem de esperar, por vezes muito tempo, até lhe ser entregue uma versão operacional do sistema.

Iremos agora passar a demonstrar como podemos aplicar o MATI num projecto de desenvolvimento de um sistema que utiliza o método em cascata como base de desenvolvimento. O objectivo é criar ou modificar um sistema e torna-lo interactivo e usável de forma satisfatória pelos utilizadores.

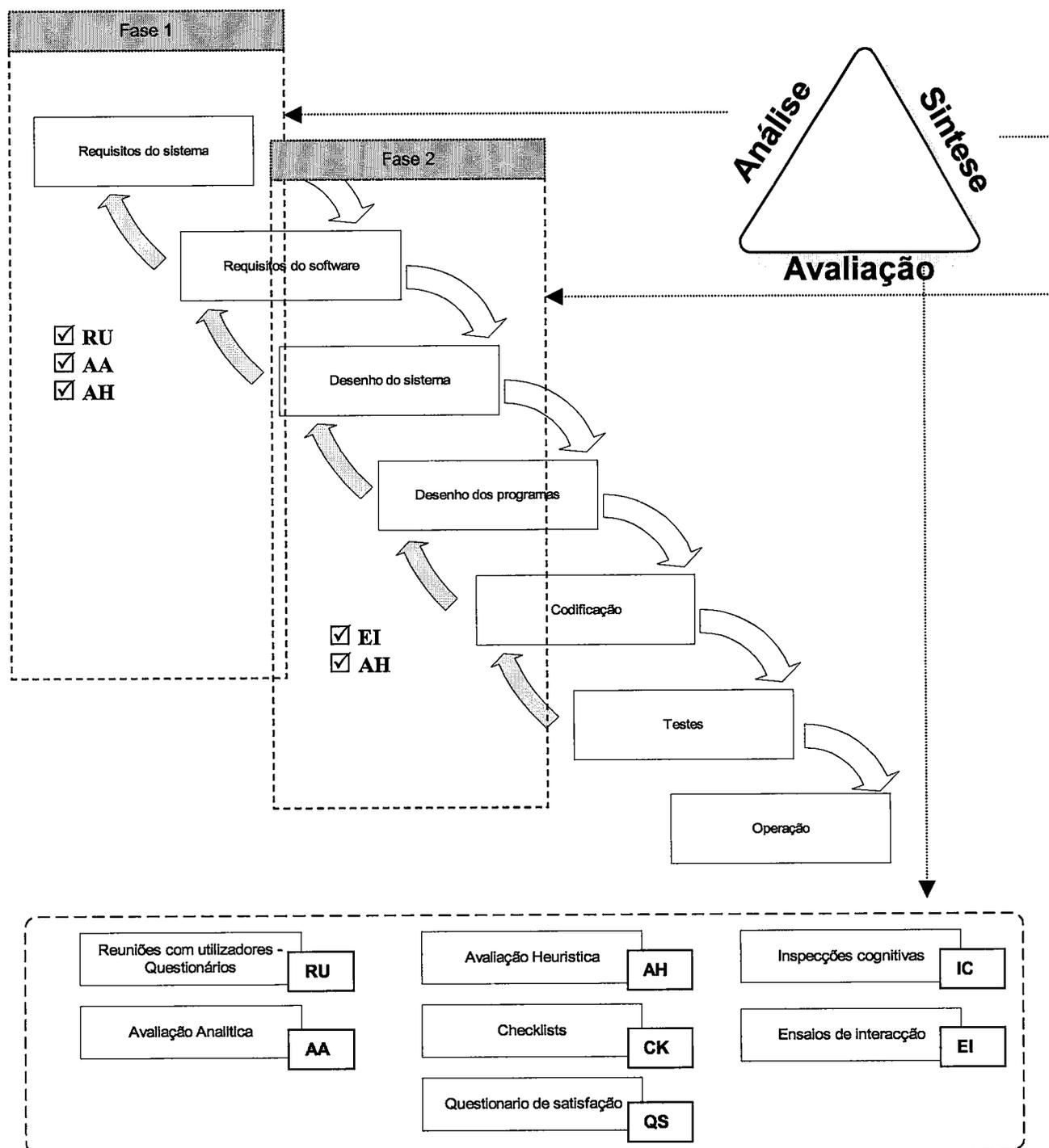


Figura 17 – Aplicação do MATI no modelo de desenvolvimento em Cascata

Como demonstrámos, a aplicação do MATI num processo de desenvolvimento utilizando o modelo em cascata é perfeitamente possível. As vantagens da aplicação deste tipo de abordagem são várias. Destacamos o facto de tornar o sistema mais interactivo e centrado no utilizador e permitir ainda eliminar algumas das lacunas encontradas no modelo. É eliminado, por exemplo, o grau de incerteza a quando do início do projecto, uma vez que este é feito de acordo com as necessidades reais do utilizador, da tecnologia disponível e das tarefas necessárias à execução do trabalho propriamente dito. Por outro lado, o utilizador já não tem que esperar durante largos períodos de tempo por uma versão operacional do sistema, uma vez que ele é parte integrante do sistema, isto é, ele participa activamente no desenvolvimento do sistema e os testes feitos com este permitem que ele se familiarize com as novas funcionalidades, com os novas interfaces, etc. Desta forma, é possível economizar em tempo de aprendizagem, manutenção e satisfação do utilizador. Por outro lado, as necessidades de revisão, ou seja o retorno as fases anteriores, apesar de possível, já não vai ser necessário, uma vez que à partida, depois dos testes feitos e dos protótipos testados, o sistema não vai necessitar de muitas alterações.

Quanto aos testes efectuados, propõe-se, numa primeira fase, a realização de Reuniões com os Utilizadores, a Avaliação Analítica e uma Avaliação Heurística. Estas serão feitas na fase de Análise, que compreende a fase de requisitos do sistema e do software do modelo tradicional de cascata. O objectivo destes testes é o de verificar se a Análise do utilizador, das tarefas e da tecnologia disponível são executadas correctamente.

Numa segunda fase, propõe-se a realização de Ensaios de Interação e uma nova avaliação heurística, acompanhada de testes de *Usabilidade*. Estes procedimentos deverão ser realizados na fase de síntese e que inclui as fases de requisitos de software, desenho do sistema, desenho de programas e codificação do modelo tradicional de cascata. O objectivo destes testes é o de garantir que o sistema seja construído de acordo com as necessidades dos utilizadores e da tecnologia disponível. A codificação é incluída nesta fase somente para que sejam criados protótipos de interfaces que sejam avaliados utilizando os ensaios de interação. Como referimos anteriormente, o nosso modelo fica completo na fase de codificação

de protótipos, uma vez que, se o processo de Análise e de Síntese forem bem efectuados, para os processos de implementação e testes basta seguir o que foi projectado e testado nas fases anteriores. De qualquer forma, o modelo prevê que o processo de avaliação, que deve estar presente em todas as fases de construção do sistema, possa ser efectuado através de Checklists, Inspeções Cognitivas e Questionários de Satisfação, nas fases de codificação, testes e operação respectivamente.

Parte 9

Conclusão e considerações finais

11. Conclusão

Os primeiros sistemas informatizados eram destinados única e exclusivamente à resolução de problemas em áreas científicas. Os utilizadores deste tipo de sistemas eram, normalmente, os próprios criadores do sistema. Mais tarde, estes programas passaram a ser distribuídos por um grupo restrito de pessoas, que recebiam formação durante várias horas ou mesmo dias, para poderem realizar um conjunto de tarefas básicas. Até essa altura, tudo ia bem no que diz respeito às interfaces homem máquina. Ao longo dos tempos, essa realidade mudou. A disseminação dos computadores gerou uma variedade de novos utilizadores que se servem do computador nas tarefas mais rotineiras, trabalho, lazer, etc.

Deste forma, a interface, e porque é ela a responsável pela comunicação entre o homem e o computador, tem de evoluir. Essa evolução passa necessariamente por produzir interfaces de acordo com as necessidades dos utilizadores. Novas questões como conhecer o perfil dos utilizadores, qual a sua experiência na utilização de sistemas informáticos, quais as suas necessidades perante a realização de uma tarefa, ou um conjunto de tarefas, são novas e importantes realidades que, até então, não eram tidas em consideração. Paralelamente ao perfil dos utilizadores, outros aspectos merecem atenção no design da interface. Aspectos como critérios ergonómicos, elementos visuais, a cor, as linhas, o tipo de letra, a forma de disposição dos elementos, etc.

Esta tese procurou, numa primeira fase, mostrar os conceitos evolutivos no domínio da construção de interfaces homem máquina, aprofundando-se o estudo das várias fases necessárias à construção de um sistema de informação, utilizando ferramentas e técnicas de construção de soluções centradas nos utilizadores. Neste sentido, foram citados e discutidos vários tipos de avaliações que têm de ser efectuadas com utilizadores reais dos sistema, as quais devem ser feitas na presença dos utilizadores utilizando um cenário real de trabalho.

Numa segunda fase, propõe-se um modelo de desenvolvimento a que demos o nome de *Modelo de Análise Triangular de Interfaces* (MATI), e que tem como objectivo o de encontrar oportunidades técnicas de verificação durante o processo de

desenvolvimento, independentemente do modelo seguido pelos projectistas e programadores do sistema de informação. Estas oportunidades estão relacionadas com a construção de interfaces homem máquina em sistemas de informação.

Por fim, o modelo proposto é aplicado num projecto que utiliza como modelo de desenvolvimento o método em cascata.

11.1. Perspectivas de investigação futura

Este trabalho pretende lançar pistas para um conjunto de investigações futuras, as quais serão desenvolvidas em diversos âmbitos. Desde logo, parece fundamental proceder ao teste do modelo numa situação real de desenvolvimento de um sistema de informação. Por outro lado, o modelo terá de ser verificado não só para um modelo de desenvolvimento que utilize como base o método em Cascata, mas para outro tipo de modelos, nomeadamente o *Modelo em V*, o *Modelo Prototipagem*, os *Modelos de Desenvolvimento Evolutivos*, o *Modelo em Espiral*, o *Modelo Dinâmico de Desenvolvimento de Sistemas*, etc. Assim, a aposta passa pelo desenvolvimento de um modelo que sirva de suporte para todos os modelos tradicionais de desenvolvimento e que os modifique para que se produzam sistemas de informação com interfaces homem máquina. Pretende-se que o mesmo se desenvolva no âmbito de um projecto tendo em vista a realização do Doutoramento nesta área do conhecimento, o qual visará, em última análise, a criação teste e validação do Modelo de Análise Triangular de Interfaces (MATI).

12. Referências Bibliográficas

Alan Dix, Janet Finlay, Gregory D. Abowd, Russel Beale, 2004, “*Human-Computer Interaction, Third Edition*”, Pearson-Prentice Hall, ISBN:0130-461091

Associação portuguesa de Ergonomia (APERGO).[consultado em 22 de Dezembro de 2003]. Disponível no site <<http://www.apergo.pt/ergonomia/defenicao.php>>.

Barros, Vanessa T.O., 2003, “ *Avaliação da Interface de um aplicativo computacional através de teste de Usabilidade, questionário ergonómico e análise gráfica do design*” Dissertação de Mestrado, Universidade Federal de Santa Catarina, orientador. Prf. Milton Luiz Horn Vieira, Dr., Florianópolis.

Booch G., Rumbaugh J., Jacobson I., 1998, “The Unified Modelling Language User Guide”, Addison Wesley.

Booch Garady, 1994, “*Object-Oriented Analysis and Design whit application*”, second edition, Addison Wesley.

Brad A. Myers , 1993, “*Why are Human- Computer Interfaces Difficult to Design and Implement?*”, Carnegie Mellon University School of Computer Science Technical Report CMU-CS-93-183, Julho de 1993.

Brenda Laurel, 1990, “ *The Art of Human-Computer Interface Design*” , Addison-Wesley, ISBN: 0201517973.

Charles H. Schmauch, 1994, “*ISO 9000 for Software Developers, Revised Edition*” ASQC Quality Press, Milwaukee, Wisconsin, ISBN: 0-87389-348-4.

Cybis, W.; Pimenta, M.; Silveira, M.; Gamez, L.; “*Uma Abordagem Ergonómica para Desenvolvimento de Sistema Iterativos*” artigo publicado pelo laboratório de utilizabilidade da Informática (LabUtil/Inf) – Universidade Federal de Santa Catarina (UFSC) – Departamento de Informática e Estatística (INE), 2004.

CYBIS,W.A. 1997, “*Abordagem Ergonómica para IHC*”, Sebenta LabIUtil, Universidade Federal de Santa Catarina.

Disponível no site<http://www.useit.com/papers/heuristic/heuristic_evaluation.html>

Cybis W. , 1993, “Introdução ao Ciclo da Engenharia da *Usabilidade*”. Disponível no site: http://www.labiutil.inf.ufsc.br/hiperdocumento/unidade3_2.html

Edward Yourdon, 1990, “*Análise estruturada moderna*”, tradução Dalton Conde de Alencar, Rio de Janeiro, Campus, ISBN 85-7001-615-8

Eric M. Schaffer,(2000) “*Human Factors International, User – Centred Solutions*”, McGraw-Hill.

Figueiredo, Bruno, 2004, “ *Web Design, estrutura, concepção e produção de sites Web*”, 2ª edição actualizada e aumentada, FCA, ISBN: 9727224334

ISO/IEC FDIS 9126-1 : Software Engineering- Product quality – parte 1: Quality Model (2000), [consultado em 10 de Dezembro de 2003]. Disponível no site: <<http://www.usability.serco.com/trump/resources/standards.html#9126-1>>

J. Nielson, 1994. “Heuristic evaluation”. In Nielson, J., and Mack, R.L. (Eds.), “Usability Inspection Methods”, John Wiley & Sons, New York, NY.

Jakob Nielson, 2003 “*Then Usability Heuristics*” , [consultado em 11 de Novembro de2004]. Disponível no site:

<http://www.useit.com/papers/heuristic/heuristic_list.html>

James Martin, Carma McClure, 1988, “ *Técnicas Estruturadas e Case*”, tradução: Lúcia Faria Silva, Makron Books do Brasil editora Lda, McGraw-Hill Ltda. São Paulo, CDD-001.624-001.6425

Laboratório da Engenharia da Usabilidade. Definição oficial 2003 [consultado em 11 de Novembro de 2003]. Disponível no site:

<<http://sphere.rdc.puc.rio.br/sobrepuc/depto/dad/leui/oficial.html>>.

Laboratório de Ergonomia e Usabilidade de Interfaces em Sistemas Homem-Tecnologia (2000). [consultado em 30 de Dezembro de 2003]. Disponível no site

<<http://sphere.rdc.puc-rio.br/sobrepuc/depto/dad/leui/>>.

Leonardo Molinari, 2003, “*Testes de Software – Produzindo Sistemas Melhores e mais Confiáveis*”, Editora Érica Ltda. ISBN: 85-7194-959-X.

Luciano Roque, 2004, “*Interacção com o Utilizar 2001/2002*”, Análise de tarefas, dei.uc.pt, Departamento de Engenharia Informática, FCT Universidade de Coimbra, [consultado em 10 de Agosto de 2004]. Disponível em:

<http://eden.dei.uc.pt/~iu/tarefas.htm>

Luís Amaral, João Varajão, 2000, “*Planeamento de Sistemas de Informação*”, FCA - Editora de Informática Lda., ISBN: 972-722-193-9

Marcus, A., (1887), “*A Tool for Computer graphics Communication*” – Colour in Computer Graphics nº. 24 – SIGGRAPH.

Mark Perrow, 2000, “*Web Site Usability Handbook*”, Charles River Media, inc., USA, ISBN: 1-58450-026-3.

Miguel Santinho, Setembro 2001. Artigo “ Avaliação heurística e teste com utilizadores: dois métodos, dois resultados - Análise Comparada da Usabilidade nos sítios das televisões portuguesas.” Disponível no site <<http://www.simplicidade.com>>

Nielsen, J., “*Heuristic Evaluation: How-To*”. [consultado em 17 de Março de 2004].

Disponível no site <http://www.useit.com/papers/>

Nuno Jardim Nunes, 2003, “*A Engenharia da Usabilidade e o Ciclo de Desenvolvimento de Software*”, cap II, Universidade de Madeira .

Paulo Alves e José A. Pires,(2000), Artigo “*A Usabilidade em Software educativo: princípios e técnicas*”.

Relatório de Investigação “*Avaliação Ergonómica de Interfaces*” 2003. Projecto: “*Validação e Desenvolvimento de um Modelo de Programas Educativos Baseados em Televisão Interactiva*”, POCTI/Com 43208/2001. Investigador Principal: **Marques, José Alexandre Cardoso**; Investigador: **Henriques, Rui**; Bolseira: **Tomas Olga**; Coordenador da Unidade de Investigação e Desenvolvimento: **Damásio, Manuel José**.

Sally Shlaer e Stephen J. Mellor, “*Análise de Sistemas Orientada para Objectos*”, revisão técnica Eber Schmitz, São Paulo; McGraw-Hill: Newstec,1990, CDD-001.6425-0001.61.

Silva, Alberto e Videira, Carlos, (2001), “*UML Metodologias e ferramentas CASE*”, Centro Atlântico.

Spool, Jared, et. Al .1997, “*Web Site Usability: A Designer’s Guide*”, New York, Morgan Kaufmann Press

Teofanos M., Laskoski, S., “The benefits of usability in the software procurement process”, National Institute of Standards and thechnology, [consultado em 14 de Junho de 2004]. Disponível no site:

<http://zing.ncsl.gov/iusr/documents/why_should_I_use_the_CIF_files/frame.html>

The usability group (1999). “What is strategic usability?”. Disponível no site <http://www.usability.com/umi_what.htm>.

13. Sites de referência

Usable information technology - <http://www.useit.com/>

Consultadoria em Tecnologias de Informação - <http://www.simplicidade.com>

Laboratório de utilizabilidade da Informática - <http://www.labiutil.inf.ufsc.br/>

Laboratório da Engenharia da Usabilidade - <http://sphere.rdc.pucrio.br/>

Associação portuguesa de Ergonomia - <http://www.apergo.pt/>

The usability group - <http://www.usability.com/>

Software Test Online Resources - <http://www.mtsu.edu/>

Seara.comodidade, sites com Usabilidade - <http://www.seara.com/>

Eropean design centre – <http://www.edc.nl/>

Nodus Information Technology - <http://www.nodus.com/>

Acessibilidade - <http://www.acessibilidade.net/>

Grupo Português dos profissionais de Usabilidade - <http://www.usabilidade.org/>

Usabilidade - <http://www.usability.net/>

Usability Heuristics - <http://www.user-experience.org/>

Interface Pessoa Máquina - <http://www.mega.ist.utl.pt/~ic.ipm/>

14. Anexos

Modelo I – Conhecimento do utilizador e de uma dada tarefa

Nome	
------	--

Categoria	
-----------	--

Idade	
-------	--

Experiência Profissional

--

Tarefa	
--------	--

<i>Experiência a executar a tarefa</i>	_____ Meses _____ Anos
--	------------------------

Experiência tecnológica

<i>Forte</i>	<input type="checkbox"/>
<i>Fraca</i>	<input type="checkbox"/>
<i>Media</i>	<input type="checkbox"/>

Frequência com que executa a tarefa (dia)

<i>5 Vezes</i>	<input type="checkbox"/>
<i>10 Vezes</i>	<input type="checkbox"/>
<i>20 Vezes</i>	<input type="checkbox"/>
<i>Mais de 20</i>	<input type="checkbox"/>

Já trabalhou com um sistema similar automatizado a executar a mesma tarefa?

Sim

Não

Se respondeu Sim diga as diferenças encontradas:

Se respondeu Não, responda:

<i>A tarefa é executada mais rapidamente</i>	<input type="checkbox"/>
<i>A tarefa demora mais tempo a ser executada</i>	<input type="checkbox"/>
<i>A tarefa é executada na perfeição</i>	<input type="checkbox"/>
<i>O esforço feito pelo utilizador é menor</i>	<input type="checkbox"/>
<i>O esforço feito pelo utilizador é maior</i>	<input type="checkbox"/>
<i>Cometem-se mais erros a executar a tarefa</i>	<input type="checkbox"/>
<i>Cometem-se menos erros a executar a tarefa</i>	<input type="checkbox"/>

<i>Gostaria de realizar as suas tarefas utilizando um sistema automatizado?</i>
<i>Sim</i> <input type="checkbox"/> <i>Não</i> <input type="checkbox"/>
<i>Porque?</i>

Modelo II – Análise do ambiente e do contexto de operação

<i>Responsável</i>	
--------------------	--

<i>Data da visita</i>		<i>Local</i>		<i>Hora</i>	
-----------------------	--	--------------	--	-------------	--

Tipo de iluminação

<i>Boa</i>	<input type="checkbox"/>
<i>Fraca</i>	<input type="checkbox"/>
<i>Media</i>	<input type="checkbox"/>

Ruído

<i>Não existe</i>	<input type="checkbox"/>
<i>Existe em pequena proporção</i>	<input type="checkbox"/>
<i>Existe em grande proporção</i>	<input type="checkbox"/>

Tecnologia Existente:

<i>Computadores</i>				
<i>Marca</i>	<i>Modelo</i>	<i>Numero</i>	<i>Especificações Técnicas</i>	<i>Estado</i>

<i>Outros dispositivos</i>		
<i>Tipo</i>	<i>Modelo</i>	<i>Estado</i>

<i>Disposição geral dos elementos no local</i>