

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Engenharia de Usabilidade

Material de Referência

Professor: Clarindo Isaías Pereira da Silva e Pádua

Belo Horizonte, MG
Agosto de 2012



SUMÁRIO

SUMÁRIO.....	2
1 Introdução	5
1.1 Apresentação	5
1.1.1 Objetivos da disciplina	5
1.1.2 conteúdo da disciplina	5
1.2 Introdução à usabilidade	7
1.2.1 Benefícios.....	9
1.3 Glossário.....	11
1.4 Bibliografia.....	12
2 Princípios de desenho.....	12
2.1 Ciclo de execução de uma ação	13
2.2 Distribuição da informação	15
2.3 "Psicologia" dos materiais/serventia	16
2.4 Visibilidade.....	18
2.5 Modelo mental	19
2.6 Mapeamentos	20
2.7 Realimentação.....	22
2.8 Atenção aos requisitos	23
2.9 Função de força	24
2.10 Bibliografia.....	25
3 Processo visando a usabilidade.....	25
3.1 Detalhes de implementação do praxis-u	27
3.2 Atividades do fluxo de usabilidade.....	27
3.2.1 Atividades gerenciais	29
3.2.2 Atividades técnicas.....	30
3.3 Glossário.....	37
3.4 Bibliografia.....	38
4 Análise de contexto de uso	39
4.1 Técnicas	40
4.1.1 Observação direta	41
4.1.2 Entrevistas	41
4.1.3 Levantamento por questionário	42
4.1.4 Fontes de informação	42
4.2 Subprocesso de Análise de contexto de uso	46
4.2.1 Visão geral	47
4.2.2 Planejamento	48
4.2.3 Preparação	51
4.2.4 Modelagem de usuários	52
4.2.5 Modelagem de tarefas	52
4.2.6 Análise de Produtos Concorrentes ou Similares	53
4.2.7 Balanço final.....	53
4.3 Análise de usuários.....	54
4.3.1 Visão geral	54
4.3.2 Objetivos.....	55
4.3.3 Personas	56
4.3.4 Detalhamento das atividades do subprocesso	70
4.4 Análise de tarefas	74
4.4.1 Visão geral	76

4.4.2	Objetivos.....	78
4.4.3	Roteiros de domínio de problema	79
4.4.4	Detalhamento das atividades do subprocesso de Análise de tarefas.....	82
4.4.5	Análise de necessidades	85
4.5	Análise de ambiente	88
4.6	Análise de produtos concorrentes ou similares	90
4.7	Glossário.....	91
4.8	Bibliografia.....	92
5	Especificação de requisitos de usabilidade	93
5.1	Visão geral.....	93
5.2	Tabela de especificação de requisitos de usabilidade	94
5.2.1	Atributos de Usabilidade	96
5.2.2	Autor	98
5.2.3	Instrumento de medida	98
5.2.4	Valor a ser medido	101
5.2.5	Níveis de desempenho.....	102
5.2.6	Diretrizes para definição da Tabela ERU	105
5.3	Glossário.....	106
5.4	Bibliografia.....	107
6	Desenho da Interação	107
6.1	Atividade de Desenho da interação	109
6.1.1	Modelagem conceitual	111
6.1.2	Modelagem de conteúdo e navegação	117
6.1.3	Desenho detalhado	118
6.2	Guia de Estilo de Usabilidade.....	119
6.3	Glossário.....	124
6.4	Bibliografia.....	124
7	Avaliação de usabilidade	124
7.1	Objetivos	125
7.2	Técnicas	127
7.2.1	Analíticas.....	128
7.2.2	Avaliações experimentais	129
7.2.3	Pesquisa de opinião.....	130
7.3	Subprocesso de avaliação de usabilidade	131
7.3.1	Visão geral	131
7.3.2	Detalhamento das atividades	135
7.4	Padrão de Descrição de Avaliação de Usabilidade	151
7.4.1	Descrição das Avaliações de Usabilidade	151
7.4.2	Relatório de Avaliações de Usabilidade.....	151
7.5	Glossário.....	152
7.6	Bibliografia.....	152
8	Diretrizes de usabilidade	153
8.1	Fundamentos de desenho	153
8.2	Princípios	154
8.3	Diretrizes	158
8.4	Glossário.....	168
8.5	Bibliografia.....	168
9	Elementos de Interação	169
9.1	Interfaces Gráficas.....	170
9.1.1	Janelas (windows).....	170
9.1.2	Menus (menu)	171
9.1.3	Formulários (Forms).....	178
9.1.4	Caixas (box)	181
9.2	Interfaces gráficas em sentido amplo	184

9.2.1	Visualização de dados.....	184
9.2.2	Bancos de dados visuais	185
9.2.3	Animações.....	186
9.2.4	Vídeo (e áudio).....	186
9.2.5	Realidade virtual	187
9.3	Linguagem de comandos.....	187
9.4	Outros elementos de interação	188
9.4.1	Tela de toque	188
9.4.2	Síntese da fala	189
9.4.3	Reconhecimento da fala e Linguagem natural.....	189
9.5	Glossário.....	190
9.6	Bibliografia.....	190
10	Glossário.....	191
11	Bibliografia	191

1 INTRODUÇÃO

1.1 APRESENTAÇÃO

Este documento foi desenvolvido para uso dos alunos da disciplina de Engenharia de Usabilidade. O conteúdo aqui abordado provê ao aluno o conhecimento básico necessário ao projeto da interação do usuário em um sistema de software, sendo este conteúdo contextualizado em um processo de desenvolvimento de software visando à usabilidade.

1.1.1 OBJETIVOS DA DISCIPLINA

A disciplina de Engenharia de Usabilidade tem por objetivo apresentar técnicas, conceitos e métodos que podem ser utilizados sistematicamente para assegurar um alto grau de usabilidade na interface final de programas de computador. Usabilidade refere-se à qualidade da interação usuário-computador proporcionada pela interface de um sistema de computação. Os benefícios alcançados pela aplicação de técnicas da engenharia de usabilidade são visíveis tanto no aspecto de eficiência e eficácia da interface como também se expressam em processos de desenvolvimento de software mais produtivos, confiáveis e com maior satisfação dos usuários e clientes.

O projeto de interfaces do usuário é um dos principais objetivos da Engenharia de Usabilidade. O conteúdo da disciplina inclui tanto o produto, a própria interface, como o processo, compreendendo metodologia e técnicas usadas dentro de um ciclo de desenvolvimento de software. Uma abordagem de Engenharia será adotada, no sentido de que as técnicas utilizadas serão baseadas em métodos onde utilizamos quantificações visando estabelecer parâmetros para avaliação de aspectos subjetivos.

Ao final do curso o aluno deverá conhecer as principais técnicas utilizadas no projeto de interface com o usuário visando à usabilidade e ser capaz de aplicar essas técnicas no contexto de um processo de desenvolvimento de software.

1.1.2 CONTEÚDO DA DISCIPLINA

Para conceituar Usabilidade, podemos utilizar as normas ISO 9241 (ISO 92141, 1998), que trata de requisitos ergonômicos de trabalhos em escritório com terminais visuais, e ISO/IEC 9126 (ISO/IEC 9126), que trata da qualidade de produtos de software. Segundo a norma ISO 9241, usabilidade é a “capacidade que um sistema interativo oferece a seu usuário, em um determinado contexto de operação, para a realização de tarefas de maneira eficaz, eficiente e agradável”. Já segundo a norma ISO/IEC 9126, usabilidade é a “facilidade com que um usuário pode aprender a operar, preparar entradas para e interpretar as saídas de um sistema ou componente”.

Simplificando, podemos dizer que a usabilidade está associada a uma característica de qualidade de software que se refere à sua adequação à utilização pelos usuários. A usabilidade trata da qualidade da interação usuário-computador proporcionada pela interface de um sistema de computação. É importante salientar que a usabilidade está sempre associada a um contexto de utilização do produto; a adequação ao uso significa adequação ao tipo de tarefas ou atividades que se pretende realizar com o produto de software, ao tipo de usuários que tipicamente utiliza o produto e ao ambiente de utilização do produto.

Segundo o dicionário Aurélio, engenharia é a “arte de aplicar conhecimentos científicos e empíricos e certas habilidades específicas à criação de estruturas, dispositivos e processos que se utilizam para converter recursos naturais em formas adequadas ao atendimento das necessidades humanas”. Neste documento, uma abordagem de Engenharia será adotada, no sentido de que serão apresentados processos e técnicas relacionadas ao desenvolvimento da interação, buscando-se a quantificação de resultados para que se possa executar o processo de forma controlada. O termo “desenvolvimento da interação” refere-se ao desenvolvimento da interface com o usuário nos aspectos relacionados à usabilidade.

A Engenharia de Usabilidade visa o desenvolvimento da interação entre o usuário e sistemas informatizados. A engenharia de usabilidade tem por objetivo oferecer técnicas e métodos que possam ser utilizadas sistematicamente para assegurar um alto grau de qualidade em termos de usabilidade da interface de programas de computador.

A fronteira entre a engenharia de software e a engenharia de usabilidade, como acontece entre várias outras áreas do conhecimento, às vezes não é muito nítida, mas, em linhas gerais, pode-se dizer que engenharia de software cuida dos aspectos construcionais de sistemas de software enquanto a usabilidade trata dos aspectos comportamentais, relacionados à interação humano-computador.

Para se conseguir o desenvolvimento de um software de boa qualidade em termos de usabilidade é necessária a realização de uma série de atividades que podem ser organizadas como um processo que deve se integrar ao processo utilizado para o desenvolvimento dos aspectos construcionais do software. O processo de desenvolvimento é um aspecto importante quando se trata de usabilidade. Por este motivo, este documento apresenta a engenharia de usabilidade no contexto de um processo, denominado Praxis-u, que vem sendo desenvolvido no Departamento de Ciência da Computação da UFMG, visando o desenvolvimento da interação usuário-computador.

O processo Praxis-u aqui apresentado foi desenvolvido de forma a se integrar com o Praxis (Padua, 2003), abordando aspectos relacionados à usabilidade. O Praxis é um processo desenhado para dar suporte a projetos didáticos em disciplinas de engenharia de software e em programas de capacitação profissional em processos de software. Apesar de se integrar ao Praxis, o Praxis_u pode ser utilizado de forma independente no desenvolvimento da interação usuário-computador ou mesmo integrado a outros processos de desenvolvimento de software.

1.2 INTRODUÇÃO À USABILIDADE

Os benefícios alcançados pela aplicação de técnicas da engenharia de usabilidade são visíveis tanto no aspecto de eficiência e eficácia da interface como também se expressam em processos de desenvolvimento de software mais produtivos, confiáveis e com maior satisfação dos usuários e clientes. Dependendo do tipo de produto, a utilização de técnicas de usabilidade pode ser imprescindível para seu sucesso ou pode resultar em um importante diferencial visando à competitividade. Por esses motivos, o desenvolvimento de métodos e práticas de engenharia que assegurem uma eficiente interação computador-usuário vem tendo uma importância crescente no desenvolvimento de software.

Segundo Jakob Nielsen (Nielsen, 1993), um pesquisador reconhecido e precursor na área de usabilidade, a engenharia de usabilidade visa o desenvolvimento de interfaces com os seguintes atributos:

- **Produtividade na realização de atividades:** a interface deve permitir bom desempenho do usuário na realização de suas tarefas. Não se está falando de desempenho do software, que é um atributo de qualidade utilizado na engenharia de

software, mas do desempenho do usuário em sua interação com um sistema de software.

- **Facilidade de aprendizado:** deve ser fácil para o usuário aprender a utilizar o software.
- **Retenção do aprendizado com uso intermitente:** a interface deve permitir que o usuário (esporádico) consiga utilizar o software adequadamente mesmo quando fica sem usá-lo por um período relativamente longo de tempo.
- **Prevenção de erros do usuário:** o sistema deve prevenir erros do usuário quando o utiliza em suas atividades. Cabe observar aqui também que não se está falando de erros no programa, mas sim de erros do usuário ao utilizar o sistema.
- **Satisfação: o usuário deve gostar de utilizar o sistema.** Observem que a satisfação é um aspecto subjetivo, pessoal, mas ainda assim importante e que deve ser buscado no desenvolvimento de um produto de software.

Cabe observar que, ainda que os cinco atributos de Nielsen sejam desejados, nem sempre se consegue contemplar todos eles em uma interface do usuário, ou, dependendo do contexto de utilização do produto, um ou outro atributo podem se tornar prioritário. Por exemplo, em um sistema usado no caixa de um banco, onde pode haver uma fila de pessoas para serem atendidas e o tempo de atendimento é um aspecto crítico, o atributo Produtividade e provavelmente “Prevenção de erros” tornam-se muito importante. É admissível inclusive sacrificar o atributo “Facilidade de aprendizado”; pode-se treinar o funcionário do banco por certo tempo para que ele adquira destreza na operação do software. Já em um software como um sítio de comércio eletrônico, onde se espera que um amplo perfil de usuários consiga utilizá-lo, os atributos “Facilidade de aprendizado” e “Satisfação” tornam-se prioritário. Note, então, que Usabilidade não significa “facilidade de uso” de um produto de software como muitos acreditam, mas sim adequação ao uso.

Além dos cinco atributos de Nielsen, outros podem ser importantes dependendo do contexto de utilização do software. Por exemplo, em um sistema de apoio ao diagnóstico médico, a precisão, ou acurácia, dos resultados pode ser um atributo muito importante a se buscar no desenvolvimento da interface.

1.2.1 BENEFÍCIOS

Um investimento em usabilidade pode trazer diversos benefícios para as partes envolvidas no desenvolvimento de um produto de software. Podemos agrupar os benefícios em três categorias:

- Organização responsável pelo desenvolvimento do software.
- Cliente contratante de um desenvolvimento de software
- Usuário do produto a ser desenvolvido

Esses benefícios advêm não só da qualidade do produto, mas também da utilização de técnicas que tornam o processo de desenvolvimento mais eficiente e efetivo.

Em termos de benefícios de negócio para a organização desenvolvedora podemos citar:

- Diminuição de custos e tempo de desenvolvimento.
- Satisfação do cliente.
- Melhoria em credibilidade no mercado.
- Diminuição de riscos de projeto.
- Melhoria radical de chances de sucesso no mercado.
- Maiores vendas: produto tem melhor aceitação já que são mais intuitivos de se usar, mais rápidos e mais efetivos.

A gerência da equipe de desenvolvimento do software também se beneficia da utilização de um processo adequado que visa à usabilidade. Isso se deve principalmente às técnicas relacionadas à usabilidade que são utilizadas no processo de desenvolvimento. Segue lista de benefícios para a gerência da equipe de desenvolvimento.

- Melhora a gerência de riscos já que alternativas de desenho são testadas e melhoradas muito antes que a codificação prossiga.
- Simplifica o planejamento: permite o cálculo mais preciso de necessidade de esforço já que reduz drasticamente a necessidade de re-trabalho devido a desenhos não satisfatórios e problemas de comunicação com o usuário
- Provê evidências de sucesso mais cedo: as avaliações e relatórios com definições de requisitos de usabilidade e registros em vídeos confirmam a validade dos desenhos ainda em estágios iniciais de desenvolvimento.

As técnicas que visam à usabilidade também ajudam no processo de desenvolvimento do software, como mostrado a seguir.

- Gera confiança em que o desenho funciona: usuários reais validam o desenho muito antes que ele seja construído.
- Propicia teste de múltiplos conceitos rapidamente: torna mais fácil e rápido tentar várias soluções de desenho para verificar-se qual a melhor.
- Evitam-se alterações de última hora, com o stress associado aos atropelos e esforço concentrado de última hora.
- Diminui-se o stress associado aos testes de aceitação. Como as soluções de desenho são bem testadas antes de sua implementação, os testes de aceitação tornam-se tarefas muito mais suaves.
- Pode levar a desenhos mais acurados: com os diversos aspectos da interação modelados e documentados, pode-se obter um quadro mais acurado do produto a ser construído. Isso porque a análise do contexto de uso do produto em desenvolvimento leva a uma visão mais acurada e documentada de como os usuários trabalham, sem suposições não fundamentadas de como os usuários vão usar a interface.
- A utilização de protótipos e sua validação mais cedo no ciclo de desenvolvimento do produto propiciam que a documentação do produto também possa se iniciar mais cedo, com mais tempo para correções e para se produzir todos os aspectos envolvendo documentação, help e treinamento. Isso também facilita a identificação de defeitos e a correção de falhas potenciais no desenho da interface do usuário antes da construção.
- Interfaces mais intuitivas e de mais fácil utilização podem diminuir a necessidade de documentação e de material de suporte.
- Permite o planejamento de testes mais cedo e com mais tempo, partindo-se dos desenhos documentados.
- Facilita a identificação de erros: com os testes de usabilidade cobrindo grande parte das interações possíveis, fica mais fácil a identificação de problemas.

Quanto ao cliente contratante do desenvolvimento de software, podemos listar os seguintes benefícios.

- Mais segurança no produto, a partir das evidências oriundas dos testes e da prototipagem, com a confiança de que o produto foi desenhado para suprir suas necessidades.

- Melhora a produtividade do trabalho de seus usuários utilizando os produtos desenvolvidos, que tendem a ser mais rápidos e provendo navegação de maior qualidade.
- Diminuição dos custos de propriedade do produto, incluindo menor necessidade de treinamento e de infra-estrutura de suporte. O custo de aquisição é apenas parte do custo de propriedade de um produto. O custo da propriedade inclui também os custos de operação, de treinamento de pessoal que vai utilizá-lo, de manutenção, etc.
- Diminuição do risco de ter que trocar de produto por não atender às suas necessidades.

Finalmente, podemos identificar benefícios para os usuários do produto em desenvolvimento. Cabe observar que as pessoas que utilizam o software costumam também ser ouvidas e têm um impacto enorme na decisão de sua compra e na compra de versões futuras do produto.

- Facilidade de uso e de aprendizado.
- Usuário pode trabalhar de maneira mais rápida e agradável, com uma ferramenta mais adequada às suas necessidades.
- Menos tempo “perdido” lendo manuais ou *Helps* e consultando o suporte, com mais tempo sendo produtivo.
- Menos stress na utilização já que o produto terá sido construído em torno das necessidades dos usuários e usando sua terminologia e conceitos.

1.3 GLOSSÁRIO

PROCESSO

Processo é um conjunto de passos parcialmente ordenados, cujo objetivo é atingir uma meta. No caso de processo de desenvolvimento de software a meta é entregar um produto de software de maneira eficiente, previsível e que atinja as necessidades de negócio (Wikipedia n.d.).

PROTÓTIPO

Protótipo é uma versão parcial de um produto desenvolvida com um mínimo de custo com o objetivo de prover uma visão mais concreta de algum aspecto do produto antes de sua conclusão. O protótipo geralmente é usado para validação antecipada (antes da conclusão)

de algum aspecto do produto. Em usabilidade, o protótipo de interface é muito utilizado para validação com os usuários do software em desenvolvimento.

1.4 BIBLIOGRAFIA

ISO 9241-11 Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability 1998.

ISO/IEC 9126 Information technology – software product quality- part 1: quality model 1999, (FDIS).

Nielsen, J. Usability Engineering. Chestnut Hill, MA, Academic Press, 1993. Livro clássico, precursor, sobre usabilidade.

PADUA, W. *Engenharia de Software: Fundamentos, Métodos e Padrões*. 2. ed. Rio de Janeiro: Editora LTC, 2003. v. 1. 604 p.

Wikipedia, <http://pt.wikipedia.org/wiki/processo> em 10/2009

2 PRINCÍPIOS DE DESENHO

Uma série de princípios que se aplicam ao desenho (design) de objetos de modo geral também se aplica ao desenho de interfaces do usuário. O livro de Norman (Norman, 1998), muito citado em várias referências sobre usabilidade, aborda este tema de uma maneira muito instigante, constituindo uma leitura quase obrigatória para quem se interessa pelo tema de usabilidade. Donald A. Norman é professor de ciência cognitiva na Universidade da Califórnia em San Diego e Professor de Ciência da Computação na Universidade Northwestern, mas seus trabalhos de hoje são na maioria na Engenharia de Usabilidade. O conteúdo deste capítulo é baseado nesta obra de Norman.

MOTIVAÇÃO

Em nosso dia-a-dia, freqüentemente nos deparamos com dificuldade na utilização de algum objeto. Por exemplo, portas que tentamos abrir puxando quando o correto seria empurrando, equipamentos eletrônicos que não conseguimos operar adequadamente, painéis cheios de botões e controles que não conseguimos manejar, etc. Costumamos

“culpar” a nós mesmos pela dificuldade em operar esses equipamentos, mas, na verdade, muitos desses problemas poderiam ser evitados através de um desenho adequado dos objetos. É fácil reconhecer que se um objeto foi desenhado para ser usado por determinado perfil de usuário e se os usuários com esse perfil não conseguem utilizar adequadamente o objeto, então o problema não está nos usuários, mas sim no projetista que não conseguiu realizar um desenho adequado do objeto. O objetivo do desenho, tornar o objeto acessível aos seus usuários, não foi atingido.

Muitos desses problemas poderiam ser evitados se alguns princípios fossem conhecidos do projetista e utilizados no desenho do objeto. O mesmo se aplica ao desenho de interfaces de usuários e muitos problemas de dificuldade de acesso a sítios web poderiam ser evitados pela observância, pelos desenvolvedores, de alguns princípios básicos de design.

Nas seções seguintes apresentaremos vários desses princípios que podem ser usados como base para o entendimento do processo de interação do usuário e para o desenho de interfaces mais adequadas à interação.

2.1 CICLO DE EXECUÇÃO DE UMA AÇÃO

Segundo Norman (Norman, 1998), toda ação que realizamos passa por sete estados que vão desde a formação de um objetivo que buscaremos atingir através da ação até a conclusão dessa ação. Para que a ação possa ser bem executada, isto é, para que um indivíduo consiga interagir de forma satisfatória com o ambiente onde realiza suas atividades, é necessário que este indivíduo tenha condições de atingir os sete estados envolvidos na realização da ação. A realização de qualquer atividade, portanto, envolve esses vários estados onde continuamente executamos ações e avaliamos o que estamos fazendo.

A Figura 2-1 mostra um ciclo de estados que ocorrem quando interagimos em um ambiente na realização de uma ação. Esse ciclo pode ser caracterizado pelos estados ou etapas mostradas a seguir. Para ilustração, vamos imaginar o cenário que uma pessoa, Maria, que vai pintar as paredes de sua sala de visitas.

- Formação do objetivo: tudo se inicia com a formação de um objetivo. Esse objetivo pode ser de vários tipos, por exemplo, lazer, uma atividade do dia-a-dia, uma atividade profissional, a interação com um sítio Web, etc. Em nosso exemplo, Maria está em sua sala, pensando na vida, e, de repente, se sente aborrecida com a aparência de sua sala e pensa que seria bom mudar a cor de sua pintura.

A partir da formação de um objetivo, a realização de uma ação envolve aspectos relacionados à execução da atividade e aspectos relacionados à avaliação do está sendo executado para verificar se está tudo indo bem ou se é necessário alguma correção ou ajuste no que está sendo feito tendo em vista o objetivo definido.

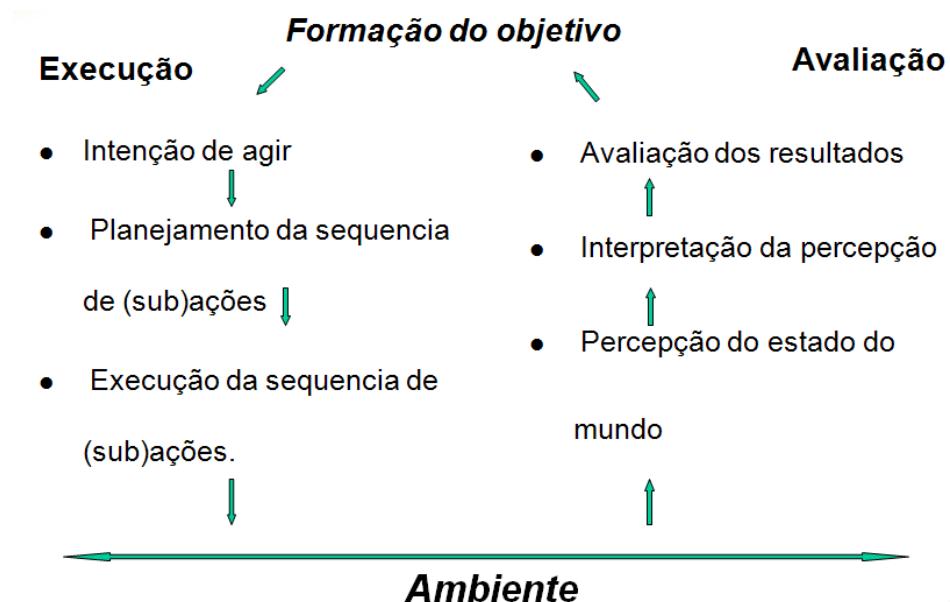


Figura 2-1 Ciclo da interação: os sete estados da ação (adaptado de Norman, 1998)

A execução da ação envolve os seguintes estados:

- Intenção de agir: muitas vezes temos vontade de fazer alguma coisa, mas não agimos realmente para atingir nosso objetivo. A realização de uma ação começa realmente quando tomamos a decisão de agir. Maria decide que vai pintar as paredes de sua sala de visita.
- Planejamento de (sub) ações: planejamos nossas ações ou “subações” se considerarmos ações mais complexas como formadas por subações. Esse planejamento pode até ser feito de forma não consciente, mas qualquer sequencia de ações tem que ser planejada antes por nosso cérebro. Maria pensa em comprar tinta e outros materiais, reserva um dia para a pintura e planeja outras providências.
- Execução da sequencia de (sub) ações: a ação é efetivamente executada. Maria pinta as paredes de sua sala.

Já a avaliação da ação envolve os seguintes estados:

- Percepção do estado do mundo: quando realizamos uma ação, temos a necessidade de continuamente perceber o que estamos fazendo. A percepção se dá através qualquer de nossos cinco sentidos, e qualquer dificuldade na percepção pode ser um obstáculo à realização de uma ação. Maria se utiliza principalmente da visão, mas provavelmente também do tato e olfato, para perceber o resultado de sua atividade.
- Interpretação da percepção: precisamos ser capazes de interpretar o que estamos percebendo tendo em vista nossos objetivos. Em nosso exemplo, Maria vai interpretar que há alguma coisa errada na rugosidade que esta percebendo em sua pintura.
- Avaliação dos resultados: para atingir nossos objetivos, é importante também a capacidade de avaliar o que foi percebido e interpretado. Neste estado, avaliamos se o que estamos fazendo está de acordo com nossas expectativas e, caso necessário, fazemos correções visando nossos objetivos. Maria avalia a rugosidade que está observando em sua pintura. Considera se que a causa disso pode ser não ter lixado a parede antes ou que a tinta requeira uma maior diluição. Depois de ler as instruções na embalagem da tinta, chega à conclusão que usou pouco diluente e que vai ter que refazer a pintura da parede que estava realizando.

Contextualizando no desenho da interação no desenvolvimento de software, podemos observar que um usuário vai conseguir utilizar adequadamente um software se sua interface lhe capacita a passar pelos sete estados envolvidos na realização de uma ação e vai ter dificuldades ou mesmo utilizar o software de uma maneira não efetiva, caso não consiga agir adequadamente como se espera em quaisquer desses estados.

É interessante observar também que esse ciclo de estados da ação está na origem de vários princípios de desenho que são apresentados adiante no texto. Por exemplo, a realimentação (feedback). Na utilização de um software, um usuário necessita uma realimentação contínua e completa sobre os resultados de suas ações. Uma realimentação adequada vai permitir ao usuário perceber, interpretar e avaliar o resultado de suas ações.

2.2 DISTRIBUIÇÃO DA INFORMAÇÃO

O comportamento de uma pessoa quando interage em um ambiente é guiado pela combinação do conhecimento que a pessoa tem e da informação que capturam do mundo. Um comportamento adequado do usuário pode se dar mesmo com pouco conhecimento do

ambiente com o qual interage, por que ele se utiliza também da informação que está no “mundo”, ou seja, no ambiente com o qual está interagindo.

Para ilustrar esse aspecto, imagina que você mora em uma grande cidade e vai de sua casa até uma rua afastada que você conheça no extremo oposto da cidade. Você consegue, sem muita dificuldade, chegar ao seu destino, afinal você já foi lá várias vezes. Mas imagina se você for explicar a uma pessoa que não conheça sua cidade com fazer esse trajeto.

Provavelmente, essa pessoa não conseguirá repetir esse trajeto a partir de sua explicação somente. Por que isso acontece? Afinal, se você tem a informação sobre o caminho você poderia passá-la à outra pessoa. Isso acontece por que não nos guiamos somente pela informação que temos “na cabeça”, memorizada, mas sim pela combinação do que temos memorizado com a informação que está no mundo. E um comportamento “perfeito” desejado pode ser alcançado mesmo com informações imprecisas pelas seguintes razões.

- Informação no mundo. O mundo nos apresenta muita informação que nos ajuda em nossa navegação. Por exemplo, placas, avisos e objetos que chamam nossa atenção.
- Um comportamento perfeito resultará se o conhecimento que temos fornece a informação ou comportamento que seja suficiente para se distinguir a escolha correta dentre todas as outras. Não é necessário conhecer todas as alternativas; apenas aquelas relacionadas ao que estamos fazendo.
- Restrições naturais estão presentes. O mundo restringe os comportamentos admissíveis. As próprias propriedades físicas dos objetos restringem as operações que podemos executar. Por exemplo, uma cidade tem ruas e avenidas que restringem e facilitam nossa “navegação”.
- Restrições culturais estão presentes. A sociedade cria inúmeras convenções artificiais que foram aceitas e incorporadas em nosso comportamento. Essas convenções podem ser utilizadas pelo projetista de um equipamento para facilitar sua utilização. Por exemplo, leis de trânsito podem restringir, mas também facilitar nossa movimentação em uma cidade.

2.3 “PSICOLOGIA” DOS MATERIAIS/SERVENTIA

As pessoas têm diferentes reações frente aos objetos com que estão interagindo. Essa reação muitas vezes depende do tipo de material. É como se houvesse uma psicologia das pessoas em relação aos materiais e objetos. Por exemplo, uma tesoura serve para cortar,

um papel serve para se escrever ou para embrulhar coisas. Madeira é associada com solidez, opacidade e serve para suportar coisas ou para se entalhar. Dependendo da forma, um botão serve para se girar, se pressionar ou se mover como em um *joystick*. Bolas servem para se atirar ou jogar; placas para se empurrar; ranhuras para se encaixar coisas, etc.

Podemos identificar para que serve um objeto por meio de suas características percebidas e reais. Nossa aprendizado diário de como funcionam os objetos e para que servem ocorre a partir de informações apreendidas pela aparência dos objetos (a psicologia dos objetos) e pela forma com que o designer que o projetou nos apresenta a operação do objeto.

No desenho de um objeto, o projetista pode explorar a noção de serventia das pessoas para dar indicações sobre operação ou utilização. A vantagem dessa abordagem é que o usuário entende o que fazer com aquele objeto apenas através do olhar. Muitas vezes, não são necessárias instruções. Cabe aqui observar que quando coisas simples necessitam de instruções é uma boa indicação de que o design apresenta falhas.

A primeira imagem na **Figura 2-2** mostra o exemplo de um bom desenho: só de olhar para a maçaneta de um veículo, pela sua forma podemos identificar se a maçaneta é de puxar ou de empurrar. A segunda imagem mostra o desenho de uma maçaneta que necessitou de um rótulo para tornar mais claro como operar manipulador de uma porta.

Bom desenho



Solução não-tão-boa: recomenda-se rótulo

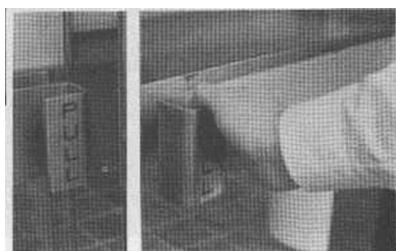


Figura 2-2 Maçanetas de veículos

Já a Figura 2-3 apresenta um desenho de uma maçaneta com qualidade questionável já que sua forma não dá uma boa indicação sobre se ela deve ser girada ou puxada/empurrada; além disso, sua forma arredondada e polida é escorregadia, dificultando sua manipulação. Talvez por isso tenham envolvido a maçaneta em fita adesiva!



Figura 2-3 Desenho questionável

2.4 VISIBILIDADE

O usuário deve saber dizer o estado em que se encontra um dispositivo e as opções de ação a partir daquele estado apenas olhando para ele. Visibilidade é a capacidade de uma interface de informar ao seu usuário sobre as ações disponíveis em determinado momento ou sobre o estado em que se encontra o equipamento ou sistema de software.

Um bom contraexemplo são os telefones do tipo PABX, utilizado para gerenciar vários ramais em uma empresa (Figura 2.4 Telefone PABX), que muitas vezes proveem um grande número de funções mas não tornam visíveis quais são as funções disponíveis para os usuários. Muitas vezes esses aparelhos não oferecem nem mesmo uma tela com a informação necessária aos usuários e as funções tem que ser comandadas através de códigos numéricos. O resultado disso é que em geral várias das funções acabam não sendo utilizadas seja porque o usuário não sabe que elas existem ou por que já se esqueceram, não sabem mais como utilizá-las e não estão dispostos a consultar o manual.



Figura 2-4 Telefone PABx

2.5 MODELO MENTAL

Modelo conceitual ou modelo mental é o modelo que formamos para explicar o funcionamento de um objeto. Para utilizar o objeto recorremos ao modelo mental que temos do modo como ele funciona. Por exemplo, temos um modelo mental de como funciona a caixa de marcha de um veículo e utilizamos esse modelo para passar as marchas de forma adequada quando dirigimos.

Para um objeto ser utilizado adequadamente, não necessitamos de um modelo conceitual detalhado nem mesmo de um modelo muito fiel ao modelo real. O importante é que nosso modelo mental seja consistente com o modelo real. Não necessitamos saber, por exemplo, que a caixa de marchas de um veículo é constituída de engrenagens que reduzem o número de giro do motor e transmitem potência às rodas. É suficiente sabermos que o veículo tem cinco marchas, que a primeira deve ser usada para a partida, que as marchas mais “fortes” são usadas para subidas mais íngremes.

Modelos mentais são importantes por que o projetista de um equipamento ou interface deve ter claro que ele precisa transmitir ao seu usuário um modelo conceitual correto para que ele consiga usar adequadamente o equipamento. A Figura 2-5 apresenta um teclado piano real, um instrumento físico, e um produto de software, onde o modelo mental que o usuário, músico provavelmente, já possui foi explorado no desenho de sua interface. A interface fica de fácil utilização para o usuário músico que já conheça o instrumento físico.



Figura 2-5 Teclado: instrumento físico e interface de software

A Figura 2-6 também apresenta uma interface de software representando a parte de trás de um equipamento de som. A interface é tão realística que se podem mudar conexões arrastando-se os cabos com a utilização de mouses.



Figura 2-6 Teclado Implementado em Software

2.6 MAPEAMENTOS

Mapeamento é uma propriedade que se refere à determinação do relacionamento entre as ações e os resultados, entre os controles e seus efeitos, entre o estado do sistema e o que é visível. Um bom desenho de um objeto deve facilitar o mapeamento ao usuário. Bons mapeamentos tiram vantagens de analogias físicas e padrões culturais. Por exemplo, o

mapeamento entre o giro do volante de um carro e o acionamento das rodas. Também um controle em que um som mais alto significa “mais” alguma coisa, ou uma situação em que se movendo o controle para o alto o objeto move-se para o alto também. Outro bom exemplo seria o mapeamento entre os movimentos do mouse e do cursor em uma interface gráfica.

A Figura 2-7, que mostra detalhe de um fogão doméstico, ilustra a dificuldade do usuário em fazer o mapeamento entre um registro e seu respectivo queimador. Para facilitar o mapeamento, o fabricante utiliza pequenos ícones que mostram o posicionamento do queimador. Com o passar do tempo, com a limpeza, o desenho costuma se apagar e uma pessoa que não tenha familiaridade com o fogão, como uma visita, pode ter dificuldade em fazer o mapeamento. A Figura 2-8 apresenta duas soluções com um melhor desenho em termos de mapeamento.



Figura 2-7 mapeamento registro x queimador em um fogão

A Figura 2-9 mostra um equipamento que pode ser usado para controle de iluminação de grandes ambientes como em um teatro. É fácil observar que é importante que a interface desse tipo de equipamento permita um bom mapeamento ao usuário.

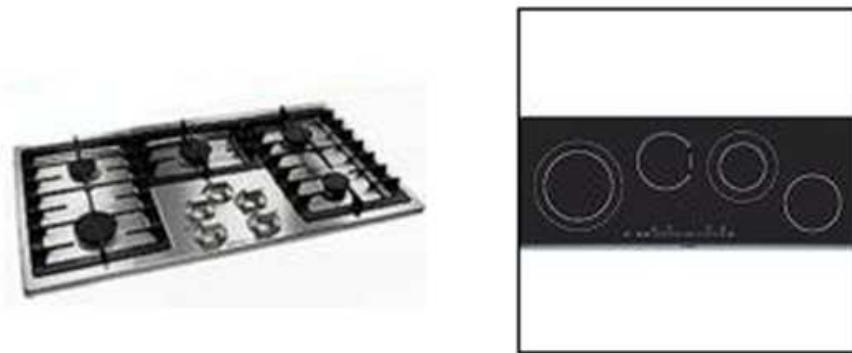


Figura 2-8 Fogões com melhor solução para o mapeamento



Figura 2-9 equipamento de controle de iluminação

2.7 REALIMENTAÇÃO

Na sua interação com o ambiente onde realizam suas atividades, as pessoas necessitam continuamente perceber, interpretar e avaliar o resultado de suas ações. Isso foi mostrado na seção 132.1 sobre o ciclo de execução de uma ação, nos estados associados à avaliação do que está sendo executado.

A cada ação realizada por um usuário, a interface deve enviar-lhe de volta informação sobre o que foi efetivamente realizado. O usuário deve receber uma realimentação contínua e completa sobre os resultados de suas ações.

O tipo de realimentação a ser dado por uma interface pode depender muito do tipo de situação. Em certas situações a realimentação deve ser mais visível. Por exemplo, em caso de alertas sobre situações em que a ação do usuário potencialmente pode causar algum tipo de dano.

Em outros casos, a realimentação visa informar ao usuário sobre o resultado de uma ação que ele realizou. Por exemplo, quando um usuário de um editor de texto comanda a substituição de todas as ocorrências da palavra "x" por "y", o sistema deve lhe dar uma realimentação sobre o resultado de sua ação. A realimentação poderia ser mais completa, por exemplo, mostrando cada substituição e, possivelmente, até solicitando permissão para cada substituição. Pode ser desejável também uma realimentação mais simples, mais rápida para o usuário. Por exemplo, o sistema poderia simplesmente informar ao usuário quantas substituições foram feitas e, com essa informação, o usuário busca avaliar se tudo ocorreu conforme sua expectativa.

A realimentação, dependendo do contexto, pode e deve ser sutil. Por exemplo, ao passarmos o cursor do mouse sobre a interface gráfica de um software, o sistema pode mudar frequentemente de estado. A informação sobre o estado do sistema é sutil, muitas vezes expressa somente pela forma do cursor. Essa realimentação do sistema informa ao usuário, de uma forma não agressiva, simples, porém eficaz, que o sistema está percebendo o movimento do mouse e que a cada contexto, correspondente às diferentes formas do cursor, diferentes ações podem estar disponíveis.

2.8 ATENÇÃO AOS REQUISITOS

Outro princípio de desenho mencionado por Norman (Norman, 1998) refere-se à atenção aos detalhes. Por exemplo, no desenho de um objeto (ou interface), considerar sempre quem seriam seus possíveis usuários, suas necessidades e as condições de uso. No processo de desenvolvimento que visa à usabilidade que veremos adiante, essa informação vem da atividade que chamamos de Análise de contexto de uso. Um dos objetivos da Análise de contexto de uso é justamente coletar informação sobre os perfis de usuários, suas necessidades, tarefas que realizam, sobre o ambiente de realização de suas atividades, dentre outras.

Um bom exemplo desse princípio poderia ser um o desenho do pen-driver. O pen-driver é prático, fácil de ser usado quase que por qualquer tipo de usuário, resistente, tem uma capa protetora e um conector USB que também só se encaixa se for colocado na posição correta.

2.9 FUNÇÃO DE FORÇA

Função de força é uma restrição à utilização de um objeto em determinadas circunstâncias.

Podemos observar três tipos de função de força:

- Intertravamento: acontece quando se força uma ordem de execução ou não permite operações que não devem ser executadas em dado momento. Por exemplo, quando uma console de jogo impede a retirada de um cartucho quando o equipamento está em operação, isto é, quando o usuário está jogando. A restrição visa proteger o equipamento de possíveis danos ou evitar que o sistema fique em um estado indefinido em determinada situação.
- “Lockin”: mantém uma operação ativa enquanto necessário. Por exemplo, o interruptor de energia não desliga o computador imediatamente já que poderia deixar o sistema operacional em um estado inconsistente. Nesse caso, o sistema operacional providencia um fechamento lógico (log off) antes do desligamento físico da fonte de energia.
- “Lockout”: impede o usuário de entrar em um local perigoso ou impede a ocorrência de um evento indesejado. Por exemplo, a Figura 2-10 mostra um elevador para residências ou pequenos prédios onde um mecanismo de *lockout* impede a abertura da porta quando o elevador não está posicionado no mesmo nível, prevenindo acidentes.



Figura 2-10 Elevador com mecanismo de lock-out. Fonte:
<http://www.dwa.eng.br/elevadores.html>

2.10 BIBLIOGRAFIA

<http://webmail.faac.unesp.br/~paula/Paula/everydaythings.pdf>

Norman, D. A. *The Design of Everyday Things*. New York, Basic Books, reimpresso em 1998.

3 PROCESSO VISANDO A USABILIDADE

Como já mencionado anteriormente, para conseguirmos um produto de software de boa qualidade em termos de usabilidade é necessário seguirmos um processo de desenvolvimento que inclua atividades que visem à usabilidade. Não é razoável, por exemplo, tentar desenhar uma interface do usuário, onde se busca qualidade em termos de usabilidade, sem o conhecimento de características importantes de seus usuários. A Análise de usuários e das tarefas que eles realizam, portanto, é uma atividade importante e que deve ser realizada antes do desenho da interface do usuário.

Um bom produto não surge por acaso, vai ser desenvolvido através de várias atividades que são definidas de uma maneira sistematizada em um processo. Um processo tem como objetivo uma meta. Aqui, estamos falando de um processo cuja meta é o desenvolvimento da interação humano-computador com qualidade em termos de usabilidade.

Neste capítulo apresentamos o Praxis-u, um processo de desenvolvimento de software visando à usabilidade que vem sendo desenvolvido no Departamento de Ciência da Computação da UFMG. O processo Praxis-u aqui apresentado foi projetado de forma a se integrar ao Praxis (Padua, 2003), abordando aspectos relacionados à usabilidade. No Praxis-u, as atividades que visam o desenvolvimento da interação usuário-computador foram organizadas em uma disciplina, ou subprocesso específico, o subprocesso de usabilidade. Como parte do Praxis-u, foram definidos artefatos e atividades que, integrados ao Praxis, constituem um processo de desenvolvimento de software visando à usabilidade. Apesar de se integrar ao Praxis, o Praxis_u pode ser utilizado de forma independente no desenvolvimento da interação usuário-computador ou mesmo integrado a outros processos de desenvolvimento de software.

Basicamente, o fluxo de usabilidade define um ciclo de atividades onde se analisa o problema, define metas de usabilidade, desenha soluções e avalia os resultados buscando a verificação das metas definidas. As soluções são inicialmente realizadas na forma de protótipos, no final chega-se ao desenho da interface do usuário em um processo iterativo.

A utilização de protótipos vem da necessidade de se avaliar soluções o mais cedo possível visando reduzir os custos das (inevitáveis) mudanças. As avaliações, em suas diversas formas, constituem uma atividade importante da engenharia de usabilidade. A usabilidade requer experimentação e avaliação porque há a necessidade de verificação da qualidade de soluções que envolvem a interação com o ser humano. Como não se consegue modelar efetivamente o comportamento do ser humano, devido a sua complexidade e variedade, as avaliações com a participação dos usuários são muito utilizadas para validação das soluções.

As avaliações de usabilidade visam à verificação da qualidade da interface, comparando-se os resultados obtidos com as metas definidas anteriormente, com o objetivo de se definir se o produto está concluído, com um nível aceitável de qualidade, ou se será necessário uma nova iteração pelo fluxo de usabilidade.

3.1 DETALHES DE IMPLEMENTAÇÃO DO PRAXIS-U

Um resumo do Praxis 2.0-u, incluindo seus artefatos e as atividades típicas de cada iteração, está disponível no sítio <http://www.dcc.ufmg.br/~clarindo/disciplinas/eu/material/index.htm>. Cabe observar que no desenvolvimento do Praxis-u, para vários artefatos que fazem parte do processo, havia duas possibilidades:

1. Estender os artefatos do Praxis com os aspectos de usabilidade;
2. Criar novos artefatos compreendendo os aspectos de usabilidade.

De maneira geral, a segunda abordagem foi a preferida, visando tornar a documentação dos aspectos relativos à usabilidade mais independente do resto do processo. Essa opção tem como justificativa objetivos didáticos, por facilitar o ensino, mas visou também facilitar a evolução e o uso dos processos de engenharia de software e de usabilidade de maneira mais independente. Em alguns casos, preferimos estender o artefato do Praxis com os aspectos de usabilidade; por exemplo, o modelo de cadastro de requisitos foi estendido com uma folha para registro dos requisitos de usabilidade.

3.2 ATIVIDADES DO FLUXO DE USABILIDADE

A Figura 3-1 apresenta um diagrama mostrando as principais atividades e artefatos utilizados no Praxis-u. As atividades do fluxo de usabilidade estão organizadas em duas raias, correspondentes a atividades técnicas e gerenciais. As atividades técnicas são de responsabilidade da equipe técnica de usabilidade e as atividades gerenciais são típicas da gerência de projeto referentes aos aspectos de usabilidade. Cabe observar que não estamos falando de uma gerência do projeto específica de usabilidade, normalmente isso não vai acontecer, mas sim de uma gerência de projeto que vai ser responsável também pela gerência do esforço associado à usabilidade.

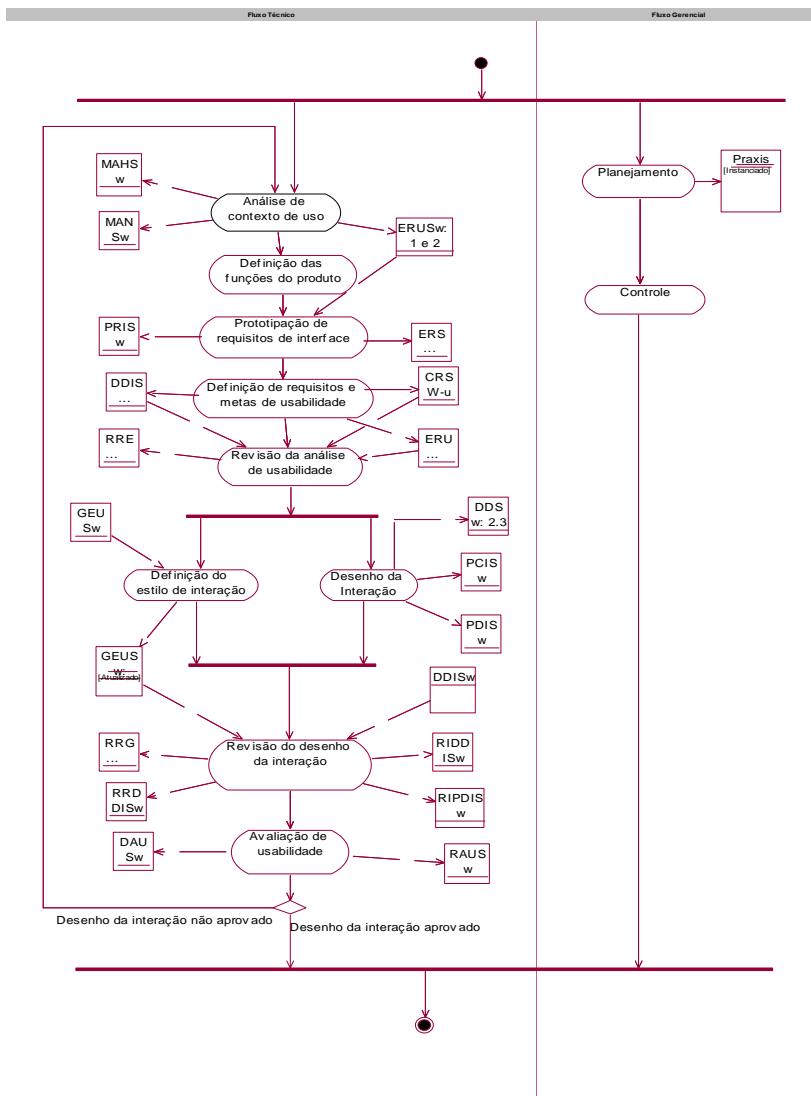


Figura 3-1 Praxis-u: diagrama de atividades

O diagrama utilizado na Figura 3-1 é um diagrama de atividades da UML (Booch, G. et al, 2005), (Rumbaugh, J. et al, 2004). Sendo assim, o retângulo com bordas arredondadas denota atividades e os outros retângulos denotam objetos, no caso, artefatos, que podem ser consumidos ou produzidos pelas atividades.

Tipicamente, durante o desenvolvimento de um produto de software, podem ser realizados vários ciclos utilizando o fluxo de usabilidade. Pode-se, inclusive, adotar o padrão em que um ciclo pelo fluxo de usabilidade seja realizado em cada iteração, sendo que em cada ciclo as atividades são realizadas com maior ou menor grau de detalhamento (podendo até ser omitida). No Praxis-u, um planejamento dos ciclos de usabilidade (quais e em quais iterações serão realizados) deverá ser feito na atividade de personalização do processo para projetos específicos.

As atividades, mostradas no diagrama da Figura 3-1 e listadas abaixo, serão apresentadas nas seções a seguir.

- Planejamento;
- Controle;
- Análise de contexto de uso;
- Definição das funções do produto;
- Prototipagem de requisitos de interface;
- Definição de requisitos e metas de usabilidade;
- Revisão da análise de usabilidade;
- Definição do estilo de interação;
- Desenho da interação;
- Revisão do desenho da interação;
- Avaliação de usabilidade e
- Balanço final.

3.2.1 ATIVIDADES GERENCIAIS

As atividades de Planejamento e Controle são atividades gerenciais; geralmente estão sob responsabilidade da gerência do projeto.

3.2.1.1 Planejamento

O Planejamento compreende a personalização do processo de desenvolvimento com relação aos aspectos de usabilidade e uma estimativa de recursos necessários ao cumprimento do

escopo do produto a ser desenvolvido. A personalização do processo visa à definição de uma instância do fluxo de usabilidade a ser utilizada em um projeto específico.

A estimativa de recursos visa à determinação do esforço (número de horas) e outros recursos necessários ao projeto, no que se relaciona às atividades que visam à usabilidade. A necessidade de recursos deve ser mapeada a marcos que indicam o progresso na evolução do escopo durante a execução do projeto. O mapeamento em marcos de progresso permitirá o acompanhamento (controle) do projeto durante sua execução, possibilitando a confrontação de metas atingidas de esforço, escopo, prazo e custo.

3.2.1.2 Controle

O Controle compreende o acompanhamento do progresso do projeto, durante sua realização, por meio da confrontação de metas de esforço, escopo, prazo e custo, comparando o previsto no Planejamento com o realizado até um determinado momento.

3.2.2 ATIVIDADES TÉCNICAS

3.2.2.1 Análise de contexto de uso

A Análise de contexto de uso tem como objetivo principal a caracterização dos usuários, das tarefas que eles realizam, de produtos semelhantes ou concorrentes e do ambiente onde será utilizado o produto em consideração. A Análise de contexto de uso produz informações que constituem importante insumo para várias atividades subsequentes no ciclo de desenvolvimento de software. Principalmente, essas informações são utilizadas para a definição do produto, para o desenho da interface com o usuário e para as avaliações de usabilidade.

Cabe esclarecer que o termo Análise aqui utilizado tem uma conotação diferente do termo usado na Engenharia de software. O termo Análise é usado na Engenharia de software para denotar as atividades que visam à criação do modelo de Análise. O modelo de Análise define o produto de software em nível de definição do problema, cuja solução em termos de sistema de software será definida no modelo de Desenho. Já o termo Análise de contexto de uso, usado na Usabilidade, refere-se à atividade que visa à modelagem de todo o contexto onde o produto de software será utilizado.

É importante ressaltar que o objetivo da Análise de contexto de uso é o de caracterizar a situação existente antes do desenvolvimento do software. O conhecimento obtido pela

Análise de contexto de uso, em um segundo momento, pode ser utilizado para se caracterizar as tarefas que se deseja informatizar e incorporar ao produto na forma de requisitos funcionais, e para se caracterizar os usuários alvos do software a ser desenvolvido. Não cabe, na Análise de contexto de uso, definir-se soluções de desenho da interface com o usuário, já que a Análise de contexto de uso visa justamente levantar informações importantes para o posterior desenho da interface. Deve-se, no entanto, como resultado do trabalho de Análise de contexto de uso, produzir-se recomendações para as atividades subsequentes no desenvolvimento do software, incluindo, com ênfase, recomendações para o desenho da interface com o usuário.

A Análise de contexto de uso pode ser considerada como um detalhamento da modelagem de processo de negócio nos aspectos que influenciam a usabilidade do produto. Por exemplo, a Análise de contexto de uso deve investigar as características das tarefas realizadas pelos usuários e a forma como essas tarefas são realizadas visando à obtenção de informação que vai ajudar, posteriormente, no desenho de uma interface mais adequada para a realização das mesmas tarefas com o apoio do sistema em desenvolvimento.

A Análise de contexto de uso, assim como outras atividades visando à usabilidade, tem certa interdependência com outras atividades da engenharia de software. A Análise de contexto de uso tem uma forte interdependência com o levantamento dos requisitos do software. Isso porque as funções que vão ser implementadas pelo software em desenvolvimento, seus requisitos funcionais, devem dar apoio aos usuários na realização das tarefas mais importantes (ou parte delas) que os usuários realizam.

A Análise de contexto de uso envolve diversos tipos de análise que serão apresentadas a seguir:

- Análise de usuários,
- Análise de tarefas
- Análise de Ambiente
- Análise de Produtos Concorrentes ou Similares |

O Praxis-u provê um gabarito (modelo no sentido de *template*) de um artefato, denominado ERUSw.dot, para a documentação da Análise de contexto de uso.

[S1] Comentário: Considerar colocar análise de modelos mentais e análise de necessidade como análise separadas e, juntamente com análise de ambiente, podendo ser ligado a usuários ou a tarefas

ANÁLISE DE USUÁRIOS

A análise dos usuários visa uma caracterização dos diversos perfis de usuários com relação a aspectos que interessam para o desenvolvimento do produto de software. A caracterização

dos usuários é feita em termos de um conjunto abstrato de necessidades, interesses, expectativas, comportamentos e responsabilidades dos diversos atores envolvidos na interação com o produto a ser desenvolvido.

A Análise de usuários combina resultados de teorias relacionadas com o processo cognitivo do ser humano (fatores humanos) com informações específicas sobre os usuários potenciais e o ambiente onde desenvolvem suas atividades. As informações sobre os usuários podem ser obtidas através da observação, de pesquisas de marketing, questionários e estudos observacionais do futuro local de implantação do sistema ou entrevistas com usuários e com especialistas no domínio de aplicação. Reuniões e grupos de discussão em que desenvolvedores e representantes dos usuários interagem para determinar as necessidades e características dos usuários também são freqüentemente utilizadas. O resultado da Análise de usuários é registrado em modelos próprios e documentado na especificação de requisitos de usabilidade.

Na realização de suas atividades no dia a dia, as pessoas criam e utilizam modelos mentais que explicam e ajudam no controle dos objetos com os quais interagem. Na seção 2.5 apresentamos modelos mentais associados a um princípio básico utilizado em *design*. Esses modelos mentais podem ser explorados no desenho da interface do usuário, tornando a interface mais intuitiva e de utilização mais fácil para os usuários. Sendo assim, a Análise de usuários deve incluir também uma análise dos modelos mentais mais importantes que as pessoas envolvidas (potenciais usuários) utilizam na realização de suas atividades.

ANÁLISE DE TAREFAS

Esta atividade tem como objetivo a caracterização das tarefas realizadas pelos usuários ou potenciais usuários em suas atividades relacionadas com o produto em consideração, aí incluindo a definição das necessidades que as tarefas visam suprir, o ambiente onde as tarefas são realizadas e a definição das tarefas que serão automatizadas ou realizadas pelo sistema.

As tarefas a serem realizadas pelo sistema, quando o produto estiver em operação em interação com os usuários, correspondem aos casos de uso. As outras tarefas realizadas pelos usuários são consideradas fora do escopo do produto a ser desenvolvido, mas a Análise de tarefas como um todo constitui informação muito importante para o desenvolvimento da interação humano-computador.

ANÁLISE DE AMBIENTE

As pessoas não trabalham ou exercem suas atividades isoladas do meio ambiente. O ambiente de realização das atividades pode ter muita influência na utilização do software e a incompatibilidade do ambiente com o software pode até resultar na rejeição do produto pelo usuário. A Análise de Ambiente visa uma caracterização do ambiente onde os usuários, ou potenciais usuários, realizam suas atividades relacionadas com o produto em consideração.

Cabe ressaltar que o ambiente de realização das atividades pode ser considerado como parte da Análise de tarefas assim como parte da Análise de usuários. A decisão sobre a melhor forma de se modelar o ambiente deve ser baseada no fato do ambiente estar mais associado às pessoas ou às atividades que elas realizam. Por exemplo, o risco de um ambiente exposto à radioatividade pode estar associado à tarefa de se tirar radiografias, para qualquer que seja o usuário envolvido nesta tarefa. Já o aspecto da pressão social por não se cometer erro pode ser modelado como associado ao usuário médico, independente das tarefas que ele realiza.

ANÁLISE DE PRODUTOS CONCORRENTES OU SIMILARES

A análise de concorrência visa o estudo de produtos semelhantes ao produto em consideração com o objetivo de:

- familiarização com o domínio do problema,
- identificação de oportunidades que possam diferenciar o produto.

Não se trata, obviamente, de cópia ou violação de direitos de propriedade, mas de conhecer os possíveis concorrentes ou produtos similares ao que está sendo desenvolvido com o objetivo de identificar oportunidades de diferenciais, conhecer limitações ou mesmo utilizar como uma referência para comparações.

3.2.2 Definição das funções do produto

Um dos objetivos das atividades de Análise de contexto de uso é a definição de quais tarefas, dentre as identificadas, serão realizadas ou apoiadas pelo produto em perspectiva. Os requisitos funcionais de um produto de software definem as funções que serão providas pelo produto para apoiar a realização dessas tarefas. Portanto, é importante que as pessoas que vão definir os requisitos funcionais tenham conhecimento das tarefas que o produto apoiará, ou seja, que tenha participado da Análise de contexto de uso.

A atividade de Definição das funções do produto fica na fronteira das áreas de engenharia de software e usabilidade. Em uma abordagem muito comum na engenharia de software, a

definição dos casos de uso é feita simplesmente a partir de informações levadas pelos usuários em reuniões de levantamento de requisitos. Acontece que os usuários, muitas vezes, têm uma visão mais localizada com relação às necessidades a serem cobertas pelo produto em consideração. A Análise de contexto de uso pode fornecer uma visão mais global das questões envolvidas na definição e priorização das funções a serem providas pelo produto.

A definição de quais funções (ou casos de uso) devem ser incorporados em um software deve ser feita tendo com base na Análise de tarefas que os usuários realizam. As tarefas caracterizadas na Análise de contexto de uso podem ser consideradas como candidatas a serem contempladas como funções (casos de uso) do produto em consideração. Em outras palavras, pode-se dizer que dentre as tarefas caracterizadas na Análise de contexto de uso, algumas, ou partes delas, vão poder ser realizadas pelos usuários com o apoio do produto quando ele estiver concluído. A Análise de tarefas permite que a definição dos casos de uso do produto seja feita com base em informações mais consistentes e com uma melhor visão do problema.

3.2.2.3 Prototipagem de requisitos de interface

O objetivo desta atividade é a criação de um protótipo para validação dos requisitos de interface com os usuários. Requisitos de interface são, basicamente, requisitos de entrada e saída de informação associados aos casos de uso. Em geral, esses requisitos são representados na forma de telas ou mesmo de tabelas, sem preocupação com os aspectos de design.

É importante não confundir os requisitos de interface com o desenho da interface. O Desenho será feito em uma etapa posterior quando, aí sim, aspectos de design como design gráfico, metáfora utilizada, organização de conteúdo, navegação, etc, serão considerados.

O Protótipo de requisitos de interface tem o objetivo de prover uma visão mais concreta dos requisitos de interface para que seja utilizado na validação com os usuários e cliente. O Praxis-u prevê um artefato denominado PRI como Protótipo de Requisitos de Interface. O PRI comprehende os aspectos de conteúdo e de navegação da interface, entendidos como requisitos de interação. O PRI pode ser registrado no documento de Especificação de Requisitos de Software do Praxis, na seção correspondente ao requisito de interface externa.

Apesar de sugerir o desenvolvimento do PRI, o Praxis-u não especifica um gabarito para este artefato. Isso porque se preferiu deixar livre para a organização desenvolvedora do software

definir a melhor maneira de se realizar a prototipagem. Existem inúmeras possibilidades que vão desde um documento como a própria especificação de requisitos ou uma apresentação (como um Power Point), até uma solução mais sofisticada como um produto específico contendo navegação de forma funcional.

3.2.2.4 Definição de requisitos e metas de usabilidade

Essa atividade visa à definição de níveis de desempenho almejados para os atributos de usabilidade considerados importantes para o produto em consideração. Sem especificações mensuráveis, é impossível definir-se parâmetros de qualidade em termos de usabilidade e dizer se o produto final alcança o nível de qualidade desejada. Com o estabelecimento dos requisitos e ou metas de Usabilidade o mais cedo possível no processo de desenvolvimento, e monitorando-as a cada iteração, pode-se determinar quando a interface está, de fato, melhorando em qualidade.

Quando envolve medidas objetivas, como, por exemplo, o tempo que o usuário gasta para realizar uma tarefa, os níveis de desempenho são definidos para Tarefas de referência (benchmark), ou seja, define-se o desempenho esperado dos usuários na realização de tarefas de referência. Quando envolve aspectos subjetivos como Satisfação do usuário, questionários podem ser utilizados. Por exemplo, pode-se definir como requisito de usabilidade que o produto de software deverá ser avaliado com uma nota média mínima de 8,5 em um questionário de satisfação que será aplicado entre os usuários.

Apesar de usarmos ambos os termos requisitos e metas de usabilidade com sentido semelhante, ou seja, no sentido de uma medida do desempenho esperado dos usuários quando utiliza o software, usamos o termo “requisito” quando estes são definidos pelo cliente ou usuários na especificação de requisitos e o termo “meta” quando os níveis de desempenho são decisões de desenho. Na prática, significa que os requisitos são considerados critérios para a aceitação do software, podem ser exigidos pelo cliente da mesma forma que outros requisitos também o são. Já as metas de usabilidade são utilizadas pela equipe de desenvolvimento com referência para a avaliação da qualidade da interação proporcionada pela interface.

3.2.2.5 Revisão da análise de usabilidade

O objetivo dessa atividade é avaliar a qualidade e aperfeiçoar os artefatos produzidos nas atividades de Análise de contexto de uso e de especificação de requisitos. Como parte do

trabalho de revisão, dever ser realizado uma validação do PRI (Protótipo de Requisitos de Interface) com a participação dos usuários.

3.2.2.6 Definição do estilo da interação

O desenho da interface do usuário envolve a criação e utilização de padrões, usualmente chamados de Guia de Estilo, relacionados à interação. Esses padrões podem especificar, por exemplo, tipo de fonte a ser utilizado, formato de telas ou páginas web, comportamento e forma de elementos de interação, dentre outros. Padrões são importantes no desenho de interface visando à usabilidade não só porque facilitam a utilização pelos usuários, mas também pelo potencial de reuso que representam para o desenvolvimento.

Um Guia de Estilo pode abranger padrões, diretrizes ou até métodos para o desenvolvimento da interação visando garantir a consistência entre famílias de produtos ou mesmo dentro de um produto. Uma organização que desenvolve software deve documentar os estilos de interação que utiliza em um Guia de Estilo, ou em um conjunto de guias de estilo para utilização interna pelas equipes de desenvolvimento da interface com o usuário. Uma coleção de guias de estilo pode estar organizada em uma estrutura hierárquica, envolvendo relacionamentos de herança entre os documentos, de modo que um padrão definido em um determinado nível nesta hierarquia deve conformidade a padrões definidos em níveis superiores da hierarquia.

A atividade de definição de estilo da interação visa o desenvolvimento de um estilo de interação ou a atualização ou a melhoria de estilos criados anteriormente. O Praxis-u provê um gabarito para a definição do estilo da interação denominado "GEUS": Guia de Estilo de Usabilidade.

3.2.2.7 Desenho da interação

A atividade de Desenho da Interação parte dos resultados das atividades de análise e de especificação de requisitos com o objetivo de se desenvolver a especificação da interface do usuário pronta para ser implementada em uma plataforma específica. Cabe observar que a implementação da interface desenhada normalmente é considerada fora do escopo da Engenharia de Usabilidade, sendo considerado um aspecto construcional e, portanto, como assunto para a Engenharia de Software. Assim, o resultado do desenho da interação é uma especificação pronta para ser desenvolvida sob o aspecto construcional, objeto das disciplinas de Desenho e Implementação normalmente incluídas em um processo de desenvolvimento de software.

O Desenho da interação é uma atividade complexa, objeto de um subprocesso no Praxis-u, envolvendo, inclusive, atividades de avaliação da usabilidade com a utilização de protótipos de desenho da interface do usuário.

3.2.2.8 Revisão do desenho da interação

O objetivo dessa atividade é avaliar a qualidade e aperfeiçoar os artefatos produzidos nas atividades de Definição do Estilo de Interação e Desenho da Interação.

3.2.2.9 Avaliação de usabilidade

A Avaliação de usabilidade visa à avaliação da qualidade da interface como instrumento da interação usuário-computador. É comum serem feitas várias avaliações de usabilidade durante o ciclo de desenvolvimento de um produto de software. Um planejamento inicial da freqüência e tipo de avaliações deve ser feito dentro da atividade de personalização do processo para projetos específicos. Um planejamento detalhado de cada avaliação deve ser realizado dentro do cada ciclo pelo fluxo de usabilidade.

Diferentes tipos de avaliação, com objetivos e características próprias, podem ser utilizados. Existem técnicas de avaliação, como a Avaliação Heurística, que utilizam o formato de revisões. As avaliações mais confiáveis, no entanto, envolvem experimentações com a utilização de protótipos ou do produto final e a participação de representantes dos usuários. Estas avaliações, também chamadas de testes com os usuários, normalmente fazem uso de roteiros de tarefas que são executadas por usuários com a utilização de protótipos ou com o produto final, dependendo do estágio de desenvolvimento do software.

No Praxis-u, o planejamento das avaliações de usabilidade deverá ser documentado em um documento próprio: "dausw" - Descrição de Avaliações de Usabilidade. Um relatório de cada avaliação também deverá ser registrado em um relatório denominado "rausw" – Relatório de Avaliações de Usabilidade.

3.3 GLOSSÁRIO

REVISÃO

Um processo ou reunião durante o qual um produto de trabalho ou um conjunto de produtos de trabalho é apresentado a desenvolvedores, gerentes, usuários, clientes ou outras partes interessadas, para comentários ou aprovação. (IEEE em Pádua Filho, 2009)

UML

A Unified Modeling Language (UML) é uma linguagem de modelagem que constitui um padrão utilizado mundialmente para descrição de modelos usados no desenvolvimento de software. Foi definida inicialmente pelos criadores do UP (Unified Process), atualmente sob os cuidados da OMG.

Basicamente, a UML permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados. Junto com uma notação gráfica, a UML também especifica significados, isto é, semântica. É uma notação independente de processos, embora o RUP (Rational Unified Process) tenha sido especificamente desenvolvido utilizando a UML (Wikipedia n.d.).

OMG

O Object Management Group, ou OMG, é uma organização internacional que aprova padrões abertos para aplicações orientadas a objetos. O Object Management Group foi fundado em 1989. (Wikipedia n.d.).

3.4 BIBLIOGRAFIA

Booch, G.; Rumbaugh, J.; Jacobson, I., *Unified Modeling Language User Guide*, 2nd Edition, Addison Wesley, 2005.

ISO 9241-11 *Ergonomic requirements for office work with visual display terminals (VDTs)* -- Part 11: Guidance on usability 1998.

ISO/IEC 9126 Information technology – software product quality- part 1: quality model 1999, (FDIS).

PADUA, W. *Engenharia de Software: Fundamentos, Métodos e Padrões*. 2. ed. Rio de Janeiro: Editora LTC, 2003. v. 1. 604 p.

Rumbaugh, J.; Jacobson, I.; Booch, G., *The Unified Modeling Language Reference Manual*, Addison Wesley, 2nd edition, 2004.

Wikipedia, em <http://pt.wikipedia.org/wiki>, acessado em 10/2009

4 ANÁLISE DE CONTEXTO DE USO

Este capítulo detalha a atividade de Análise de contexto de uso do Praxis_u que foi introduzida no capítulo 3 deste texto. A Análise de contexto de uso visa caracterizar todo o contexto envolvendo as pessoas e as atividades que elas realizam com o objetivo de coletar informações importantes para outras atividades do processo de desenvolvimento.

Neste texto, focamos a situação em que a Análise de contexto de uso vai ser utilizada no processo de desenvolvimento de um produto de software. Isso, claro, não impede que esse tipo de análise possa ser usado também para um produto já existente com objetivo de se fazer melhorias ou desenvolver uma nova versão, ou mesmo visando à obtenção de informação para se fazer uma avaliação mais criteriosa do produto.

A Análise de contexto de uso envolve diversos tipos de análise que podem ser categorizadas como:

- Análise de usuários
- Análise de tarefas
- Análise de Ambiente
- Análise de Produtos Concorrentes ou Similares

É importante ressaltar que os diversos tipos de Análise de contexto de uso são interdependentes: por exemplo, a Análise de tarefas fornece elementos para a Análise de usuários e vice-versa. Até por motivos didáticos e para organizar o texto, separamos os diversos tipos de análise. No entanto, é normal que no levantamento de informações associadas a um tipo de análise sejam identificados aspectos relacionados a outro tipo de análise.

É comum também que durante um tipo de análise sejam lembrados de elementos que pertencem a outro tipo de análise. Por exemplo, é comum acontecer que durante uma Análise de tarefas se identifique um novo papel de usuário, que ainda não havia sido registrado, e que se volte para melhorar a Análise de usuários. É por isso que se diz que o trabalho de modelagem é por natureza iterativo e interativo!

Há ainda o caso de atividades do processo que envolvem aspectos relacionados a mais de um tipo de Análise de contexto de uso. Por exemplo, o Levantamento de modelos mentais

está ligado à Análise de usuários, mas envolve também aspectos relacionados à execução de tarefas. Outro exemplo, o ambiente de realização de atividades é um aspecto que pode estar relacionado a um papel de usuário ou a uma tarefa que pode ser de responsabilidade de vários papéis de usuários.

Algumas técnicas são utilizadas para se obter as informações desejadas na Análise de contexto de uso. Essas técnicas e aspectos a elas relacionados são apresentados na próxima seção.

4.1 TÉCNICAS

A caracterização do contexto de uso de um produto de software deve ser realizada por meio do emprego de várias técnicas para obtenção de informação, como as visitas a campo, entrevistas e reuniões de prospecção. É importante obter essas informações sistematicamente para se garantir a qualidade dos dados obtidos e ouvindo, preferencialmente, os próprios usuários. Essa prática está dentro da abordagem de desenvolvimento orientado pelo cliente e/ou usuários.

A seleção da técnica mais apropriada depende da informação desejada e dos custos envolvidos. Para gerar idéias e conceber um modelo que leve em conta o ponto de vista dos usuários do produto, as técnicas qualitativas são as mais apropriadas. Essas técnicas permitem produzir um conjunto de informações, o mais amplo possível, evitando-se idéias preconcebidas, buscando o conhecimento simplesmente ouvindo e observando os usuários ou outras fontes de informação apropriadas. As técnicas qualitativas mais usadas são a observação direta e as entrevistas individuais ou em grupo.

Quando se deseja obter informações numéricas, tais como, grau de preferência entre versões de um mesmo produto, as técnicas quantitativas são necessárias. Neste caso, pode-se usar o levantamento por questionário (*survey*), que pode ser aplicado por meio de entrevistas diretas ou utilizando o correio e outros. A escolha da forma de aplicação do questionário depende do tipo de informação que será coletada, da velocidade desejada na resposta e orçamento disponível.

Para o desenvolvimento de um produto de sucesso deve-se fazer uso de ambos os tipos de técnicas, qualitativas e quantitativas, uma vez que estas são complementares no tipo de informação que fornecem.

4.1.1 OBSERVAÇÃO DIRETA

Algum tipo de observação direta, formal ou informal, é essencial para se obter uma compreensão do contexto de trabalho dos usuários finais (Hackos & Redish 1998). As observações formais podem ser realizadas no próprio ambiente de trabalho dos usuários ou em laboratório. A observação pode ser passiva (simplesmente ouvindo e observando) ou ativa (questionando). As observações passivas podem servir de insumo para elaboração de questionários e geralmente necessitam da realização de entrevistas posteriores para descobrir as razões de certas ações. A realização das observações depende dos prazos, permissão dos clientes e custos.

4.1.2 ENTREVISTAS

As entrevistas têm o objetivo de colher informações sobre os usuários e as atividades que realizam através de conversas com pessoas envolvidas. As entrevistas se apóiam na experiência dos entrevistados. As entrevistas podem ser feitas individualmente ou em grupos.

Nas entrevistas individuais, deve-se buscar descobrir as características declaradas e as latentes, a forma com que a pessoa trabalha no cotidiano, as dificuldades e os aspectos positivos no modo com realizam suas atividades e as diferenças individuais dos usuários.

Já as entrevistas de grupos, são realizadas por meio de discussões abertas com um grupo de usuários. A forma como a reunião é estruturada e conduzida depende da técnica usada: *brainstorming* ou oficinas de JAD (Joint Application Development) são algumas técnicas normalmente utilizadas. O JAD é uma técnica de oficina desenvolvida originalmente na IBM, mas hoje em dia utilizada largamente em várias organizações com o objetivo de se ganhar objetividade e eficiência em reuniões de grupo - veja Glossário.

Entrevista é uma técnica muito utilizada, provavelmente ser mais rápida e prática do que a observação direta dos usuários realizando suas atividades. No entanto, é importante observar que a entrevista tem a limitação de que a pessoa entrevistada vai sempre oferecer a sua versão de como as coisas acontecem. Inevitavelmente, a opinião de cada pessoa expressa uma tendenciosidade em função de sua experiência pessoal. Essa tendenciosidade pode até ser intencional, quando envolve algum interesse da pessoa entrevistada. Mas a questão não é esta, não é que exista má fé nas pessoas que concedem as entrevistas, isso é um processo natural e que pode gerar distorções. Por exemplo, a pessoa entrevistada tende

a exagerar a importância das coisas que faz bem feito e com as quais tem facilidade. Por outro lado, podem omitir ou minimizar suas dificuldades.

Prós e contras considerados, pode-se dizer que a observação direta e a entrevistas são técnicas que se complementam e que levam a informações mais realistas e fidedignas sobre o contexto onde os usuários realizam suas atividades.

4.1.3 LEVANTAMENTO POR QUESTIONÁRIO

Essa técnica é indicada quando se deseja obter informações tais como informações sobre a experiência, atitudes e preferências dos usuários que podem afetar o modo como eles realizam suas atividades. Geralmente são utilizadas quando se deseja informações estatísticas sobre uma grande população de usuários.

Questionários e levantamentos (*surveys*) são um meio alternativo para se saber a opinião dos usuários sobre algo e também descobrir quais características do produto são mais importantes para um determinado grupo de usuários. Quando comparados às entrevistas face-a-face ou oficinas de JAD (Joint Application Development) e Brainstorming, os questionários são mais limitados na obtenção de informações. As questões abertas podem contornar essa limitação, uma vez que oferecem ao respondente a opção de livre expressão; no entanto, consome mais tempo para resposta e análise. Geralmente, os questionários são fontes de informação suplementar ou de confirmação de conjecturas.

A construção de questionários exige muitos cuidados e aplicação de técnicas estatísticas para se conseguir coletar a informação desejada, fidedigna e com o mínimo de erros. As perguntas devem ser redigidas sem ambigüidades, as escalas escolhidas com critérios apropriados e os pré-testes realizados com cautela. Por esses motivos, sempre que possível, deve-se recorrer a profissionais da área.

4.1.4 FONTES DE INFORMAÇÃO

Outro ponto a ser considerado na Análise de contexto de uso são as fontes de informação. A principal fonte de informação e a que deve ser primeiramente consultada, quando possível, é o próprio usuário do produto. No entanto, outras fontes podem oferecer informações complementares importantes para o desenho de uma interface, sob uma perspectiva diferente daquelas dos usuários finais. É preciso ressaltar, no entanto, que a utilização de

outras fontes de informação não deve substituir, mas complementar, a observação direta e a entrevista envolvendo os usuários.

As fontes alternativas de informação sobre o contexto de uso do software são mais úteis à medida que oferecem maior conhecimento sobre o usuário e suas necessidades reais. Por ordem decrescente de importância, podemos citar as pessoas que atuam junto aos usuários, os informantes e intérpretes e as fontes não humanas.

4.1.4.1 Pessoas que atuam junto aos usuários

Além das pessoas que serão usuários diretos do produto em consideração, há outros atores que atuam junto aos usuários ou que representam papéis semelhantes e que, para alguns propósitos, podem complementar as informações que são obtidas com os usuários finais. Os tipos de papéis que podem ser considerados relacionados aos usuários e de interesse na Análise de contexto de uso são apresentados a seguir.

ESPECIALISTAS NO DOMÍNIO DA APLICAÇÃO

Especialistas no domínio da aplicação são aquelas pessoas que têm um bom conhecimento do domínio de que trata o software em consideração, ainda que não sejam usuários (ou futuros usuários) do produto. Por exemplo, um bom Contador será um especialista no domínio de Contabilidade quando estamos considerando um software para ser usado por um lojista, mas que vai envolver aspectos de contabilidade da loja onde for utilizado.

Geralmente, os especialistas no domínio da aplicação oferecem perspectivas sobre as necessidades dos usuários finais que nem sempre os próprios usuários saberiam expressar. Entretanto, é preciso cautela, pois o especialista pode não ter um conhecimento direto do trabalho a ser suportado, ou seja, as demandas e questões que surgem com a prática no trabalho diário. Ou seja, o especialista complementa, mas nem sempre substitui, o usuário como fonte de informação.

USUÁRIOS DE SISTEMAS SIMILARES

Usuários de produtos similares ao que está sendo desenvolvido podem fornecer informações comparativas importantes e até estratégicas sobre diferenciais do outro produto, por exemplo, sobre pontos em que apresentam boas soluções ou sobre necessidades que não são cobertas pelo produto que se tem em mente. Mesmo aspectos negativos do outro produto podem ser relevantes na medida em que podem ser explorados positivamente como um diferencial competitivo para o produto em consideração.

INSTRUTORES

Instrutores são responsáveis por treinar pessoas em aspectos específicos de uma atividade alvo do sistema em consideração. O treinamento pode ser também no uso de um software semelhante ou uma versão anterior ao software em consideração. Instrutores tendem a ter alguma familiaridade com princípios gerais e assuntos práticos relacionados ao uso de um produto. Geralmente identificam o que no sistema ou no ambiente dificulta o aprendizado dos usuários ou causa ineficiência na realização de algumas tarefas.

SUPERVISORES

Supervisores são os gerentes diretos dos usuários finais. São boas fontes de informação quando têm conhecimento de como realmente os usuários realizam o trabalho, não somente de como deveria ser realizado.

4.1.4.2 Informantes e Intérpretes

São aquelas pessoas que podem falar sobre as necessidades dos usuários ou interpretá-las, mas não os substituem efetivamente. Geralmente, têm um conhecimento complementar, relacionado a algum aspecto do papel de usuário efetivo. Entre os candidatos a potenciais informantes e intérpretes estão os profissionais de marketing, vendedores, e pessoal do suporte técnico

SUPORTE TÉCNICO

As pessoas responsáveis pelo suporte técnico geralmente têm contato direto com os usuários e conhecem seu ambiente de trabalho. Podem ser uma boa fonte de informação sobre os usuários e o tipo de uso do software. O contato direto com os usuários finais propicia o conhecimento dos problemas que ocorrem quando estes utilizam o produto. No entanto, costumam saber pouco sobre como normalmente os usuários trabalham e sobre os aspectos que são relevantes em um projeto de interface dos usuários.

MARKETING

As pessoas responsáveis pelo marketing podem servir como uma ponte entre os projetistas de interface e os usuários, mas, no entanto, podem ter somente uma visão limitada dos usuários e do produto. Muitas práticas da área de marketing focalizam o mercado, tendo, portanto, conhecimento limitado sobre as necessidades do usuário e as circunstâncias de uso do produto. Entretanto, há também práticas específicas de marketing que visam entender as

necessidades do usuário, aquilo que ele valoriza, para que a empresa fornecedora do produto obtenha melhor posicionamento no mercado.

É importante, no entanto, considerar que, na maioria das vezes, as informações obtidas via departamento de marketing, são *insights* sobre o que seria bom para os grupos potenciais que utilizam ou utilizarão o produto. Não podem ser confundidas com as informações essenciais sobre os usuários e suas atividades, suas necessidades e a prática de utilização do produto.

VENDEDORES

De forma semelhante ao pessoal que cuida do marketing, para o vendedor às vezes é mais importante conseguir clientes do que apreciar suas necessidades, ou seja, nem sempre o vendedor terá uma visão real da utilização do produto pelos usuários. No entanto, em geral são mais indicados para serem ouvidos que os responsáveis pelo marketing em face do contato direto com os usuários finais e a facilidade de acesso a eles.

ESPECIALISTAS EM DOCUMENTAÇÃO

Os especialistas em documentação são redatores e especialistas que escrevem manuais de usuários e criam artefatos de ajuda (*help*). Podem ser uma rica fonte de informação sobre os usuários e o modo de usar o produto, já que freqüentemente são consumidores desse tipo de informação. Cabe lembrar que a documentação também é objeto do trabalho que visa garantir a usabilidade global do sistema, ou seja, a documentação faz parte do software e tem impacto em sua usabilidade.

4.1.4.3 Fontes não humanas

Os projetistas de interfaces também podem obter informações a partir de objetos e artefatos produzidos nas empresas. Essas fontes podem ser importantes e usadas para complementar as informações obtidas de humanos associados ao software em consideração.

MANUAIS

Manuais de padrões, processos e procedimentos elaborados para definir o trabalho e a forma como deve ser realizado podem ser usados pelos desenvolvedores da interação como fonte de informação. No entanto, é importante considerar que nem sempre expressam o modo prático como o trabalho é efetivamente realizado e, portanto, devem ser utilizados com

cautela. Outro aspecto a considerar é que os manuais podem ser usados para orientar o tratamento de aspectos específicos ou de circunstâncias raras ou excepcionais.

DADOS DERIVADOS

Esta fonte refere-se à informação sobre os usuários e suas necessidades que é derivada de registros ou informações obtidas para outros propósitos. Publicações internas arquivadas na empresa podem ser uma boa fonte de informação indireta sobre o trabalho e necessidades dos usuários. Além disso, banco de dados ou arquivos em papel contendo FAQ's (Frequently Asked Questions) contêm informações sobre os problemas mais freqüentes encontrados pelos usuários.

Nem toda essa informação está relacionada com a usabilidade do produto, mas pode ter implicações diretas ou indiretas. Por exemplo, um recurso ou serviço que o usuário não consegue encontrar, talvez possa estar relacionado com o layout da interface ou a navegação. Feedbacks espontâneos e queixas voluntárias relacionadas a versões anteriores do produto ou a produtos similares representam uma boa amostra a ser pesquisada. Ao se avaliar uma amostra desse tipo, deve-se pesquisar as causas dos problemas visando à solução, uma vez que geralmente nesse tipo de informação não se registra quais características as pessoas consideram que proporcionariam um uso mais eficiente.

4.2 SUBPROCESSO DE ANÁLISE DE CONTEXTO DE USO

Nesta seção apresentamos o subprocesso de Análise de contexto de uso do Praxis-u. As atividades e artefatos do subprocesso são apresentados de forma sistematizada, visando o leitor interessado no desenvolvimento da interação. A Análise de contexto de uso produz artefatos que definem modelos, como o modelo de Personas que descreve papéis de usuários, e o documento “erusw.dot” que registra toda a informação produzida.

O diagrama de atividades abaixo apresenta o subprocesso de Análise de contexto de uso. Cabe lembrar que, no diagrama, o retângulo com bordas arredondadas denota atividades e os outros retângulos denotam objetos, no caso, artefatos, que podem ser consumidos ou produzidos pelas atividades.

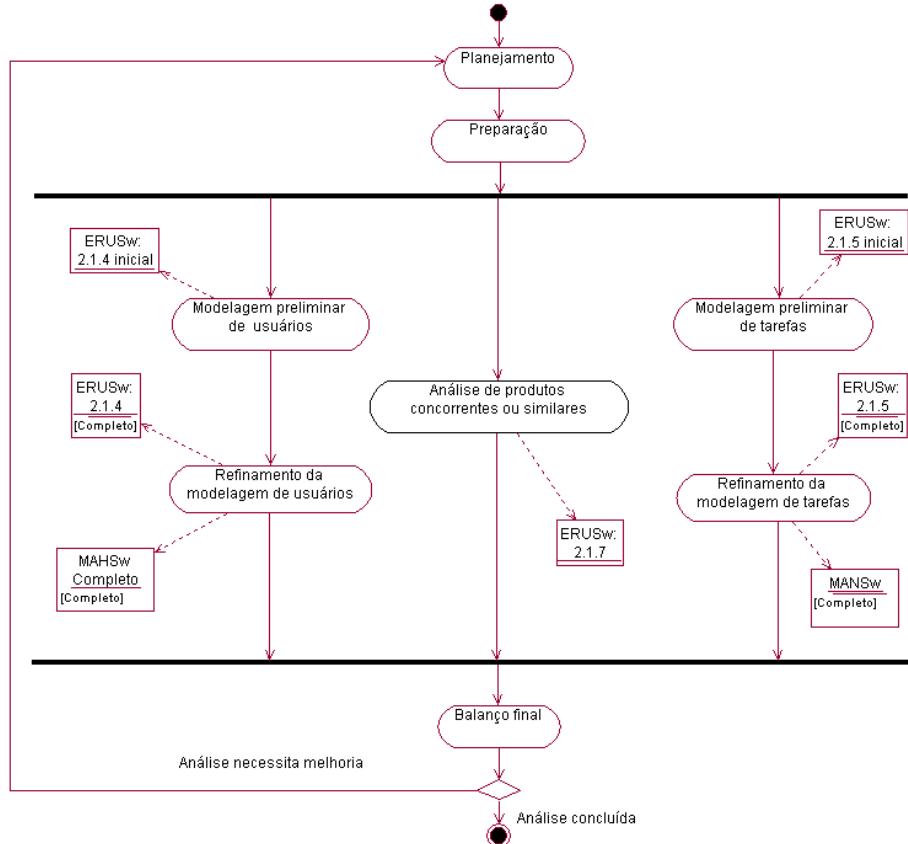


Figura 4-1: Atividades e artefatos do sub-fluxo de Análise de contexto de uso

4.2.1 VISÃO GERAL

Basicamente, o subprocesso se inicia com as atividades de planejamento e preparação. A seguir, o diagrama apresenta três fluxos de atividades que podem ser realizados em paralelo. Dois desses fluxos correspondem às atividades que visam à modelagem de usuários e de tarefas. Os principais resultados da Análise de contexto de uso são relacionados à modelagem de usuários e de tarefas, que têm a finalidade de caracterizar os atores, as tarefas que eles realizam e os relacionamentos envolvendo os atores. Cada um desses fluxos é constituído de uma modelagem preliminar e outra mais detalhada. O outro fluxo trata da Análise de Produtos Concorrentes ou Similares.

Cabe observar que, em geral, grande parte das tarefas de Análise de contexto de uso são realizadas em campo, isto é, no ambiente onde os potenciais usuários exercem suas

atividades. O trabalho de campo envolve observação das pessoas no local onde realizam suas atividades e a realização de entrevistas, conversas ou reuniões de grupo com a participação das pessoas envolvidas visando à coleta de informação para utilização na modelagem. Portanto, as atividades de modelagem envolvem a realização de trabalho de campo em paralelo com as atividades de modelagem no ambiente de trabalho dos desenvolvedores.

Detalhando, as atividades da Análise de contexto de uso, conforme prescritas no Praxis_u, são descritas abaixo.

4.2.2 PLANEJAMENTO

Nesta atividade, analisa-se o problema a ser resolvido, considerando-se os diversos tipos de Análise de contexto de uso, e faz-se o planejamento dos trabalhos. O planejamento tem os seguintes objetivos:

- Identificação das pessoas de contatos, aquelas que podem dar orientações iniciais (como por exemplo, sobre pessoas e locais para o trabalho de campo) para o trabalho de análise a ser realizado.
- Definição da metodologia de trabalhos, especialmente do trabalho de campo. São definidas as formas do trabalho de observação e de entrevistas com as pessoas interessadas no produto.
- Programação das atividades a serem realizadas, incluindo cronograma de atividades e estimativas de escopo, recursos

Além disso, o Planejamento inclui a Definição estratégica conforme apresentado a seguir.

4.2.2.1 Definição estratégica

A Definição estratégica é a atividade inicial que visa prover informação para o Planejamento e outras atividades da Análise de contexto de uso. Nesta atividade, estuda-se o problema a ser resolvido, considerando o ambiente de negócio do qual fará parte o produto em desenvolvimento. A Definição estratégica envolve identificação de aspectos estratégicos relacionados aos objetivos futuros de negócio da empresa que possam ter impacto no produto em desenvolvimento. Por exemplo, a empresa quer expandir ou manter a posição no mercado? Quem são os clientes e concorrentes?

Os resultados da atividade de Definição Estratégica são descritos em uma Declaração de Visão.

DECLARAÇÃO DE VISÃO

A Declaração de visão deve ser descrita no documento “eruw”. Esta apresenta uma concepção em termos estratégicos do produto em desenvolvimento. A Declaração de visão está muito relacionada à análise do negócio, muitas vezes executada na disciplina de Modelagem de Processos de Negócio utilizada na engenharia de software. O principal objetivo da Modelagem de Processos de Negócio é se conhecer o negócio para apoio do qual o software será desenvolvido, para que o software a ser desenvolvido possa estar alinhado com as necessidades de negócio. Sendo assim, podemos considerar a Declaração de Visão como uma síntese da descrição do negócio em termos estratégicos que é usada como guia para o trabalho de Análise de contexto de uso.

A Declaração de Visão deve conter os seguintes elementos:

- objetivos, pressupostos e uma análise inicial das partes interessadas (stakeholders). Deve também analisar o impacto desses aspectos no produto;
- conceito do produto em desenvolvimento;
- identificação de clientes, possíveis usuários e demais partes interessadas:
 - Quem são e quais suas características?
 - Como é a interação dos clientes com as mudanças no negócio?
- requisitos do negócio;
- descrições do contexto do negócio corrente
 - o que está mudando?
 - o que é importante?
 - problemas que podem ser visualizados;
 - quais tipos de resultados são esperados;
- possíveis cenários que podem vir a ocorrer;
- concorrentes:
 - quem são e o que fazem?
 - de que modo estão mudando seus negócios?

- tamanho e posição na indústria:
 - como o negócio está posicionado na indústria?
 - como a indústria está mudando?
- ambiente do negócio:
 - quais mudanças (especificamente política e tecnológica) estão ocorrendo?
- percepção pública:
 - como o público percebe a organização?
 - em que aspectos é desejável mudar essa percepção?
- nível de serviço:
 - qual é o nível do serviço ao cliente?
 - o serviço pode ser melhorado ou estendido?

4.2.2.2 Detalhamento da atividade de planejamento

A atividade de planejamento é realizada por meio de várias tarefas que são apresentadas detalhadamente na Tabela 4-1. Além de um detalhamento das tarefas, a tabela apresenta também os artefatos do processo Praxis-u envolvidos.

Descrição	Planejamento
Insumos	1 Proposta de Especificação do Software (pesw) 2 Especificação de Requisitos do Software (ersw – escopo do produto, funções do produto, restrições, materiais de referência).
Tarefas	1 Definição estratégica <ul style="list-style-type: none"> Atividade inicial que visa prover informação para o Planejamento e outras atividades da Análise de Contexto. Estuda-se o problema a ser resolvido, considerando o ambiente de negócios do qual fará parte o produto em desenvolvimento. Envolve identificação de aspectos estratégicos relacionados aos objetivos futuros de negócios da empresa que possam ter impacto no produto em desenvolvimento. Os resultados da atividade de Definição Estratégica são descritos em uma Declaração de Visão no documento erusw. 2 Identificação das pessoas de contato. 3 Consulta a contatos e estudo da documentação disponível, buscando informações sobre os usuários e suas atividades relacionadas com o produto em desenvolvimento, tais como: <ul style="list-style-type: none"> informações dos usuários coletadas em entrevistas durante as sessões de JAD ou à distância, sobre suas tarefas e modo de realizá-las;

	<ul style="list-style-type: none"> ◦ informações internas à empresa fornecedora do produto, ou seja, referentes ao histórico de produtos similares ou a ser substituído; ◦ reclamações dos usuários, no caso de produto a ser melhorado. ◦ as principais tarefas de usuários às quais o sistema deverá dar suporte; os objetivos dessas tarefas e a maneira como se relacionam logicamente; ◦ os recursos técnicos e físicos disponíveis para a realização da tarefa, incluindo treinamento e apoio técnico; ◦ o fluxo de informações, ou seja, quem emite e quem recebe informações na realização de uma determinada tarefa na empresa cliente; quais são os documentos produzidos ou consumidos, suas estruturas, volume e modo de utilização; ◦ o escopo do sistema: principais requisitos funcionais e não funcionais e restrições sobre o desempenho, segurança, equipamentos. ◦ Identificação de produtos concorrentes ou similares com registro de suas características mais relevantes.
Resultados	<ol style="list-style-type: none"> 1 Eventuais pedidos de esclarecimentos às pessoas responsáveis pelos outros tipos de análises. 2 Levantamento inicial dos requisitos. 3 Programação das atividades de Análise de contexto de uso. 4 Eventuais solicitações de melhorias nos resultados de análises realizadas anteriormente.

Tabela 4-1 Atividade de planejamento

4.2.3 PREPARAÇÃO

Nesta atividade identifica-se a população alvo, isto é, definem-se quais são as pessoas que deverão ou poderão usar o sistema e classifica-se essa população, estabelecendo-se grupos de usuários com limites bem claros. Cabe observar que não se trata de uma Análise de usuários, apenas de uma identificação em mais alto nível de abstração da população alvo com o objetivo de se definir as pessoas que serão objeto da Análise de usuários. Preparam-se os artefatos a serem usados durante o trabalho de análise, especialmente no trabalho de campo. Faz-se o contato com as pessoas selecionadas para o trabalho de campo visando prestar esclarecimentos sobre as atividades a serem realizadas e combinar os encontros de trabalho.

4.2.3.1 Detalhamento da Preparação

A atividade de Preparação da Análise de contexto de uso consiste principalmente no levantamento da população alvo, apresentado na Tabela 4-2.

Descrição	Preparação
Insumos	1 Especificação de requisitos de Software (ERS) 2 Documentação do planejamento da análise
Tarefas	1 Identificação das pessoas que potencialmente usarão o produto, direta ou indiretamente, em termos dos papéis que podem assumir em relação ao produto, podendo ser: <ul style="list-style-type: none"> ◦ pessoas que comprariam o software e o utilizariam sem assistência ou interação com outras pessoas, em casa ou em seus locais de trabalho; ◦ grupos de pessoas que usariam o produto como parte do trabalho que realizam ou como parte do processo de negócio; ◦ aqueles que iriam administrar o produto; ◦ pessoas que dariam suporte ao produto; ◦ pessoas que seriam responsáveis pela instalação do produto para si e para outras; ◦ clientes dos usuários que sofreriam algum tipo de impacto devido ao uso do produto. 2 Categorização dos possíveis usuários em grupos e definição das categorias que o produto atenderá prioritariamente. 3 Preparação dos artefatos a serem usados nos trabalhos. 4 Fazer contatos e marcar encontros de trabalho.
Resultados	1 Lista preliminar de usuários potenciais e respectivas categorias.

Tabela 4-2 Levantamento da população alvo

4.2.4 MODELAGEM DE USUÁRIOS

Esta atividade visa à modelagem dos papéis de usuários, incluindo os vários aspectos a eles relacionados. A modelagem de usuários foi subdividida em duas atividades que visam uma modelagem preliminar e um refinamento da modelagem dos usuários. Essas atividades são apresentadas na seção 4.3 Análise de usuários adiante.

4.2.5 MODELAGEM DE TAREFAS

Esta atividade visa à modelagem das tarefas, realizadas pelos usuários, que vão ter apoio do sistema em consideração. O apoio do software à realização de uma tarefa pode ser parcial, apoiando a realização de partes das tarefas consideradas, ou pode se dar em maior grau com o sistema apoiando até a realização completa da tarefa. Há várias formas de modelagem de uma tarefa, a escolha da forma de representação vai depender do tipo de tarefa, do tipo de apoio que se deseja que o sistema proveja e do detalhamento que se deseja atingir na modelagem da tarefa.

De forma semelhante à modelagem de usuários, a modelagem de tarefas também foi subdividida em duas atividades que visam uma modelagem preliminar e um refinamento da modelagem. Essas atividades são apresentadas na seção 4.4 Análise de tarefas adiante.

4.2.6 ANÁLISE DE PRODUTOS CONCORRENTES OU SIMILARES

Esta análise visa à coleta de informações para que se possa melhorar o produto em consideração a partir do conhecimento das fraquezas e pontos fortes dos produtos concorrentes ou similares. A Análise de Produtos Concorrentes ou Similares pode ser realizada por meio de avaliações de usabilidade de maneira mais formalizada ou de forma simplificada, como também pode ser complementada com análises disponíveis em revistas ou outros meios que podem dar subsídios valiosos para o desenvolvimento de um sistema. Quando possível, a análise de vários produtos concorrentes permite uma avaliação comparativa bem interessante.

4.2.7 BALANÇO FINAL

Esta atividade visa basicamente à verificação dos artefatos produzidos na Análise de contexto de uso, incluindo a verificação da consistência entre os diversos modelos. A atividade de Balanço Final tem também o objetivo de registrar as conclusões e lições aprendidas. As atividades de Balanço Final estão descritas na Tabela 4-3.

Descrição	Revisão
Inssumos	1 Especificação de requisitos de Software (ersw) 2 Modelo de Atores Humanos 3 Informação levantada no trabalho de campo. 4 Manuais de usuários (de sistemas similares, se existirem, ou do atual, a ser melhorado).
Tarefas	1 Verificação dos artefatos produzidos, analisando sua correção e facilidade de entendimento. 2 Registro de conclusões e lições aprendidas. 3 Realização de melhorias no modelo, se necessário.
Resultados	1 Artefatos relativos à modelagem de usuários, de tarefas e à análise de produtos concorrentes e similares completos e revisados. 2 Seção da “erusw” que documenta a Análise de contexto de uso completa e revisada.

Tabela 4-3 Balanço final

4.3 ANÁLISE DE USUÁRIOS

4.3.1 VISÃO GERAL

A atividade de Análise de usuários visa caracterizar a população-alvo envolvida na utilização do produto de software em perspectiva. A Análise de usuários é bastante relacionada com a Análise de tarefas; em geral, essas atividades são realizadas em um processo incremental e iterativo, onde a Análise de usuários dá subsídios para a Análise de tarefas, inclusive sugerindo novas tarefas a serem consideradas para serem contempladas no produto, e vice-versa.

Do ponto de vista de engenharia de usabilidade, nos interessa caracterizar os usuários somente com relação aos aspectos que têm algum impacto no desenvolvimento do produto.

O termo utilizado por (Constantine & Lockwood 1999) para representar uma abstração das características relevantes dos usuários é o de *Papel-de-usuário*, segundo ele: uma coleção abstrata de necessidades, interesses, expectativas, comportamentos e responsabilidades, caracterizando um relacionamento entre uma classe de usuários e o sistema. No contexto deste documento, assim como no Praxis-u, além do termo *Papel-de-usuário* usaremos também o termo *Autor* já que este termo é consagrado na Engenharia de Software. No entanto, lembramos que o termo *Autor* usado na usabilidade refere-se somente a atores humanos.

É importante também notar que o Autor é uma abstração, não correspondendo a pessoas com existência real. Um Autor pode abstrair o papel de várias pessoas assim como uma pessoa pode desempenhar papéis que são modelados em vários autores.

Diferentes técnicas podem ser usadas para a modelagem de usuários, por exemplo, os papéis de usuário propostos por Constantine e Lockwood (Constantine, L.L. & Lockwood, L.A.D. 1999), as formas mais descritivas propostas por Hackos, JT & Redish (Hackos, JT & Redish, 1998) ou modelos mais simples como proposto por Hix e Hartson (Hix, D.; Hartson, H. R., 1993). Aqui, preferimos sugerir a técnica de Personas, apresentada na seção 4.3.3,

por ser considerada barata, fácil e, de certa forma, intuitiva e lúdica para a equipe de desenvolvimento de um projeto.

4.3.2 OBJETIVOS

A Análise de usuários visa à obtenção de conhecimento detalhado dos possíveis usuários para que se possa desenvolver uma solução de interface adequada para o tipo de utilização que se espera do produto.

A Análise de usuários de um sistema interativo tem como objetivos específicos:

- Identificar e conhecer os diversos tipos de usuários que potencialmente utilizarão o produto;
- Caracterizar os usuários em todos os aspectos relacionados com seu modo de trabalho ou de comportamento que possam ter importância para o desenvolvimento do produto;
- Gerar um modelo de usuários, incluindo o relacionamento entre atores, a partir das informações obtidas;
- Fornecer insumos para a definição de requisitos e metas de usabilidade que devem ser observados para determinados atores;
- Definir os atores focais, ou seja, quais atores devem ser considerados prioritariamente no desenvolvimento do produto;
- Fornecer insumos para o desenho da interação, tais como a arquitetura global, a navegação e o conteúdo da interface, o layout e o look-and-feel dos elementos de interação.
- Fornecer insumos para o planejamento e especificação das avaliações de usabilidade.

A Análise de usuários representa um importante insumo para se conseguir agregar valor ao produto no sentido de torná-lo mais efetivo como ferramenta de apoio ao trabalho, ou outro tipo de atividade do usuário. Abaixo são listadas algumas questões mais comuns que se procura investigar na Análise de usuários.

- Quais são os grupos de usuários envolvidos direta ou indiretamente com o produto em perspectiva?

- O que os usuários do software irão fazer que tenha relação com o produto que está sendo desenvolvido?
- Quais são os interesses dos usuários que possam estar relacionado com o produto em desenvolvimento?
- Em que os usuários necessitam do sistema para realizar algo ou alguma tarefa?
- Como o sistema deve prover o que eles necessitam?
- O que no momento atual não está funcionando ou é ineficiente?
- Como o sistema seria mais efetivo para suportar as atividades do usuário?

4.3.3 PERSONAS

4.3.3.1 Introdução

Persona era o nome da máscara que atores do teatro grego usavam na caracterização de um personagem (Wikipedia n.d.). O uso de personas como uma técnica foi popularizado por Alan Cooper, em seu livro “The Inmates are Running the Asylum” (Cooper, 2000). Cooper introduziu o uso desse termo como uma ferramenta ou técnica usada no desenho da interação no desenvolvimento de software. O livro de Cooper, por sua vez, trata das características, da utilização e de boas práticas para a criação de personas.

A Persona é um personagem fictício usado para caracterizar um Papel-de-usuário de um produto de software. A figura 1 (Wikipedia n.d.) ilustra um exemplo de um grupo de personas utilizado em um projeto de desenvolvimento de software. A Persona é como uma ficha de personagem de RPG do usuário-modelo, criada a partir de dados reais como: nome, gostos, hábitos, habilidades etc (Mulder, 2006). Essas informações podem ser obtidas através de entrevistas com usuários potenciais ou através de conversas com quem lida freqüentemente com esse público. A técnica de Personas é aqui muito utilizada para a modelagem de usuários onde um Papel-de-usuário é modelado como uma persona.

Kivio Users				
	The researcher	The Sysadmin	The OSS developer	The CS student
Name	Alexander Weiß	Donald M. Berry	Kristian Larsson	Eric Neville
Age	30	30	26	24
Location	Germany	US	Sweden	France
Social Life	Alexander lives with his girl-friend in a flat in Hamburg.	Donald lives with his wife and 1-year old daughter in a house in Portland.	Kristian shares an apartment with two friends in Stockholm. His girl-friend lives in Uppsala. They see each other every weekend.	Eric lives with his parents in a small city close to Lyon. He visits the university there. Often, he stays at his friend's apartment for playing PC games and programming.
Work Life	He works at centre for environmental systems research and designs plans for replacable energies in a EU-funded project.	He is a lead system administrator in a huge network solutions company in Portland.	A software developer with a dayjob in a medium-sized software company. Works on KDE in his spare time.	He is a student of computer science. Besides university, he performs small programming jobs for people in his neighbourhood.
Computer Experience		All are highly experienced with computers.		
Time at a computer per week	26-50 hours per week	35-50++ hours per week	30-50++ hours per week	25-45 hours per week
Computer tasks	Office tasks and Field-dependent. Also educational and recreational. No development.	Development and network administration. Does not use PC for office tasks, educational, and even recreational.	Mostly development and recreational. Also network administration and office.	Mostly development. Also educational, recreational, and network administration. Does not use for office work.
Relation to OSS.	He is not passionate about OSS.	He is a convinced user of OSS.	He is involved with OSS development.	He is a convinced user of OSS.
Requirements wrt diagramming	office requirements	highest claims	easy-going	eager beaver
Frequency of drawing diagrams	Each 2 nd month	Twice per month	Once a month	Each 2 nd month
Diagram main type	Flowcharts. Also visualising thoughts. No technical ones.	All, except sitemap	Visualising thoughts	Diagrams mostly UML
Size and complexity	15-20 elements, 2-3 levels, 3-7 shapes	15-30 elements, 2-5 levels, 4-9 shapes	15-20 elements, 2-3 levels, 3-7 shapes	15-20 elements, 2-3 levels, 3-7 shapes
Diagram purpose and context.	For non-IT job.	Diagrams are for the IT-Job, never for himself.	Diagrams are mostly for himself but, also for formally presenting.	For formally presenting in university. Not for himself, as work input or for any jobs
Current diagramming	Power Point or OOo.	Visio,	Pen and Paper.	Dia, Umbrello.

Figura 4-2 Grupo de Personas usado no projeto da ferramenta de diagramação Kivio (Similar ao Microsoft Visio)

MOTIVAÇÃO

No desenvolvimento de uma boa interface de software é importante conhecer muito bem os perfis das pessoas que vão utilizar o produto (Cooper, 1999). Porém, como registrar esse conhecimento de forma prática durante o desenvolvimento? Como lidar com as diferenças que existem entre os usuários? Como não perder de vista as características relevantes dessas pessoas?

As Personas se aproveitam do poder das narrativas para aumentar a atenção, a memorização e a organização dos dados coletados sobre os usuários. A técnica de Personas é considerada barata, fácil e até divertida para a equipe de desenvolvimento de um projeto. No ritmo agitado da produção tecnológica, poucos são os que têm tempo (e interesse) em

ler relatórios de dezenas de páginas sobre os estudos de usabilidade, a etnografia e as pesquisas de mercado do marketing (Grudin, 2002).

Quando uma descoberta importante é feita sobre a utilização de um sistema que está sendo desenvolvido, é muito mais fácil comunicar a equipe toda, dizendo, por exemplo: "o Adalberto não está conseguindo usar a ferramenta de busca na nossa página", no lugar de "uma quantidade representativa dos participantes nos testes de usabilidade tiveram problemas com a ferramenta de busca".

COMO PODEM SER UTILIZADAS

As Personas podem ser utilizadas em diferentes fases do projeto, contribuindo de forma distinta para cada uma delas. Abaixo são listados alguns exemplos do uso de Personas:

Fase do projeto	Exemplo de uso
<u>Definição do produto:</u>	Personas podem ajudar a determinar o que vai ser incluído no produto e o que vai sair. Elas podem ser utilizadas na especificação de requisitos do sistema.
<u>Desenho do produto:</u>	Personas podem ajudar a equipe de desenho a definir como uma ferramenta irá se comportar. Para isso, podem ser criados, com o auxílio delas, cenários realistas de utilização do sistema.
<u>Medição e avaliação do produto:</u>	Personas podem ajudar a garantir a qualidade dos testes do produto, possibilitando os testadores escrever <u>scripts</u> de teste mais realísticos. É possível também que seja feita uma priorização na correção de <i>bugs</i> do sistema.

VANTAGENS

As principais vantagens da técnica de Personas são (Mulder, 2006):

- Engaja e conscientiza a equipe de projeto;
- Mantém o foco no usuário durante todo o projeto
- Facilita a chegada em um consenso sobre os interesses do usuário;
- Facilita a tomada de decisões no projeto, porque não é preciso consultar usuários reais a cada etapa de desenvolvimento;

De forma geral, Personas são um meio muito eficaz de comunicação interna da equipe. Na Microsoft, por exemplo, esse método é muito utilizado nos projetos de software. Eles criam

cartazes atraentes comparando as características das personas, imprimem camisetas, bonés e até mesmo canecas com os seus rostos, tudo para lembrar constantemente a equipe do foco do projeto.

Outro ponto forte no uso de Personas é sua capacidade de concisão para apresentar resultados de pesquisa.

CUIDADOS

Apesar de constituir um método eficiente de representação de um Papel-de-usuário, Personas que não se baseiam em pesquisas rigorosas podem ser manipuladas para defender "pontos de vista" de membros da equipe de projeto. A tentativa de criar e alterar uma Persona de acordo com o que for mais cômodo para a equipe, ou para um profissional em particular, pode ser desastrosa. É possível que criadores das Personas, por exemplo, se sintam tentados a usar caricaturas para torná-las ainda mais atrativas. Uma Persona muito usada com esse intuito é a chamada "minha mãe" ou "minha avó". Muita gente já deve ter ouvido frases do tipo: "isso aí nem minha mãe conseguiria usar!".

Em geral, as pessoas acham mais fácil usar esses termos, porque podem assim julgar pelo senso-comum. Trata-se de um mero artifício retórico para parecer que se está preocupando com o usuário. Na maioria das vezes, a "mãe" (ou a "avó") nem faz parte do público-alvo da interface em questão e, mesmo quando faz, sua capacidade de representar a totalidade do público acaba sendo exagerada. Isso é muito usado porque é uma figura familiar, fácil de compreender. Se você fala da sua mãe, eu consigo imaginar melhor como ela vai reagir do que uma figura abstrata representando "mulheres entre 40 e 55 anos". Entretanto, essa abordagem corre o risco de distanciar as decisões da realidade, desviando o sentido original das Personas, que é orientar o desenho centrado em usuários reais.

De que adianta saber somente que o usuário tem entre 30 e 45 anos na hora em que você vai definir o fluxo de uma determinada tarefa? É muito mais interessante saber como essas pessoas agem em situações parecidas.

Alan Cooper, criador da técnica de Personas, recomenda que sejam feitas sempre entrevistas, observações, testes de usabilidade, card-sorting, análises das tarefas, leituras e estudos etnográficos etc. As Personas vêm depois, para resumir tudo aquilo que costuma ser colocado em extensos relatórios. Vale ressaltar que cada detalhe da persona deve estar muito bem embasado em dados reais, não em meras presunções.

Personas não é uma técnica de coleta de dados, mas sim uma técnica de agrupar e apresentar resultados de pesquisas. Quando são tratadas como fim e não meio, acontece toda uma distorção, pois a pesquisa passa a ser enviesada para sustentar os perfis, muitas vezes definidos antes mesmo que ela aconteça.

4.3.3.2 Categorização de Personas

A subdivisão de Personas em categorias permite que, no projeto, sejam focados os usuários cujas necessidades são as mais importantes. A distribuição das categorias (e a nomenclatura delas) pode variar de um autor para outro na literatura, mas o mais importante é que sejam definidos critérios de priorização para elas.

PERSONAS PRIMÁRIAS

Representam os usuários mais críticos no projeto. Os seus objetivos têm de ser satisfeitos, ou o risco de que a experiência da maioria dos usuários ao utilizar o sistema seja frustrante aumenta muito. Cada persona primária requer uma interface única, a fim de que suas metas sejam efetivamente atendidas.

PERSONAS SECUNDÁRIAS

Personas secundárias são usuários que influenciam o *design* da interface, mas não representam necessariamente o foco principal do projeto. Personas secundárias incluem os usuários de nível iniciante, utilizadores intermitentes e até mesmo os peritos. Uma interface que satisfaz apenas as necessidades das Personas secundárias irá frustrar e confundir as primárias. As necessidades únicas das Personas secundárias, no entanto, devem ser atendidas, a fim de tornar os produtos viáveis para um grande número de usuários no mundo real.

PERSONAS SUPLEMENTARES

São usuários que precisam ser representados, porque sua participação influencia no projeto da interface, ainda que não façam parte diretamente dos usuários focais do projeto.

4.3.3.3 Criação de Personas

Dois aspectos são muito importantes durante um trabalho de Análise de contexto de uso de um sistema utilizando a técnica de Personas:

- Entender bem o que os usuários fazem e dizem;

- Definir a metodologia de pesquisa que será utilizada;

O QUE OS USUÁRIOS DIZEM E O QUE FAZEM

A importância maior de se estudar minuciosamente o que as pessoas dizem e aquilo que elas fazem está no fato de que “o que as pessoas dizem fazer não é necessariamente o que elas fazem” (Mulder, 2006).

Em testes de usabilidade, por exemplo, os usuários muitas vezes claramente lutam tentando realizar uma determinada tarefa, mas, ao final, acabam dizendo que a tarefa foi fácil. Portanto, para entender bem os usuários de um sistema e realizar uma boa atividade de Personas, é preciso entender ambos os aspectos.

Alan Cooper incentiva o uso de técnicas etnográficas para o estudo dos usuários, porque tais técnicas assumem que as atitudes de um sujeito e seus comportamentos são tão habituais que podem estar no inconsciente de cada um. Ao invés de perguntar o que querem os usuários, é mais eficaz concentrar-se naquilo que os usuários fazem, o que os deixa frustrados, e o que lhes dá satisfação. Combinando entrevistas com a observação direta dos usuários é possível obter uma grande quantidade de dados muito rapidamente. Se houver tempo e orçamento, as descobertas poderão ser verificadas com levantamentos quantitativos ou outras técnicas, mas elas não podem substituir a observação direta.

O QUE OS USUÁRIOS DIZEM

Atitudes revelam como as pessoas percebem a si próprias e suas experiências. Entrevistas e pesquisas com questionários são métodos comuns para pesquisar o que as pessoas dizem e para aprender sobre seus objetivos e atitudes. Aquilo que as pessoas dizem revela seus objetivos, ou seja, aquilo que elas querem fazer. Se os objetivos não forem bem entendidos, não será possível saber o que melhorar e como melhorar no projeto de uma interface.

O QUE OS USUÁRIOS FAZEM

O comportamento das pessoas revela aquilo que elas fazem. Através da análise do comportamento, é possível saber, por exemplo, onde os usuários podem estar tendo problemas na realização de suas atividades. As análises feitas em um teste de usabilidade, por exemplo, compreendem também o comportamento do usuário enquanto utiliza uma

determinada interface. A observação ajuda a minimizar os chamados "comportamento auto-reportados", que muitas vezes são imprecisos.

4.3.3.4 Metodologias de pesquisa

Nesta subseção, serão apresentadas as principais metodologias de pesquisa utilizadas para a construção das Personas, detalhadas no livro "The User is Always Right: A Practical Guide to Creating and Using Personas for the Web", de Steve Mulder. Para cada uma das metodologias, serão abordadas três etapas básicas de construção das Personas: condução da pesquisa, segmentação dos usuários baseada na pesquisa e definição das Personas para cada segmento. Ao final da apresentação das metodologias, será apresentada uma tabela comparativa entre elas, com diretrizes sobre quando utilizar cada uma das abordagens.

4.3.3.5 Pesquisa qualitativa

É um tipo de pesquisa que visa descobrir informações novas sobre um pequeno conjunto de amostra. Essa metodologia é utilizada quando o estudo para formar as Personas é feito com um conjunto pequeno de usuários (de 10 a 20).

Uma desvantagem da pesquisa qualitativa é que ela não prova nada definitivamente, já que os dados não são analisados por meio de técnicas estatísticas. Esse tipo de metodologia é utilizado para definir um problema, gerar hipóteses sobre ele, identificar alguns determinantes e desenvolver meios de pesquisa quantitativa. A grande vantagem é que esse tipo de pesquisa pode ser feita de forma relativamente rápida. Abaixo apresentamos os passos principais na condução de uma pesquisa qualitativa.

PASSO 1: CONDUÇÃO DA PESQUISA

Neste passo, são realizadas entrevistas com usuários, que representam a forma mais comum de pesquisa qualitativa. É feito também um trabalho de análise do ambiente dos usuários, em que ocorre a observação de suas atividades e comportamentos no ambiente real de utilização do futuro sistema. Enquanto se observa o comportamento dos usuários, questionamentos podem ser feitos sobre seus objetivos e atitudes.

Você saberá que pode parar de entrevistar quando se pode prever como cada usuário irá responder. Isso significa que padrões estão começando a surgir. Assim que as entrevistas forem terminadas, devem ser listadas todas as variáveis comportamentais, ou seja, as diferentes formas de comportamento dos entrevistados. A maioria das variáveis pode ser representada como intervalos com dois extremos. Em um domínio de um shopping, por

exemplo, poderá haver variáveis tais como a freqüência de compras, o grau de diversão, a importância para o preço e o serviço de orientação. Também poderá haver variáveis demográficas que pareçam afetar o comportamento dos usuários, tais como a idade ou a habilidade técnica. Mas é importante ser cuidadoso ao focar na demografia durante a criação das Personas, uma vez que variáveis comportamentais vão ter muito mais impacto sobre o desenho das interfaces.

PASSO 2: SEGMENTAÇÃO DOS USUÁRIOS

Segmentação é a arte de pegar muitos pontos e criar agrupamentos que podem ser descritos em função de aspectos comuns entre os membros de um grupo. Em Personas, o objetivo é encontrar padrões que permitam agrupar pessoas com perfis similares juntas em determinados “tipos de usuários”. Essa segmentação das Personas é normalmente baseada em seus objetivos, atitudes e/ou comportamentos.

Porém é importante ressaltar que segmentos não são personas. Segmentos são listas de características, possivelmente suportadas por planilhas eletrônicas de números. Personas são personificações dessas características, uma transformação dessas listas de fatos e características em histórias que outras pessoas entendam rapidamente, relembrarem e utilizem, agindo de acordo com elas.

Como criar esses agrupamentos é absolutamente crítico, e talvez, a parte mais difícil durante a criação das personas. Isso porque se o principal aspecto da definição de personas, a segmentação, não é claro ou útil, a utilização de personas não se torna um hábito.

Não há uma maneira sempre certa de fazer a segmentação. Segmentação não é uma ciência. Pense em segmentação como a arte de encontrar padrões e histórias nos dados. O processo de segmentação é colaborativo e iterativo. Colaborativo, porque independentemente da abordagem, terá que envolver todos os interessados (stakeholders) chaves de todas as partes da organização: donos dos produtos, marketing, desenho, serviços de clientes, vendas e outros que interagem com os usuários. Iterativo, porque todo ano pesquisas adicionais são conduzidas com os usuários reais para validar sua segmentação inicial. Mudanças de negócio, de ambiente e de usuários implicam na necessidade de validar as personas iniciais para verificar se ainda estão válidas.

Para optar por uma segmentação particular, algumas perguntas precisam ser feitas, por exemplo: os segmentos explicam as diferenças chaves que têm sido observadas? Conversando com os usuários, são notadas as diferenças entre indivíduos: o que eles fazem, como eles fazem, o que eles pensam e/ou quem eles são? A abordagem de segmentação utilizada mostra essas diferenças-chave? Os segmentos são suficientemente diferentes uns

dos outros? Se a única diferença entre dois segmentos é a idade dos usuários, então eles representam o mesmo segmento. A segmentação pode ser descrita rapidamente? É melhor encontrar um, dois ou três fatores que melhor define cada segmento, descrever bem esses fatores e simplificar ligeiramente visando aumentar a compreensão. Os segmentos cobrem todos os usuários? Deve estar claro que todo usuário entrevistado se ajusta a um dos segmentos que está sendo explorado. A segmentação pode ser descrita rapidamente? É melhor encontrar um, dois ou três fatores que melhor define cada segmento, descrever bem esses fatores e simplificar ligeiramente visando aumentar a compreensão. Os segmentos cobrem todos os usuários? Deve estar claro que todo usuário entrevistado se ajusta a um dos segmentos que está sendo explorado.

Na metodologia de Personas qualitativa, a segmentação também é um processo qualitativo. Tratamos aqui menos de ciência e mais sobre o público. É importante rever aqui as anotações feitas no Passo 1 e escutar novamente os usuários, quando necessário. No desenvolvimento de um site de compras, por exemplo, os usuários podem ser entrevistados e, então, segmentados, baseando-se em seus objetivos: comprar uma casa, encontrar um apartamento, vender uma casa etc.

ATRIBUTOS QUE DEVEM SER CONSIDERADOS NA SEGMENTAÇÃO:

Recomenda-se utilizar atributos que revelem como as pessoas atualmente usam o produto. No caso de um site, são eles: objetivos do usuário ao usar o produto: o que os usuários querem fazer. Comportamento: como eles fazem isso? Atitudes: como eles percebem a experiência ou eles mesmos? Comece segmentando por objetivos. Caso a primeira abordagem não seja a mais adequada, segmente por ciclo de vida de uso. Se a segunda não for melhor opção, segmente baseando-se em uma combinação de comportamento e atitudes.

Outro exemplo: imagine que você tenha um conjunto de rochas. Sua missão é descrever os diferentes tipos de rochas nesses conjuntos. Você pode organizar as rochas pelo tamanho e, então, descrever cada grupo. Você pode agrupar pela cor ou textura, ou você pode considerar a classe geológica e então agrupar pelo tipo (sedimentar, ígnea etc.). Organizar todos essas rochas individuais em clusteres é tudo o que a segmentação faz. Todos os usuários são únicos. Mas, para falar sobre quem são seus usuários e o que fazem com seus conhecimentos, você tem que agrupá-los em segmentos que no final serão transformados em Personas.

PASSO 3: DEFININDO PERSONAS

Neste passo, são criadas Personas para cada segmento. Cada “tipo” de usuário é representado por uma Persona, de acordo com os detalhes levantados envolvendo seus objetivos, comportamentos e atitudes. As Personas se tornarão realísticas quando forem relacionadas a elas um nome, uma foto, informações demográficas, cenários etc.

CRIANDO O NOME

A escolha do nome é importante porque de certa forma sintetiza o caráter da Persona. Por ser importante, o processo de escolha do nome deve ser colaborativo. Pessoas têm respostas diferentes para nomes diferentes. Dessa forma, é interessante encontrar um nome que todos sintam ser o nome certo para a Persona. Você saberá que suas Personas estão funcionando quando os membros da equipe na empresa começarem a referenciá-la pelo nome.

ESCOLHENDO A FOTO

Personas não adquirem vida sem foto. A foto que você escolhe tem um forte impacto em como os outros verão as Personas. Assim como o nome, a escolha da foto é um processo colaborativo. Quando se está trabalhando com a abordagem de segmentação, todos começam a formar uma imagem mental de quem são essas pessoas. Dessa forma, pode levar tempo para encontrar uma foto que todos concordem como sendo verdadeiramente a Persona representada. Não use pessoas que são modelos, ou que, de alguma maneira, podem ser vistas como modelo. É importante escolher características realísticas que representem usuários reais. A escolha de fotos que não representem pessoas reais, com falhas reais, terminará com a criação de perfis de usuários que podem causar confusões durante as tomadas decisões (Chapman, 2006).

Não se preocupe em escolher fotos imperfeitas, porque elas são para uso interno. Entretanto, para qualquer imagem utilizada, tenha certeza de que ela é a certa para o que você pretende. Evite poses fotográficas. Uma face sorrindo para a câmera é agradável, mas se a imagem for cuidadosamente planejada ou não natural ela não é adequada para a Persona. Assim como o nome, fotos devem ser apropriadas para a personalidade que está sendo criada. Pense nas roupas que a pessoa costuma vestir, no estilo de cabelo, na maquiagem e constituição, entre outros.

Escolher uma foto significa escolher uma idade, mas não fique confinado a isso. Não é interessante ter quatro personas com a mesma idade. Opte pela diversidade, particularmente de raça.

Recomenda-se usar rosto e ombros para as fotos, porque a face captura um excelente nível de detalhes de personalidade. Certifique-se de escolher fotos em que as pessoas estão olhando na direção da câmera fotográfica e não que estejam de perfil. Você deve ser capaz de ver os rostos das Personas claramente. Não é bom que a foto tenha outras pessoas ou objetos próximos. Isso pode distrair aqueles que irão utilizá-la. Use fotos com cores, com alta resolução e que não sejam distorcidas quando forem impressas. Isso porque você usará as fotos de várias maneiras (impressa e em apresentações).



(a) Usuário com atenção dispersa



(b) Usuário acompanhado

Figura 4-3 Fotos Ruins



Figura 4-4 Fotos Boas

NÚMERO DE PERSONAS

Recomenda-se de 3 a 7 personas (Cooper, 2004). São recomendadas de 3 a 6 personas por sítio Web. Com menos de duas personas você está perdendo algumas diferenças chaves entre os usuários. Com mais de seis, provavelmente você pode combinar as personas. Além disso, podem ser desenvolvidas personas demais para a equipe mantê-las corretas. Quanto mais personas, mais risco de interesses conflitantes e confusões.

4.3.3.6 Pesquisa quantitativa

Pesquisa quantitativa é uma metodologia de pesquisa utilizada para testar ou provar alguma coisa com tamanho de amostra grande. As informações e dados coletados são analisados estatisticamente, através de métodos conhecidos. De maneira prática, nessa abordagem são testadas uma variedade de modelos de segmentação, com o objetivo de encontrar o modelo que é mais adequado e útil para criar as Personas.

Sua vantagem é oferecer uma compreensão melhor e mais realista da amostra de usuários que está sendo estudada. Esse tipo de pesquisa é muito utilizado na análise do tráfego em sites. Por exemplo, dizer que 35% dos sites nunca foram visitados é uma informação com considerável grau de precisão.

Para que sejam criadas as Personas através de uma metodologia de pesquisa quantitativa, são realizados cinco passos, descritos abaixo.

PASSO 1: CONDUÇÃO DA PESQUISA

De forma semelhante ao que ocorre na pesquisa qualitativa, são analisados aqui também os objetivos, comportamentos e atitudes dos usuários. A pesquisa pode ser feita por meio de questionários e entrevistas, no ambiente real de utilização do futuro produto.

PASSO 2: FORMULANDO HIPÓTESES PARA A SEGMENTAÇÃO

São levantadas as várias potenciais maneiras de segmentar os usuários, de acordo com os dados levantados no passo anterior. O objetivo é uma fazer uma lista de uma variedade de candidatos que poderão ser analisados.

PASSO 3: REUNINDO DADOS PARA A SEGMENTAÇÃO

O objetivo aqui é reunir mais dados para a etapa de segmentação. Para cada potencial opção de segmentação, há questões particulares que precisam ser respondidas através de

questionários, ou questões que precisam ser respondidas através da análise do tráfego de um site. Por exemplo: histórias de usuários sobre a experiência de utilização de um site pode ser uma maneira de segmentar. Sendo assim, uma questão do questionário preparado para a pesquisa pode abordar quanto tempo e com que freqüência os usuários utilizam o site.

Se uma longa lista de opções de segmentação estiver sendo avaliada, para ter os candidatos mais promissores, uma estratégia muito útil é colocar as variáveis em um gráfico. Da lista de variáveis, selecionam-se duas. Coloque um atributo no eixo vertical e outro na horizontal e um gráfico será construído com quatro quadrantes. Cada quadrante representa uma possível configuração das pessoas que combina as duas características. A partir da segmentação realizada, decide-se se cada quadrante pode ser uma persona. Pode-se descrever os segmentos criados para cada quadrante. Então veja se a segmentação passa pelo teste de qualidade da segmentação apresentado no passo de segmentação da pesquisa qualitativa. Se não passar pelo teste, tenta-se uma combinação diferente de comportamento e atitudes em uma nova matriz até resultar em um modelo de segmentação útil.

Não é necessário restringir sua segmentação a duas variáveis, entretanto recomenda-se utilizar poucas variáveis, sendo de preferência duas. Quanto mais variáveis usar para fazer a segmentação, mais difícil fica entender o gráfico montado e lembrar as histórias que se constrói com a persona. Lembre-se de que você não está tentando descobrir um modelo de segmentação correto de forma mágica. Sua missão é explorar todas as potenciais abordagens de segmentações, e pela avaliação de cada uma, determinar qual será a mais útil para a sua situação específica.

PASSO 4: SEGMENTANDO COM CLUSTERIZAÇÃO

Neste passo, são utilizados algoritmos estatísticos para guiar o processo de segmentação. Coloca-se um conjunto de variáveis em uma ferramenta, que procurará ocorrências de clusters (agrupamentos) baseadas em alguns conjuntos de associação. É possível terminar com um número de clusters e um número de atributos como diferenciadores chaves entre os clusters.

Esse processo é um pouco mais complexo e muito influenciado pela maneira como ele é executado. É显著mente diferente das outras abordagens, porque a segmentação é dirigida por dados bem como dirigida por humanos.

PASSO 5: DEFININDO PERSONAS

Com a análise da semelhança dos clusters com os segmentos, os dados são transformados em reais através dos processos de: adicionar nome, foto, histórias etc. A planilha de dados é transformada em informações reais de pessoas.

4.3.3.7 Pesquisa qualitativa com validação quantitativa

É uma metodologia de pesquisa que combina aspectos qualitativos e quantitativos para a construção das Personas.

PASSO 1: CONDUZINDO A PESQUISA

De forma semelhante ao que ocorre na pesquisa qualitativa, são também utilizadas entrevistas e questionários com os usuários, além de estudos de campo que revelem hipóteses sobre os objetivos, comportamentos e atitudes dos usuários.

PASSO 2: SEGMENTAÇÃO DOS USUÁRIOS

A segmentação de usuários, por sua vez, é baseada em pesquisa quantitativa (ver Passos 2, 3 e 4 da Pesquisa quantitativa). São preparados alguns tipos de segmentação que resultarão em um número de segmentos baseados em objetivos, comportamentos e ou atitudes particulares de usuários.

PASSO 3: TESTE DE SEGMENTAÇÃO

Neste passo, inicialmente, através de questionários ou outra forma de pesquisa qualitativa, testa-se o modelo de segmentação usando um tamanho grande de amostra (para assim verificar se o modelo criado reflete exatamente a realidade). Os dados da pesquisa levantados com os questionários são melhores para avaliar objetivos e atitudes dos usuários.

A análise dos dados pode ser feita por meio de simples cruzamentos, ou então utilizar técnicas de análise quantitativa mais complexas. Se o objetivo for, por exemplo, testar o modelo de segmentação em função dos objetivos dos usuários, o questionário deve conter uma questão do tipo: "qual a razão dos usuários visitarem o site?".

PASSO 4: NOVO TESTE DE SEGMENTAÇÃO

Neste passo, sobretudo, é preciso observar se existem diferenças significativas no perfil dos usuários que dão suporte ao seu planejamento de segmentação. Se sim, então ele foi um

sucesso; caso contrário, será preciso tentar outras maneiras de segmentar os usuários e testar a segmentação.

PASSO 5: DEFININDO AS PERSONAS

A criação das Personas é baseada nas análises dos dados qualitativos. As Personas precisarão ser segmentadas de acordo com alguns fatores. É necessário fazer uma pesquisa para cada usuário do modelo de segmentos. Posteriormente, devem ser analisados os resultados para cada usuário da segmentação, também de forma separada. Daí em diante serão feitas as atividades para tornar as Personas realísticas, com a escolha de nomes, fotos, informações demográficas, cenários etc (Rever Passo 3 da Pesquisa qualitativa).

4.3.4 DETALHAMENTO DAS ATIVIDADES DO SUBPROCESSO

Esta seção apresenta um detalhamento das atividades de modelagem dos usuários introduzida na seção 4.3. É importante notar que diferentes formas ou técnicas de representação dos papéis de usuários podem ser usadas no processo apresentado e que poderia ser necessária uma adaptação da técnica utilizada para que seja mapeada nas atividades de modelagem em duas etapas (preliminar e detalhada) como sugerimos nas seções 4.3.4.1 e 4.3.4.2 abaixo. Sendo assim, a técnica de Personas sugerida para a modelagem de papéis de usuários na seção 4.3.3 poderia ser mapeada em atividades de modelagem preliminar e detalhada. Claro que, também, é possível considerar a atividade de modelagem dos usuários em nosso processo como constituída de uma única atividade que seria realizada de acordo com a técnica utilizada. No entanto, a proposta de se fazer a modelagem dos usuários em duas etapas (preliminar e detalhada) normalmente será uma prática saudável já que a atividade de modelagem é por natureza iterativa.

Um dos aspectos a serem levantados, e que merece um destaque especial, são os modelos mentais que os usuários utilizam na realização de suas atividades. A descrição dos modelos mentais foi considerada como parte da modelagem dos usuários já que esses modelos são abstrações que os usuários utilizam na realização de suas atividades. No entanto, como esses modelos estão envolvidos na realização das atividades dos usuários, o levantamento desses modelos poderia também ser considerado como parte da Análise de tarefas.

Realmente, com já mencionamos, há uma grande interdependência entre os diversos tipos de Análise de contexto de uso. Sendo assim, é normal que um Modelo Mental de um usuário

seja identificado na análise de uma tarefa que este usuário realiza e é aceitável, ou até indicado, que este modelo mental seja caracterizado e mencionado na descrição da tarefa.

LEVANTAMENTO DOS MODELOS MENTAIS

O Levantamento dos modelos mentais visa à descrição dos modelos mentais mais importantes que as pessoas envolvidas (potenciais usuários) utilizam na realização de suas atividades. A descrição pode ser textual, se necessário com a utilização de figuras explicativas.

Na realização de suas atividades no dia a dia, as pessoas criam e utilizam modelos mentais que explicam e ajudam no controle dos objetos com os quais interagem. Por exemplo, quando estamos dirigindo um veículo, utilizamos modelos mentais que nos explicam como acelerar, como frear, como fazer uma curva, etc. Utilizando esses modelos mentais, conseguimos controlar o carro. O modelo mental não necessita ser preciso nem mesmo ser coincidente como o modelo real de como funciona o objeto, basta ser compatível. Um motorista precisa ter a noção de que quando pressiona o pedal de freio, uma força, proporcional à pressão que aplica no pedal, é transmitida à roda. Mas ele não precisa conhecer o mecanismo real que realiza a frenagem do veículo.

A descrição pode ser textual, se necessário com a utilização de figuras explicativas.

O conceito de modelos mentais foi apresentado na seção 2.5. O levantamento desses modelos pode ser feito durante as atividades de modelagem preliminar e modelagem detalhada dos usuários, descritas a seguir.

4.3.4.1 Modelagem preliminar de usuários

Esta atividade consiste na criação preliminar de um modelo de usuários onde se define quais são os atores que interagirão com o produto e como esses atores se relacionam entre si. As principais atividades realizadas são:

- identificação de atores a partir dos grupos de usuários levantados;
- análise preliminar da forma de trabalho ou do comportamento relacionado com o produto, das pessoas representadas por cada Ator.
- ordenação e simplificação inicial do modelo de atores com base no relacionamento entre eles.
- Identificação dos atores focais, com a finalidade de definir quais atores, daqueles que representam a população alvo, são mais importantes e devem ser considerados com

mais atenção no desenvolvimento do sistema. Os atores focais são aqueles considerados particularmente importantes do ponto de vista do negócio da empresa cliente ou de riscos, ou em função de alguma consideração técnica.

A Tabela 4-4 apresenta um detalhamento das atividades de modelagem preliminar dos usuários.

Descrição	Modelagem preliminar
Insumos	1 Especificação de requisitos de Software (ersw) 2 Documentação do planejamento da análise 3 Lista preliminar de usuários potenciais e respectivas categorias. 4 Informação já levantada no trabalho de campo. 5 Manuais de usuários (de sistemas similares, se existirem, ou do atual, a ser melhorado).
Tarefas	1 Identificação dos atores a partir das categorias de usuários levantadas; 2 Realização do trabalho de campo de observação e consulta às pessoas envolvidas visando à modelagem de usuários. 3 Levantamento das características principais dos atores, identificando: <ul style="list-style-type: none"> ◦ experiência no trabalho, nível educacional, necessidades de treinamento; ◦ idade, sexo e diferenças físicas que podem ser significantes; ◦ localidades geográficas, cultura e nacionalidades; ◦ linguagem usada, terminologias; ◦ nível de trabalho, tais como técnicos e especialistas; ◦ o modo de trabalhar do usuário; 4 Verificação das diferenças relevantes entre as características levantadas dentro de grupos de usuários. (preliminar). Verifica se as características similares indicam diferenças significativas que não tinham sido observadas, considerando-se várias perspectivas, por exemplo, a distribuição esperada de freqüência de utilização (preliminar) do produto em consideração. 5 Ordenação, simplificação ou generalização de atores do modelo de usuários com base no relacionamento entre eles (preliminar). 6 Determinação dos atores focais considerando-se: <ul style="list-style-type: none"> ◦ freqüência de uso; ◦ os processos de negócio; ◦ os riscos associados a custo e utilização do produto. Obs.: Os critérios mencionados acima podem ser considerados isoladamente ou em conjunto utilizando-se uma metodologia como QFD, por exemplo. Esta última abordagem é mais precisa, onde se determina a importância de cada critério, utilizando-se, por exemplo, o método Analytic Hierarchy Process (AHP) (Cheng & Scapin, 1995). O grau de importância de cada Ator é, então, calculado multiplicando-se o valor de correlação existente entre os critérios com a importância de cada critério e somando-se os produtos resultantes.
Resultados	1 Modelo de Atores Humanos preliminar

Tabela 4-4 Atividades da modelagem preliminar dos usuários

4.3.4.2 Refinamento da modelagem de usuários

Esta atividade visa um refinamento e melhoria do modelo de usuários. Faz-se o detalhamento dos atores, descrevendo suas necessidades, interesses, expectativas e comportamentos e faz-se a identificação dos relacionamentos envolvendo esses atores. Os atores focais devem receber mais atenção no trabalho de desenvolvimento da interação. Neste trabalho de modelagem detalhada, o modelo de usuários pode ser reorganizado com a criação ou extinção de atores ou com a redefinição de relacionamentos entre os atores. A organização e simplificação mais apurada do modelo de usuários são realizadas com base em um trabalho de consulta e validação dos dados levantados.

Descrição	Refinamento da modelagem de usuários
Insumos	1 Especificação de requisitos de Software (ersw) 2 Documentação do planejamento da análise 3 Modelagem preliminar de atores humanos. 2 Informação levantada no trabalho de campo. 4 Manuais de usuários (de sistemas similares, se existirem, ou do atual, a ser melhorado).
Tarefas	1 Realização do trabalho de campo de observação e consulta às pessoas envolvidas visando à modelagem de usuários. 2 Detalhamento da caracterização dos atores, principalmente para os atores focais, descrevendo: <ul style="list-style-type: none"> ◦ proficiência: como a proficiência de uso é distribuída em certo intervalo de tempo e entre os usuários de um determinado papel; ◦ interação: quais são os padrões de uso associados com um determinado papel. ◦ informação: qual a natureza da informação manipulada pelos usuários em um determinado papel ou o intercâmbio entre os usuários e o sistema; ◦ critério de usabilidade: qual a importância relativa de objetivos de usabilidade específicos em relação a um dado papel; ◦ suporte funcional: funções específicas, características ou facilidades necessárias para usuários em um papel específico. ◦ risco operacional: tipo e nível de risco associado com a interação usuário-sistema para um Papel-de-usuário específico; ◦ restrições de dispositivos: limitações ou restrições de um equipamento através do qual um usuário, exercendo um determinado papel, interage com o sistema; ◦ fatores relevantes do ambiente de trabalho, incluindo aspectos físicos, culturais e sociais, no qual um usuário interage com um sistema; ◦ como os usuários aplicam o conhecimento e experiência na realização das tarefas (modelos mentais). 3 Melhoria e conclusão da modelagem de papéis de usuários validando os relacionamentos previamente levantados ou identificando novos relacionamentos envolvendo os atores. Este trabalho pode levar ao desdobramento de atores considerados complexos em atores mais simples ou à fusão ou generalização de

	<p>atores com base no relacionamento entre eles e suas características.</p> <p>4 Verificação das diferenças relevantes entre as características levantadas dentro de grupos de usuários. Verifica se as características similares indicam diferenças significativas que não tinham sido observadas, considerando-se várias perspectivas, por exemplo, a distribuição esperada de freqüência de utilização.</p> <p>5 Ordenação, simplificação ou generalização de atores do modelo de usuários com base no relacionamento entre eles.</p>
Resultados	<p>1 Descrição de Características de Atores Humanos completa.</p> <p>2 Modelo de Atores Humanos completo.</p>

Tabela 4-5 Refinamento da modelagem de usuários

A Tabela 4-5 apresenta as atividades de refinamento da modelagem de usuários.

4.4 ANÁLISE DE TAREFAS

A Análise de tarefas visa caracterizar as tarefas realizadas pelos usuários ou (futuros usuários) em suas atividades relacionadas com o produto em consideração. Note que “tarefas” aqui têm a conotação de atividades que as pessoas realizam ou vão realizar, podendo ser uma atividade de lazer ou qualquer outro tipo de atividade relacionada com o produto em consideração.

Como já mencionamos, a Análise de tarefas é bastante relacionada com a Análise de usuários; em geral, essas atividades são realizadas em um processo incremental e iterativo, onde a Análise de tarefas dá subsídios para a Análise de usuários, inclusive sugerindo novos Papéis-de-usuários a serem considerados para serem associados ao produto, e vice-versa.

Os sistemas de software são criados para apoiar a realização de tarefas, daí a importância da caracterização dessas tarefas antes do desenvolvimento de um software. Note que, muitas vezes, um sistema apóia apenas parcialmente a realização de uma tarefa, o restante da tarefa é realizado manualmente ou com a ajuda de outros meios.

As tarefas a serem realizadas pelo sistema, quando o produto estiver em operação em interação com os usuários, correspondem aos casos de uso. Sendo assim, a Análise de tarefas é importante não só para propiciar o desenvolvimento de uma interface do usuário adequada, como também para ajudar na definição e priorização das funções (casos de uso) que são ou serão providas pelo software em consideração. As outras tarefas realizadas pelos usuários são consideradas fora do escopo do produto em consideração, mas a Análise de

tarefas como um todo constitui informação essencial para o desenvolvimento da interação humano-computador.

EXEMPLO PARA MOTIVAÇÃO

Segue um exemplo interessante que ilustra bem a importância da Análise de tarefas no desenvolvimento de um produto de software. Uma empresa fornecedora de água para a população resolveu desenvolver um sistema para apoiar a necessidade de leitura de medidores para a produção das contas. A Figura 4-5 a seguir mostra o roteiro que um leitourista segue na realização de suas medições sem o apoio de um sistema. No seu trabalho, o leitourista percorre a rua uma vez só, fazendo a leitura nas residências dos dois lados, quando necessário tocando a campainha (para chamar o dono) de várias casas ao mesmo tempo.



Figura 4-5 Roteiro normalmente seguido pelo leitourista

Um sistema foi desenvolvido para o apoio à medição do consumo de água das residências. O sistema utiliza uma tecnologia que permite que a conta seja impressa logo após a leitura, Figura 4-6.



Figura 4-6 Leiturista com equipamento de registro de leituras de conta

No entanto, com o objetivo de “otimizar” a leitura, o sistema impõe uma seqüência de domicílios a serem visitados. A próxima figura mostra um roteiro que o leiturista segue na realização de suas medições conforme exigido pelo sistema de apoio. Observe que o leiturista percorre cada rua nos dois sentidos.

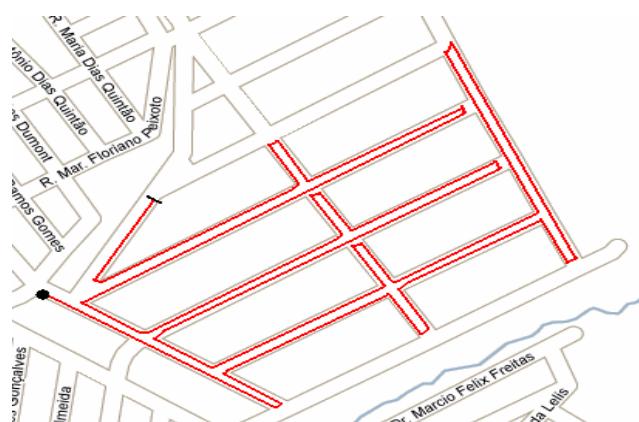


Figura 4-7 Leiturista com equipamento de registro de leituras de conta

Apesar de, à primeira vista, representar um trajeto otimizado, a experiência dos usuários não foi levada “em conta”. O resultado foi que a solução com o apoio do sistema automatizado se mostrou ineficiente na prática, requerendo muito mais tempo (da ordem do dobro) para a realização das leituras. O sistema está sendo rejeitado pelos usuários, apesar dos avanços que oferece, por tornar o trajeto mais longo e a execução da leitura mais demorada.

4.4.1 VISÃO GERAL

A atividade de Análise de tarefas prescreve que, inicialmente, realizemos a Análise de Necessidades. A Análise de Necessidades pode ser considerada como uma análise dos

interesses e das necessidades das pessoas envolvidas, não estritamente como uma Análise de tarefas. No entanto, tarefas são realizadas, em última análise, para suprir necessidades dos seres humanos e o entendimento dessas necessidades é muito importante para o entendimento das motivações por trás da realização de tarefas.

A descrição das tarefas dos usuários pode ser feita de diversas formas, cada uma delas correspondendo a um modelo que propicia uma determinada visão da questão. Dentre as diversas formas, devem ser escolhidas aquelas que sejam mais convenientes dadas as características das tarefas a serem modeladas. Algumas formas de definição de tarefas são listadas abaixo.

1. Fluxo de trabalho
2. Trabalho individual
3. Seqüência de tarefas
4. Hierarquia de tarefas
5. Procedimentos

É importante lembrar que o objetivo da Análise de tarefas é o de caracterizar a situação existente antes do desenvolvimento do software. Não cabe, na Análise de tarefas, definir-se soluções de desenho da interface com o usuário, já que a Análise Tarefas visa justamente levantar informações importantes para o posterior desenho da interface. Devemos, no entanto, como resultado do trabalho de Análise de tarefas, bem como em outras atividades da Análise de contexto de uso, produzir recomendações para as atividades subsequentes no desenvolvimento do software, incluindo, com ênfase, recomendações para o desenho da interface com o usuário.

Diferentes técnicas podem ser usadas para a modelagem de tarefas, por exemplo, as formas mais descritivas propostas por Hackos, JT & Redish (Hackos, JT & Redish, 1998) ou modelos mais simples como proposto por Hix e Hartson (Hix, D.; Hartson, H. R., 1993). Aqui, preferimos sugerir a técnica de Roteiros de Rosson e Carroll (Rosson e Carroll, 2002), apresentada na seção 4.4.3, por ser considerada barata, fácil e, de certa forma, intuitiva e lúdica, para a equipe de desenvolvimento de um software.

De certa forma, podemos traçar um paralelo entre a técnica de Roteiros usada na descrição de tarefas e a técnica de Personas usada na modelagem de usuários, no sentido de que ambas se utilizam de narrativas que constituem uma espécie de cenário ilustrativo da situação, que se utiliza de histórias com um espírito de certa forma lúdico.

4.4.2 OBJETIVOS

A Análise de tarefas visa à obtenção de conhecimento detalhado das atividades realizadas pelos usuários para que se possa desenvolver uma solução de interação adequada para o tipo de utilização que se espera do produto.

A Análise de tarefas de um sistema interativo tem como objetivos específicos:

- conhecer os interesses, motivações e necessidades dos usuários relacionados às tarefas que eles realizam que têm relação com o produto em consideração;
- caracterizar as tarefas em todos os aspectos, como freqüência de realização, modo de interação envolvendo os usuários, etc, que possam ter importância para o desenvolvimento da interface do usuário;
- gerar um modelo (descrição) de tarefas a partir das informações obtidas;
- contribuir para a especificação de requisitos e metas de usabilidade, na definição das tarefas para as quais o desempenho dos usuários em sua execução deverá ser medido;
- fornecer insumos para o desenho da interação, tais como a arquitetura global, a navegação e o conteúdo da interface, o leiaute e o comportamento (*look-and-feel*) dos elementos de interação;
- fornecer insumos para o planejamento e especificação das avaliações de usabilidade;

A Análise de tarefas representa um importante insumo para se conseguir agregar valor ao produto no sentido de torná-lo mais efetivo como ferramenta de apoio ao trabalho ou a outro tipo de atividade realizada pelos usuários. Abaixo são listadas algumas questões mais comuns que se procura investigar na Análise de tarefas;

- Que características das tarefas têm impacto no solução em termos da interface do usuário?
- Quais são os interesses dos usuários que possam estar relacionado com o produto em desenvolvimento?
- Quais as necessidades das pessoas envolvidas que podem ser supridas com o apoio do produto em consideração ou que tem impacto na especificação ou na solução do produto em consideração?
- O que os usuários do software irão fazer que tenha relação com o produto que está sendo desenvolvido?

- Em que os usuários necessitam do sistema para realizar algo ou alguma tarefa?
- Como o sistema deve apoiar a realização das tarefas dos usuários?
- O que no momento atual não está funcionando ou é ineficiente?
- Como o sistema seria mais efetivo para suportar as atividades do usuário?

4.4.3 ROTEIROS DE DOMÍNIO DE PROBLEMA

As metodologias de engenharia de usabilidade propõem a descrição das tarefas a serem contempladas no produto de uma forma mais ampla do que normalmente utilizado na área de engenharia de software. Por exemplo, Carroll & Rosson (2002), sugerem a utilização de roteiros na descrição das funções do produto. A técnica de Roteiros utiliza-se de histórias que descrevem um cenário onde a tarefa é realizada. Não se trata de uma simples descrição de tarefas, mas de uma descrição enriquecida com aspectos relevantes como a caracterização das pessoas envolvidas, os objetivos da tarefa, as estratégias utilizadas pelas pessoas envolvidas na realização das tarefas, etc. Estes aspectos constituem informações importantes usadas no desenvolvimento da interação para o produto em consideração.

Os Roteiros são histórias sobre pessoas e suas atividades, descrevendo situações de interesse onde pessoas realizam suas atividades. Os autores Rosson e Carroll propõem dois tipos de roteiro: os Roteiros de domínio de problema, que descrevem essas histórias no contexto de uso antes da introdução do sistema que está sendo desenvolvido, e os Roteiros de desenho da interação, que descrevem histórias no contexto de uso desenhado, isto é, mostram a tarefa sendo realizada com o apoio do sistema desenvolvido.

A Tabela 4-6 mostra os vários aspectos que devem ser registrados em um roteiro e o significado desses aspectos.

Aspectos usados no Roteiro	Significado dos aspectos usados no roteiro
Cenário	Detalhes da situação que motivam ou explicam os objetivos, ações, reações dos atores.
Atores	Papéis de pessoas que interagem com o ambiente do cenário; eventualmente com suas características relevantes.
Objetivos da tarefa	Aspectos da situação que motivam as ações realizadas pelos atores.

Modelos mentais	Modelos conceituais e atividade mental de que os atores se utilizam para converterem objetivos em comportamentos.
Avaliação	Atividade mental dos envolvidos que visa interpretar características da situação.
Ações	Comportamento observável das pessoas.
Eventos	Ocorrências externas ou internas que podem influenciar a atividade. Podem não ser percebidas pelos atores, mas são importantes para o roteiro.

Tabela 4-6 Roteiros de domínio do problema

O quadro abaixo mostra um exemplo de Roteiro que descreve um cenário onde um cliente (José) de um supermercado deseja comprar um vinho para presentear um amigo e o vendedor (James) do setor de vinhos do supermercado vai apoiar a venda

James, um vendedor especializado em vinhos, trabalha no setor de vinhos de um supermercado sofisticado. José, um cliente, deseja comprar uma garrafa de vinho para presentear um amigo. José está indeciso sobre o que comprar, deseja otimizar a relação custo-benefício.

José tem em mente um presente na faixa de R\$ 50,00 mas se sente constrangido em revelar este valor para o vendedor. James está posicionado discretamente, pronto para ajudar o cliente se solicitado.

Eventualmente, José solicita ajuda ao vendedor na escolha do vinho. James, pergunta sobre o gosto do amigo de José. Com habilidade, procura conhecer mais detalhes do perfil de vinho desejado pelo cliente, principalmente a faixa de preços que o cliente se dispõe a gastar. James sabe que, muitas vezes, o cliente se sente constrangido em deixar transparecer que entende muito pouco (quase nada) de vinhos.

James comenta sobre as várias opções de tipos de vinho, de acordo com o perfil desejado pelo cliente. Em particular, se existir, enfatiza as ofertas de produtos que se encaixem no perfil de vinho desejado pelo cliente, visando oferecer um bom serviço.

De repente, outro cliente pede ajuda ao vendedor. James pede desculpas ao novo cliente mas solicita que aguarde um momento pois está atendendo outro cliente. James não quer ser indelicado com seu outro cliente José.

Observe que os aspectos indicados na Tabela 4-6 Roteiros de domínio do problema podem ser identificados no quadro apresentado. Por exemplo, o trecho do roteiro:

James, um vendedor especializado em vinhos, trabalha no setor de vinhos de um supermercado sofisticado. José, um cliente, deseja comprar uma garrafa de vinho para presentear um amigo. José está indeciso sobre o que comprar, deseja otimizar a relação custo-benefício

apresenta o cenário onde a tarefa é executada, introduz os atores *James* e *José*, e indica que o objetivo do cliente, *José*, seria “*otimizar a relação custo-benefício*”.

Os aspectos registrados no Roteiro podem ser úteis no posterior desenho da interação de um software que seria desenvolvida para realizar ou apoiar a venda de vinhos no supermercado. Por exemplo, o Cenário indica que o cliente deseja “*otimizar a relação custo-benefício*”. Isso significa que fazer uma compra com baixo custo e ao mesmo tempo com qualidade do produto adquirido representa valor para o cliente. Se o cliente está preocupado com a economia em sua compra, isso deve ser respeitado na solução de interface a ser desenvolvida no software, caso contrário o software poderá ser rejeitado ou não interessar ao usuário. Como veremos adiante, um software tende a ter sucesso quando ajuda os usuários a atingir seus interesses e objetivos e fracassam quando tornam os interesses dos usuários mais difíceis ou impossíveis de serem atingidos. Portanto, essa informação sugere que o software a ser desenvolvido deve prover mecanismos, como busca de produtos ordenados por preço, por exemplo, que permitam ao usuário (cliente do supermercado) fazer suas compras com economia.

O ultimo parágrafo ilustra a ocorrência de um evento que pode influenciar a execução da tarefa:

De repente, outro cliente pede ajuda ao vendedor. James pede desculpas ao novo cliente, mas solicita que aguarde um momento, pois está atendendo outro cliente. James não quer ser indelicado com seu cliente José.

Observe que se o software a ser desenvolvido for um sítio Web para venda do supermercado, um supermercado virtual, felizmente, isso não seria um motivo de preocupação, já que provavelmente não haveria como um usuário ser interrompido em sua compra por outro cliente. No entanto, se imaginarmos a solução informatizada na forma de um quiosque eletrônico onde o usuário realiza sua compra, a figura do cliente mal-educado seria uma preocupação já que um cliente mal-educado poderia querer interromper uma cliente que estivesse usando o terminal no quiosque!

Para ficarem mais simples e atrativos, Roteiros descrevem uma situação específica, uma instância, dentre outras, possíveis na realização de uma tarefa. Até, possivelmente, aí está a razão do uso do termo Roteiro pelos idealizadores da técnica. No entanto, roteiros podem e devem ser complementados com uma análise de possíveis variações no modo com a tarefa descrita pode ser realizada.

Possíveis variações relacionadas aos roteiros devem ser identificadas e analisadas seus prós e contras. Por exemplo, no roteiro mostrado acima, poderíamos considerar a situação em que seriam utilizados vídeos sobre vinhos para apoiar as vendas. Isso teria a favor o fato de facilitar a divulgação dos principais produtos (vinhos) a serem oferecidos aos clientes e atrair consumidores. No entanto, teria o lado desfavorável de distrair o cliente. Prós e contras devem ser considerados.

4.4.4 DETALHAMENTO DAS ATIVIDADES DO SUBPROCESSO DE ANÁLISE DE TAREFAS

Esta seção apresenta um detalhamento das atividades de modelagem dos usuários introduzida na seção 4.4. É importante notar que diferentes formas ou técnicas de representação de tarefas podem ser usadas no processo apresentado e que poderia ser necessária uma adaptação da técnica utilizada para que seja mapeada nas atividades de modelagem em duas etapas (preliminar e detalhada) como sugerimos nas seções 4.3.4.1 e 4.3.4.2 abaixo. A técnica de Roteiros foi aqui sugerida com destaque para a modelagem de tarefas, mas outras formas de descrição podem também ser usadas e uma ou outra forma pode ser mais indicada dependendo do tipo de tarefa que queremos descrever.

Claro que, também, é possível considerar a atividade de modelagem de tarefas em nosso processo como constituída de uma única atividade que seria realizada de acordo com a técnica utilizada. No entanto, a proposta de se fazer a Análise de tarefas em duas etapas (preliminar e detalhada) normalmente será uma prática saudável já que a atividade de modelagem é por natureza iterativa.

Cabe, ainda, considerar que na documentação da Análise de tarefas podemos usar vários tipos de notação. Podemos usar desde uma notação informal que pode ser simplesmente no forma de texto, podemos usar notações informais com diagramas ou na forma de algoritmos informais, ou até podemos utilizar diagramas UML. Se desejado, se julgado apropriado, podemos usar diagramas UML de atividades, de estado ou mesmo diagramas de interação para a descrição mais formal de uma tarefa.

4.4.4.1 Modelagem preliminar de tarefas

Esta atividade consiste na criação preliminar de um modelo de tarefas onde se define quais são as principais tarefas a serem modeladas. Essas tarefas são aquelas realizadas pelos usuários no ambiente onde realizam suas atividades. Normalmente, as principais tarefas são as relacionadas com os principais usuários do produto em consideração. As principais atividades realizadas são:

- identificação de tarefas a serem analisadas;
- análise preliminar do modo como as tarefas são executadas e suas características principais tendo em perspectiva o software em consideração;
- priorização das tarefas, identificando aquelas que devem ser analisadas em mais detalhes e simplificação do modelo de tarefas.

A Tabela 4-7 apresenta um detalhamento das atividades de modelagem preliminar de tarefas.

Descrição	Modelagem preliminar
Insumos	1 Especificação de requisitos de Software (ERSw) 2 Documentação do planejamento da análise 3 Lista preliminar de usuários potenciais e respectivas categorias. 4 Informação já levantada no trabalho de campo. 5 Manuais ou outras formas de documentação de procedimentos ou processos usados na organização contratante (se existirem).
Tarefas	1 Realização do trabalho de campo de observação e consulta às pessoas envolvidas visando à modelagem de tarefas. 2 Identificação das tarefas a serem analisadas a partir de reuniões com pessoas envolvidas ou de consulta à documentação existente. 3 Levantamento das características principais das tarefas, identificando: <ul style="list-style-type: none">◦ Aspectos como cenário, objetivo, atores envolvidos, planos, modelos mentais e avaliações que as pessoas envolvidas utilizam na realização das tarefas.◦ Freqüência de realização.◦ Fluxo de comunicação na interação em que os usuários realizam as tarefas.◦ Aspectos ligados à localidade geográfica, cultura e nacionalidade dos atores envolvidos se forem relevantes. 4 Verificação das diferenças relevantes entre as características de tarefas já levantadas (preliminar). Verifica se as características similares indicam diferenças significativas que não tinham sido observadas, considerando-se várias perspectivas, por exemplo, a distribuição esperada de freqüência de realização das tarefas.

	5 Ordenação, simplificação ou generalização das tarefas com base no relacionamento entre elas (preliminar).
Resultados	3 Modelo de Tarefas preliminar

Tabela 4-7 Atividades da modelagem preliminar de tarefas

4.4.4.2 Refinamento da modelagem de tarefas

Esta atividade visa o refinamento e melhoria do modelo de tarefas. Faz-se o detalhamento das tarefas consideradas mais relevantes. Essas tarefas são candidatas a serem automatizadas no sistema a ser desenvolvido, portanto, são essas tarefas ou parte delas que darão origem aos casos de uso do sistema em perspectiva. Várias formas de descrição podem ser utilizadas para cada tarefa, dependendo do enfoque que se quer dar a descrição. Por exemplo, quando se quer enfatizar a interação entre os usuários na realização de uma tarefa, pode-se usar uma descrição da tarefa na forma de “fluxo de trabalho”. Às vezes, pode também ser interessante descrever as tarefas realizadas por um ator, isto é, dar ênfase à descrição do conjunto de atividades realizadas por um ator considerado importante.

As descrições das tarefas são registradas diretamente no documento ERUSw. As tarefas consideradas mais críticas, muitas vezes aquelas realizadas pelos atores focais, devem receber mais atenção no trabalho de desenvolvimento da interação. Neste trabalho de modelagem detalhada, o modelo de tarefas pode ser reorganizado com a criação ou extinção de tarefas. A modelagem de atores também pode ser revista. A organização e simplificação mais apurada do modelo de tarefas são realizadas com base em um trabalho de consulta e validação dos dados levantados.

A Tabela 4-8 apresenta as atividades de refinamento da modelagem de tarefas.

Descrição	Refinamento da modelagem de tarefas
Insumos	1 Especificação de requisitos de Software (ERSw) 2 Documentação do planejamento da análise 3 Modelagem preliminar de atores humanos. 4 Informação levantada no trabalho de campo. 4 Manuais ou outras formas de documentação de procedimentos ou processos usados na organização contratante (se existirem).
Tarefas	1 Realização do trabalho de campo de observação e consulta às pessoas envolvidas visando à modelagem de tarefas. 2 Detalhamento das características principais das tarefas, identificando:

	<ul style="list-style-type: none"> ◦ Aspectos como cenário, objetivo, atores envolvidos, planos, modelos mentais e avaliações que as pessoas envolvidas utilizam na realização das tarefas. ◦ Freqüência de realização. ◦ Fluxo de comunicação na interação em que os usuários realizam as tarefas. ◦ Aspectos ligados à localidade geográfica, cultura e nacionalidade dos atores envolvidos se forem relevantes. <p>3 Melhoria e conclusão da modelagem de tarefas validando as descrições e considerando os relacionamentos entre tarefas. Este trabalho pode levar ao desdobramento de tarefas considerados complexas em tarefas mais simples ou à fusão ou generalização de tarefas com base no relacionamento entre elas e suas características.</p> <p>4 Verificação das diferenças relevantes entre as características levantadas das tarefas. Verifica se as características similares indicam diferenças significativas que não tinham sido observadas, considerando-se várias perspectivas, por exemplo, a distribuição esperada de freqüência de realização.</p>
Resultados	1 Modelo de Tarefas completo e documentado no ERUsw.

Tabela 4-8 Refinamento da modelagem de tarefas

4.4.5 ANÁLISE DE NECESSIDADES

A Análise de tarefas deve começar pela Análise de Necessidades já que necessidades constituem motivação para a realização de tarefas. A Análise de Necessidades visa estabelecer o que se deseja do sistema baseado na organização cliente, necessidades de mercado, objetivo dos usuários ou de qualquer outro interessado no sistema.

O objetivo da análise de necessidades é resumido a seguir.

- Caracterizar em um ou mais níveis de abstração, as necessidades ou interesses dos envolvidos que se espera serem supridas, ainda que parcialmente, pelo produto em desenvolvimento.
- Caracterizar e priorizar os benefícios que seriam conseguidos por meio da satisfação das necessidades através do produto.
- Fazer a correlação entre benefícios e necessidades, de modo que, no passo seguinte, se possa também fazer uma correlação entre benefícios e tarefas para se priorizar as tarefas a serem realizadas pelo sistema.

Cabe observar que as necessidades serão supridas através de tarefas que podem ser realizadas de maneira automática pelo sistema software/hardware ou de maneira manual (ou sem a utilização do sistema) pelos usuários.

Na Análise de Necessidades é importante considerar também os interesses de todos os envolvidos em relação ao sistema. Produtos têm sucesso quando ajudam os usuários a atingir seus interesses e objetivos e fracassam quando tornam os interesses dos usuários mais difíceis ou impossíveis de serem atingidos.

Segue um exemplo que ilustra a questão de interesses conflitantes e a importância desses interesses serem considerados na Análise de Necessidades. Consideremos um sistema, a ser desenvolvido, de apontamento de horas de trabalho de empregados relacionadas com a realização de tarefas. Quais os interesses da empresa e do empregado relacionados a este sistema? Há interesses explícitos, mas interesses não declarados também podem ser tão ou mais importantes que os interesses explicitados pelas pessoas.

Os interesses de uma empresa contratante do desenvolvimento do software poderiam, talvez, ser resumidos como se segue.

- Evitar problemas com auditores
- Aumentar lucros
- Obter o máximo dos empregados
- Obter melhores dados para futuros planejamentos
- Melhorar a organização da empresa

Já os interesses dos empregados da empresa, usuários do sistema a ser desenvolvido, poderiam ser os seguintes.

- Manter seu emprego
- Fazer seu trabalho para ir embora para casa na hora certa
- Fazer o patrão feliz para ser promovido
- Não se esquecer de anotar horas trabalhadas para evitar desconto de pagamento.

Se o software frustrar os interesses dos usuários eles podem não usar o produto como se esperava. No exemplo, se for difícil para o usuário anotar uma tarefa não usual, ele pode simplesmente lançar qualquer código que o sistema aceite. Os usuários também não ficarão nem um pouco satisfeitos se se esquecerem de fazer um apontamento de horas trabalhadas em um dia e não puder fazer as devidas correções no dia seguinte, especialmente se isso significar desconto em seu pagamento! Observe que muitos desses aspectos não seriam revelados explicitamente pelos envolvidos. No entanto, essas questões são relevantes e

podem ser, às vezes, identificadas nas observações, entrevistas e outras formas de contato com as pessoas envolvidas.

Podemos concluir que produtos de sucesso são desenhados para atingir os interesses e objetivos de todos os envolvidos: usuários, empresa, investidores, pais, filhos, etc dependendo da situação. Interesses representam valores para os usuários e, portanto, devem ser respeitados e considerados.

A Análise de Necessidades deve começar por um levantamento de necessidades, considerando-se os interesses e objetivos, dos diversos envolvidos com o sistema em perspectiva. Definem-se também os principais conceitos envolvidos na descrição das necessidades. É importante também levantar os benéficos, para os envolvidos, que o software poderá acarretar na medida em que este contribua para que as necessidades levantadas sejam supridas. O resultado da Análise de Necessidades é uma visão externa do que o usuário será capaz de fazer e ganhará com o sistema.

O modelo resultante da Análise de Necessidades pode ser descrito primeiramente na forma de um texto analisando os aspectos mais importantes, como os interesses e objetivos das partes envolvidas. Em seguida, podem ser usadas uma tabela de Necessidades e uma tabela de Benefícios para substanciar a Análise de Necessidades. Na redação de interesses e objetivos é preciso ter cuidado já que estes podem constituir assuntos sensíveis para as pessoas envolvidas. É necessária uma atenção à forma de se colocar as coisas, possivelmente até omitindo aspectos que possam ser comprometedores para as partes envolvidas.

Na tabela de Necessidades, as necessidades podem ser desdobradas em “subnecessidades” em colunas separadas, criando-se uma hierarquia de necessidades e “subnecessidades”. Isso geralmente é feito em dois ou três níveis. De forma semelhante, a tabela de Benefícios também deve apresentar uma hierarquia de benefícios e “sub-benefícios” desdobrados.

É interessante fazermos também a priorização de benefícios e necessidades e uma correlação entre necessidades e benefícios. A priorização de benefícios costuma ser mais fácil do que a priorização de necessidades. Por isso, devemos começar por pela priorização de benefícios, que deve ser feita com a participação de clientes, usuários e demais envolvidos. A correlação entre necessidades e benefícios visa determinar quais necessidades, uma vez atendidas no software em consideração, contribuem para quais benefícios. Esta correlação ajuda na priorização das necessidades, já que, normalmente, as necessidades mais importantes são aquelas que mais fortemente contribuem para a obtenção dos

benefícios. Sendo assim, é recomendável realizarmos, na sequência, a priorização dos benefícios, a correlação entre necessidades e benefícios, e a priorização das necessidades a serem contempladas no produto em perspectiva.

4.5 ANÁLISE DE AMBIENTE

As pessoas não trabalham ou exercem suas atividades isoladas do meio ambiente. O ambiente de realização das atividades pode ter muita influência na utilização do software e a incompatibilidade do ambiente com o software pode até resultar na rejeição do produto pelo usuário. A Análise de Ambiente visa uma caracterização do ambiente onde os usuários, ou potenciais usuários, realizam suas atividades relacionadas com o produto em consideração.

A análise do ambiente de realização das atividades compreende três aspectos, apresentados a seguir.

- Ambiente físico
- Ambiente social
- Ambiente cultural

O ambiente físico compreende os espaços onde os usuários realizam suas atividades. A análise de ambiente físico aborda aspectos como os listados a seguir. Esses aspectos são importantes e nos interessam na medida em que têm impacto na utilização do software em consideração.

- Meio ambiente
 - Por exemplo, poeira, óleo, ou qualquer elemento nocivo que dificulte o uso do equipamento.
 - A temperatura, umidade, ou outro fator relacionado com o clima pode afetar o trabalho do usuário.
 - O nível de ruídos pode perturbar a concentração do usuário
 - A iluminação pode colocar obstáculos à utilização de certos dispositivos de interface.
- Espaço de trabalho
 - Por exemplo, pode ser difícil utilizar-se o mouse no atendimento ao público. Pode haver dificuldade das pessoas em operá-lo ou mesmo deve ser considerada a possibilidade de ocorrência de furtos em ambientes públicos.

- Disponibilidade de equipamento
 - Pode ser relevante considerar, por exemplo, se o usuário tem seu próprio equipamento ou pode necessitar tomá-lo emprestado.
- Disponibilidade de energia elétrica
 - A energia elétrica é disponível de maneira adequada aos usuários?

Mas não é só o ambiente físico que precisa ser considerado. Muitas vezes, dependendo do tipo de produto a ser desenvolvido, o ambiente social e cultural precisa também ser considerado. O ambiente social está relacionado, por exemplo, a pressões por produtividade, por precisão ou qualquer outro fator que possa influenciar na utilização do software a ser desenvolvido. O modo como as pessoas interagem entre si, ou a separação geográfica das pessoas também são exemplos de fatores sociais que podem ser relevantes para a utilização do produto. Por exemplo, os seguintes fatores devem ser considerados.

- Usuário trabalha sob pressão por produção, rapidez, precisão ou qualquer fator que possa afetar a utilização do produto?
- Recursos disponíveis para ajudar o usuário. Existem manuais ou pessoas por perto a quem possam recorrer?
- As pessoas com as quais interagem ficam em local próximo? A separação geográfica afeta o trabalho?
- Como os usuários se comunicam? Os usuários utilizam Fax, e-mail, telefone, etc em sua comunicação? Esses hábitos são importantes, por exemplo, na definição de um mecanismo de comunicação a ser utilizado no software a ser desenvolvido.
 - Como o ambiente físico interfere no ambiente social? As pessoas trabalham em casa? As pessoas trabalham em cubículos ou em ambientes compartilhados? Um espaço de trabalho com alta concentração de pessoas pode dificultar a comunicação entre elas. Por outro lado, a presença de companheiros de trabalho pode favorecer as atividades das pessoas envolvidas.
- Como é a relação entre os usuários e seus clientes? Como se comunicam? Há tensão na relação? Há necessidade de respostas em tempos estritos?
- O desempenho do usuário é ou precisaria ser monitorado? Quais as consequências isso em relação aos interesses das pessoas envolvidas?

O ambiente cultural está relacionado a aspectos de cultura regional ou de grupos sócio-econômicos que também podem ser relevantes. Seguem alguns exemplos de fatores a serem considerados.

- A cultura do país ou região influencia no trabalho do usuário?
- Os usuários trabalham em locais com diferenças culturais significativas?
- O grupo socioeconômico a que pertencem precisa ser considerado?
- Os usuários pertencem a uma cultura profissional que determine valores, estilos de trabalho ou comportamentos que necessitem ser considerados?
- Isso tem implicações quanto ao estilo de uso, ajuda e documentação, por exemplo?
- E quanto às expectativas de tempo de resposta do sistema? Este aspecto pode ser muito importante porque determina a expectativa (ou tolerância) dos usuários em relação aos tempos de respostas dos sistemas com os quais irá interagir.

Como já mencionamos, o ambiente de realização das atividades pode ser considerado como parte da Análise de tarefas assim como parte da Análise de usuários. A decisão sobre a melhor forma de se modelar o ambiente deve ser baseada no fato do ambiente estar mais associado às pessoas ou às atividades que elas realizam. Por exemplo, o risco de um ambiente exposto à radioatividade pode estar associado à tarefa de se tirar radiografias, para qualquer que seja o usuário envolvido nesta tarefa. Já o aspecto da pressão social por não se cometer erro pode ser modelado como associado ao usuário médico, independente das tarefas que ele realiza. No documento “erusw” a análise de Ambiente pode ser documentada junto com a Modelagem dos usuários ou com a Modelagem de Tarefas.

4.6 ANÁLISE DE PRODUTOS CONCORRENTES OU SIMILARES

A Análise de Produtos Concorrentes ou Similares visa o estudo de produtos semelhantes ao produto em consideração com o objetivo de:

- coleta de informações para que se possa melhorar o produto em consideração a partir do conhecimento das fraquezas e pontos fortes de produtos concorrentes ou similares.
- familiarização com o domínio do problema;

- estudo de características de produtos similares tendo em perspectiva o software a ser desenvolvido ou analisado (produto em consideração);
- identificação de oportunidades que possam diferenciar o produto;
- utilização do produto similar para se promover avaliações de aspectos específicos quando essas características não estão ainda disponíveis em um produto em desenvolvimento ou mesmo quando se considera estender o produto em consideração com essas características;
- obter-se a visão de um produto semelhante já implementado - pode dar uma visão mais realista do que a permitida por protótipos.

Não se trata, obviamente, de cópia ou violação de direitos de propriedade, mas de se conhecer os possíveis concorrentes ou produtos similares ao que está sendo desenvolvido com o objetivo de identificar oportunidades de diferenciais, identificar limitações ou mesmo utilizar como uma referência para comparações.

A Análise de Produtos Concorrentes ou Similares pode ser realizada por meio de avaliações de usabilidade de maneira mais formalizada ou de forma simplificada, como também pode ser complementada com análises disponíveis em revistas ou outros meios que podem dar subsídios valiosos para o desenvolvimento de um sistema. Quando possível, a análise de vários produtos concorrentes permite uma avaliação comparativa bem interessante.

A análise de produtos semelhantes, funcionando com se fosse um protótipo, também pode viabilizar a comparação de diversas abordagens para questões de interação que interessam aos desenvolvedores. Pode ser proveitosa, também, a análise de produtos concorrentes com outros tipos de interface. Por exemplo, na análise de um livro eletrônico pode ser interessante analisar-se como os usuários utilizam uma enciclopédia em papel.

O Praxis-u prescreve que a Análise de Produtos Concorrentes ou similares seja registrada no documento “erusw”.

4.7 GLOSSÁRIO

JAD

Joint Application Development — JAD ou Joint Application Design é uma metodologia criada pela IBM do Canadá em 1977 e adaptada para o Brasil em 1982 para moderação de discussões de brainstorming acelerando e consolidando o desenvolvimento de aplicações de

Sistemas de Informação. JAD é uma metodologia que acelera o projeto de sistemas. Guiados por um líder de reunião, usuários e analistas projetam o sistema juntos, em sessões de grupo estruturadas. JAD utiliza a criatividade e o trabalho em equipe de dinâmica de grupo para definir o ponto de vista dos usuários sobre o sistema, desde os objetivos e aplicações do sistema até a geração de telas e projetos de relatórios. A aplicação JAD permite a criação, em menos tempo, de sistemas mais eficazes (Wikipedia, n.d.).

4.8 BIBLIOGRAFIA

Chapman, C.N. & Milham, R. *The personas' new clothes*. Human Factors and Ergonomics Society (HFES) 2006, San Francisco, CA. October 2006.

Cheng, L., Scapin, C. A. at al. *QFD: Planejamento da Qualidade. Escola de Engenharia da UFMG*. Fundação Cristiano Ottoni, Belo Horizonte, MG, 1995.

Constantine, L.L. & Lockwood, L.A.D. *Software for Use: a Practical Guide to the Models and Methods of Usage Centered Design*, Addison-Wesley, Reading-MA, 1999

Cooper, A. *The Inmates are Running the Asylum: Why High Tech Products Drive Us Crazy and How To Restore the Sanity*. Macmillan Publishing Co., 2000.

Cooper, M. *Evaluating accessibility and usability of web pages*. In Proceedings of 2nd Int. Conference on Computer-Aided Design of User Interfaces, Kluwer Academic, 33-42, 1999.

Cooper, A. e Reimann, R. *About face 2.0: The essentials of interaction design*. Information Visualization. 2004

Grudin, J. and Pruitt, J. *Personas, participatory design and product development: an infrastructure for engagement*. Paper presented at Participatory Design Conference 2002, Malmo, Sweden. June 2002.

Hackos, JT & Redish, JC, *User and Task Analysis for Interface Design*, John Wiley & Sons 1998

Hix, D.; Hartson, H. R. *Developing User Interfaces: ensuring usability through product & process*, John Wiley and Sons, 1993.

Mulder, S. and Yaar, Z. *The user is Always Right: A practical Guide to Creating and Using Personas for the Web*. New Riders Publishing Thousand Oaks, CA, USA, 2006.

Rosson, M. B., Carroll, J.M. *Usability Engineering: Scenario Development of Human-computer Interaction*. Morgan kaufmann Publishers, 2002.

Wikipedia, em <http://pt.wikipedia.org/wiki/> acessado em 10/2009

5 ESPECIFICAÇÃO DE REQUISITOS DE USABILIDADE

5.1 VISÃO GERAL

A especificação de requisitos é usada na Engenharia de Software para definir as condições que um produto de software deve satisfazer para ser aceito e considerado com um nível aceitável de qualidade. De forma semelhante, a Especificação de Requisitos de Usabilidade é uma atividade de processo que tem como objetivo a definição de metas verificáveis, geralmente quantitativas, de desempenho que são usadas como referência para avaliarmos a qualidade da interface do usuário.

A *ISO 13407 Human-centred design processes for interactive systems* é uma norma internacional que define processos de usabilidade. Um diagrama de processos, em alto nível de abstração, prescritos pela norma ISO 13407 é mostrado na Figura 5-1. Podemos observar que, em linhas gerais, o processo de usabilidade previsto pela norma ISO 13407 é constituído por uma atividade de planejamento e um ciclo onde são realizadas as atividades de:

- Análise de contexto de uso (que a Norma define como Especifica o contexto de uso);
- especificação de requisitos de usabilidade;
- desenho da interação;
- avaliação do desenho.

Observe que a norma ISO 13407 sugere um ciclo onde, em uma visão bem simplificada, o contexto é analisado, desenhos (podem ser protótipos) são produzidos e a interface é avaliada até que seja considerada satisfatória. A especificação de requisitos de usabilidade, então, serve como uma referência para avaliarmos se a interface apresenta um nível satisfatório de qualidade em termos de usabilidade.

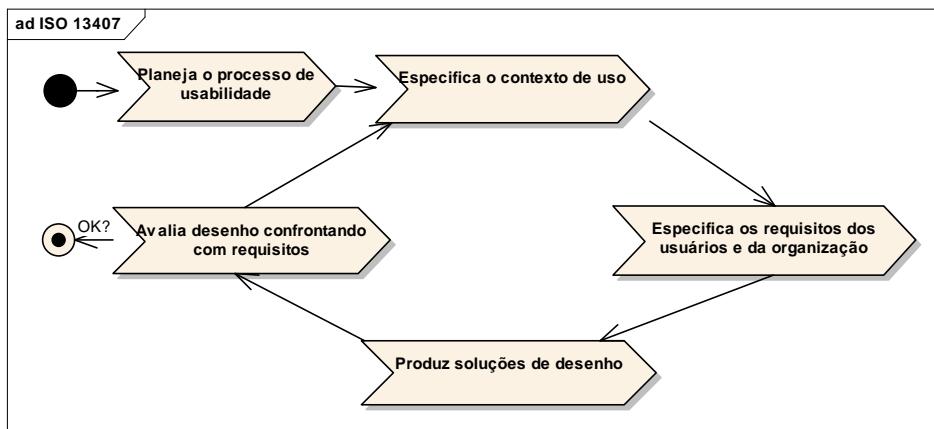


Figura 5-1 Processos de usabilidade definido pela norma ISO 13407

As metas de desempenho estabelecidas na Especificação de Requisitos de Usabilidade são níveis de desempenho que desejamos que os usuários pudessem atingir ao interagir com o sistema. A Especificação de requisitos de usabilidade, que para simplificar usaremos também a expressão “Especificação de Usabilidade” poderá, assim, ser usada como uma indicação de quando o projeto de desenvolvimento está convergindo em direção a uma interface com sucesso. Com o estabelecimento da Especificação de Usabilidade o mais cedo possível no processo de desenvolvimento, e monitorando-as a cada iteração, pode-se determinar quando a interface está, de fato, indo em direção a melhorias.

A realização de medidas é essencial para termos controle de um processo e em Usabilidade isso não é diferente. É a realização de medidas que nos permite deixar especulações, o chamado “achismo”, para termos processos mais maduros e profissionais. A importância da realização de medidas pode ser apreciada na citação dos autores Good, M. e outros (Good, M. et al, 1986): “Engenharia de Usabilidade é um processo através do qual as características de usabilidade são especificadas, antecipadamente e de forma quantitativa no processo de desenvolvimento, e medidas durante todo o processo”. Sem especificações mensuráveis, é impossível determinar metas de usabilidade e dizer se o produto final alcança essas metas. No fundo, se não conseguimos realizar medidas em uma atividade, provavelmente não conseguiremos gerenciá-la adequadamente (claro, falando de atividades complexas como o desenvolvimento de software).

5.2 TABELA DE ESPECIFICAÇÃO DE REQUISITOS DE USABILIDADE

O conceito de especificação formal de requisito em formato de tabela foi desenvolvido por Gilb (Gilb, 1981). Gilb propõe a utilização de uma tabela, que chamaremos de ERU, para a especificação de requisitos de usabilidade.

A Tabela ERU é uma das principais fontes para o planejamento das avaliações com os usuários. Deve ser utilizada tanto para avaliações formativas, aquelas realizadas durante o desenvolvimento do software, quanto somativas, ou seja, avaliações de produtos já existentes.

Or- dem	Identifi- cador	Req uisit o ou met a?	Atributo de usabili- da- de	Ator	Instrumento de medida	Valor a ser medido	Níveis de desempenho			
							Atual	Pior aceitá vel	Melhor Possi- vel	Alv o
1	RU01	R	Desempe- nho inicial	Todos	"Acrescente o compromisso..." - benchmark 1	Tempo de execução na primeira tentativa	15s	30 s	20s	10s
2	RU02	R	Desempe- nho inicial	Todos	"Apague o compromisso..." benchmark 2	Número de erros na primeira tentativa	0 erro	3 erros	1 erro	0 erro
3	RU03	R	Satisfação - inicial	Secretaria	Questionário: questões...	Média das avaliações	??	7,0	9,5	8,5
4	RU04	M								
5	RU06	R								

Tabela 5-1 Tabela ERU

A Tabela **Tabela 5-1** mostra um excerto de uma Tabela ERU ilustrando os requisitos de usabilidade para o exemplo apresentado no livro de Hix e Hartson, (Hix, D.; Hartson, 1993). Neste exemplo, os autores propõem o desenvolvimento da interface de uma agenda eletrônica que substituiria as agendas de papel, aliás, muito usadas pela praticidade.

A seguir, apresentamos os elementos que constituem a Tabela ERU, com o significado das colunas, utilizando o exemplo da agenda eletrônica para ilustração. As três primeiras colunas são:

- **ORDEM:** esta coluna é utilizada para numeração das linhas da tabela.
- **IDENTIFICADOR:** a coluna “identificador” é utilizada para associar um identificador a cada item (ou linha) da tabela. Isso pode ser útil para se fazer uma referência a um item, podendo inclusive, em uma implementação mais sofisticada, ser usado como um índice em uma tabela de banco de dados.
- **REQUISITO OU META?** Esta coluna é usada para definir se o elemento apresentado em uma linha específica da Tabela se trata de um requisito ou de uma meta de usabilidade. Os dois conceitos são semelhantes, a diferença é que consideramos como “requisito” se este for estabelecido pelo contratante do projeto e “meta” se o requisito for definido pela equipe desenvolvedora do software em consideração. Em essência, a diferença é que o “requisito” de usabilidade seria considerado uma condição para aceitação do produto enquanto a “meta” seria uma referência utilizada para a equipe desenvolvedora para avaliar a qualidade da interface

Os demais elementos da Tabela ERU serão apresentados nas seções seguintes.

5.2.1 ATRIBUTOS DE USABILIDADE

Atributo de usabilidade é a característica geral de Usabilidade a ser usada como critério para avaliação da interface. Esse elemento é essencial já que corresponde a parâmetros usados para se medir a usabilidade de uma versão da interface.

Como atributo, pode-se usar os cinco principais definidos por Nielsen (Nielsen, 1993) ou outros relevantes. Às vezes é interessante estabelecer condições em que um atributo é avaliado. Por exemplo, podemos definir “Desempenho inicial” quando é importante medir o desempenho de um usuário que nunca usou ou que é iniciante no uso do software em consideração ou “Desempenho em longo prazo” quando for interessante avaliar o desempenho de um usuário que já tenha experiência no uso de um sistema.

Para uma tarefa complexa, pode ser interessante estabelecer diferentes atributos, por exemplo, aprendizado e desempenho em longo prazo.

SUGESTÕES DE ATRIBUTOS

Seguem sugestões de atributos interessantes que podem ser utilizados na Tabela ERU

- **Desempenho inicial:** refere-se ao desempenho do usuário que está iniciando-se e não tem experiência no uso do sistema.
- **Desempenho em longo prazo:** refere-se ao desempenho do usuário em uso mais regular do sistema durante um longo período. É muito utilizado quando a tarefa a ser avaliada (descrita em “Instrumento de medida” na tabela) é uma tarefa realizada no dia a dia no ambiente de trabalho das pessoas.
- **Facilidade de Aprendizagem:** facilidade ou rapidez com que o usuário aprende a lidar com o sistema.
- **Retenção** é a capacidade de o usuário que usou um software e ficou um período de tempo sem utilizá-lo, de reter na memória o que aprendeu para que consiga utilizá-lo novamente no futuro.
- **Uso de características (*features*) avançadas** é utilizado para determinar a usabilidade de funções complexas da interface.
- **Satisfação inicial:** ou primeira impressão, é a avaliação que o usuário que está iniciando-se e não tem experiência no uso do software faz de sua utilização.
- **Satisfação em longo prazo:** é a avaliação do usuário que tem experiência na utilização do sistema.
- **Prevenção de erros:** é a capacidade do sistema de prevenir erros dos usuários em sua utilização.
- **Acurácia** é a capacidade do software de oferecer resultados na precisão desejada pelo usuário.
- **confiabilidade, clareza, etc** são exemplos de outros atributos que podem ser interessantes para determinados produtos de software.

5.2.2 ATOR

Na escolha dos atributos devemos considerar os diversos perfis de atores para os quais desejamos determinar o desempenho como prescrito na especificação de requisitos de usabilidade. Pode ser interessante utilizar-se atributos associados a papéis de usuários específicos, para isso pode ser utilizada a coluna identificada como "Ator" na Tabela ERU. Normalmente, os papéis de usuário focais é que serão de maior interesse para se ter seu desempenho medido (listado na Tabela ERU), mas, claro, qualquer ator pode ser considerado relevante o suficiente para ter o seu desempenho, na utilização do software em consideração, medido.

Em geral não é viável estabelecer metas para todas as classes de usuários ou tarefas possíveis. Especula-se que aproximadamente 80% dos usuários usam somente 20% das funcionalidades de um sistema interativo. Sendo assim, as tarefas mais importantes ou críticas, para os atores mais importantes, é que serão objeto da especificação de requisitos.

5.2.3 INSTRUMENTO DE MEDIDA

O "Instrumento de medida" descreve o mecanismo utilizado para se obter valores de um atributo particular de usabilidade. Deve ser quantitativo, isto é, poder ser medido numericamente. No entanto, a medida pode ser objetiva ou subjetiva. A medida objetiva se refere às medidas quantitativas do desempenho observável do usuário durante a realização de tarefas usando a interface e a medida é subjetiva quando baseada em opiniões de usuários sobre a interface.

Os termos objetivo e subjetivo estão associados ao modo como são obtidos os dados relacionados aos atributos de usabilidade.

Medidas são objetivas quando os dados são coletados por observação do desempenho do usuário em tarefas específicas. Essas tarefas são consideradas de *benchmark*.



Figura 5-2 Benchmark

O significado original do termo “benchmark” é o de uma marca permanente usada como referência para observações topográficas ou de nível de reservatórios ou fluxos de água. A Figura 5-2 ilustra o uso de um benchmark em um fluxo de água. Este termo é usado em várias áreas com o significado de um indicador que dá a referência de desempenho de algum objeto ou atributo que se deseja avaliar. Em computação, por exemplo, “benchmark é o ato de executar um programa de computador, um conjunto de programas ou outras operações, a fim de avaliar a performance relativa de um objeto, normalmente executando uma série de testes padrões e ensaios nele” (Wikipedia, n.d.). Aqui, em Usabilidade, usamos o termo benchmark para denotar uma tarefa usada como referência para avaliação do desempenho do usuário na interação com um sistema.

Medidas subjetivas são quando os dados são obtidos através de questionários de preferências dos usuários. As medidas objetivas e subjetivas são igualmente importantes para a especificação e para a avaliação da usabilidade de um desenho.

TAREFA DE BENCHMARK

Como mencionamos, a tarefa de Benchmark é uma tarefa usada como referência para as medições objetivas.

É importante que uma tarefa de benchmark seja bem redigida, para indicar claramente ao usuário participante de uma avaliação o que se deseja, em que consiste a tarefa, e permitir comparações. As tarefas de benchmark devem ser específicas para que o participante não desvie sua atenção para detalhes irrelevantes durante o teste. Devem ser descritas na linguagem do domínio da aplicação.

Por exemplo, uma tarefa redigida como “Utilize o sitio que você está avaliando para fazer uma reserva de hotel para o período de Natal” não é específica. Vai deixar o participante da avaliação confuso e provavelmente ele (ou ela) vai perguntar: “Mas em quais datas?” ou “O que se entende por período de Natal” ou, ainda, “mas para quantas pessoas seria a reserva?”. Para evitar essas confusões e permitir uma base melhor para fazermos comparações do desempenho entre vários participante, a tarefa de Benchmark deve ser redigida de forma precisa, clara e específica.

Outro ponto importante que deve ser observado ao definirmos tarefas de benchmark: coloque essa tarefa em termos de um objetivo (associado à realização de uma tarefa) a ser atingido. Ao iniciar uma tarefa, uma pessoa tem em mente um objetivo ou uma necessidade que a motiva. Deve-se ter o cuidado de colocar para o usuário participante de um teste de usabilidade, a proposta de realização de uma tarefa, um objetivo, não lhe dizer como realizar a tarefa. Diga o **que** o usuário participante deve fazer, mas não **como** deve fazer. O objetivo da avaliação é justamente avaliar como o usuário realiza a tarefa utilizando a interface do produto em consideração, que, no caso, é o instrumento para sua realização. Por exemplo, redija uma tarefa como “adicone um compromisso de almoço com os diretores João e Gilberto todas as quartas as 14:00h por 3 meses” e não “vá no menu de compromisso, entre na tela de repetição de compromisso e ...”. Neste último caso, você já terá dada a solução de como usar a interface para realizar a tarefa, e a avaliação da interface perderá o sentido. A redação da tarefa deve permitir avaliar se o usuário consegue perceber como realizar a tarefa.

Na escolha de tarefas de benchmark, é bom usar características (features) simples da interface ou grupos pequenos de características, para que a causa dos problemas identificados possam ser mais facilmente rastreadas no desenho. Quando as tarefas são complexas, elas devem ser convertidas em subtarefas para a avaliação, porque as tarefas complexas são demoradas para se executar, dificultam a identificação de problemas e dificultam a comparação do desempenho de diferentes participantes.

Mas, por outro lado, pode ser necessário usar-se uma seqüência de tarefas, como no caso em que elas costumam ocorrer em seqüência. Por exemplo, uma tarefa de pagamento de uma compra em um sítio de comércio eletrônico. Tarefas complexas como essas, chamadas de tarefas representativas, também têm seu interesse, mas geralmente são usadas para a obtenção de dados qualitativos já que fica difícil se obter dados quantitativos confiáveis que permitam comparações.

QUESTIONÁRIOS

Questionários podem ser usados para avaliar a satisfação subjetiva do usuário com a interface. Além de utilização em testes de usabilidade, o questionário é um instrumento que pode ser utilizado para guiar o desenho ou melhorias de desenho da interface, identificar áreas potenciais para introdução de melhorias, validar avaliações comparativas, dentre outros.

Existem métodos científicos para elaboração e validação de questionários. A utilização desses métodos é importante para que se obtenham dados confiáveis. Por exemplo, um questionário deve relacionar várias características de interface de usuário, de uma forma organizada. Montar um questionário não é somente montar uma lista de características, listá-las e colocá-las juntas. O questionário deve endereçar problemas de confiabilidade e validação, criando uma medida confiável para determinados aspectos da interface. Sendo assim, de preferência, profissionais devem ser utilizados para sua produção.

Uma alternativa a desenvolver um questionário seria utilizar um questionário já elaborado, de um fornecedor confiável, e adaptá-lo para uso em uma situação específica. Por exemplo, o [QUIS \(Quis, University of Maryland, 2009\)](#) é um exemplo conhecido de questionário para avaliação de satisfação de usuários que pode ser obtido mediante licenciamento.

O QUIS contém, primeiramente, uma parte demográfica onde se registra o perfil do participante. Depois, apresenta uma parte dedicada à medida de satisfação do usuário de modo geral, seguido de outras partes dedicadas à medida de onze fatores específicos de uma forma organizada hierarquicamente. São eles: aspectos de tela; terminologia e feedback do sistema; aprendizado; características do sistema; documentação técnica; tutoriais on-line; multimídia; reconhecimento de voz; ambiente virtual; acesso à internet e instalação do software.

5.2.4 VALOR A SER MEDIDO

Esta coluna da Tabela ERU indica o tipo dos dados, ou o tipo de valores, que serão coletados durante os testes juntos aos participantes. As medidas mais comuns são o tempo em que se completou uma tarefa e o número ou percentagem de erros, no entanto, vários outros tipos de valores podem ser de interesse.

No caso de medida de erros, é necessário definir exatamente o que significa um erro. Por exemplo, se o usuário não usa um botão ou menu esperado na realização de uma tarefa,

ainda que ele tenha conseguido realizar uma tarefa, deve ser contado como erro. Isso porque o erro de usabilidade indica qualquer aspecto da interface que mereça ser analisado e, se necessário, alterado para melhoria da interação.

Geralmente, consideramos que houve um erro de usabilidade em uma avaliação quando um participante não completa uma tarefa ou quando faz uma ação que não leva a progresso na execução de uma tarefa (o que exclui acesso a Helps e documentação). Por exemplo, se o participante toma um caminho errado, volta e toma um caminho certo na execução de uma tarefa, ainda assim ocorreu o erro.

Para um questionário, é tipicamente utilizada a média de avaliações medidas. O “Valor a ser medido” de um atributo como “Satisfação inicial” na Tabela ERU pode ser obtido como uma média entre vários itens (questões) de um questionário. •

Outro exemplo de “Valor a ser medido” que pode ser interessante é a percepção do usuário do tempo decorrido. Por exemplo, uma instalação demorada, mas na qual o usuário fica ocupado trocando CDs pode ser percebida como rápida.

Algumas outras medidas que podem ser usadas nas tarefas de *benchmark* são apresentadas a seguir.

- Percentagem de tarefas completadas em um tempo determinado.
- Proporção sucesso / fracasso na execução de uma tarefa;
- Tempo gasto em erros e recuperação de erros.
- Número de comandos/ações usados para realizar uma tarefa.
- Frequência do uso do help e documentação.
- Número de repetições ou falhas na tentativa de execução de uma tarefa.
- Número de comandos disponíveis não executados.
- Número de vezes em que o usuário expressou frustração ou satisfação.

5.2.5 NÍVEIS DE DESEMPENHO

Na Tabela ERU, os níveis de desempenho referem-se a metas quantitativas de desempenho esperado dos usuários na interação com o software em consideração.

O tempo atribuído a cada tarefa depende de sua complexidade e uso. Por exemplo, em uma tarefa freqüente, a duração admitida deve ser menor.

Quando o sistema a ser avaliado já se encontra operacional, a definição dos níveis esperados de desempenho dos usuários fica mais fácil. No entanto, mesmo quando se trata de um software em desenvolvimento, vários critérios ou fontes podem ser utilizados para se fazer as estimativas de níveis de desempenho. Alguns desses são listados a seguir.

- Um sistema semelhante existente ou versão anterior de um sistema em desenvolvimento;
- Sistemas concorrentes, principalmente aqueles com uma grande fatia do mercado ou com uma interface de usuário reconhecida pela qualidade.
- A realização de tarefas sem o uso de um sistema de computação (ex. manualmente, usando papel e caneta).
- O uso pelos desenvolvedores de seu próprio protótipo para alguma versão da interface.
- Feedback de mercado, baseado na aspiração dos usuários com sistemas similares
- Estimativa baseada na experiência dos envolvidos, popularmente chamado de “saque”, quando há pouco com o que se comparar!

Diferentes papéis de usuários podem significar necessidade de avaliação de diferentes tarefas e diferentes níveis de desempenho nas tarefas. Pode-se inclusive usar diferentes tabelas de Especificação de Requisitos de Usabilidade. Com a prática, desenvolvedores tornam-se bastante habilidosos em estabelecer especificações de usabilidade confiáveis e estabelecer níveis razoáveis de valores para os atributos.

O nível **Atual** é o nível corrente do valor a ser medido para o atributo de usabilidade na presente versão do sistema. Ou seja, é o nível atual de desempenho observado do usuário, sem a utilização do sistema em consideração. A medição do nível Atual ajuda a assegurar que os outros níveis possam ser estimados. É útil também saber como está o nível atual de desempenho em relação a um ou mais sistemas concorrentes ou similares.

No exemplo mostrado na Tabela 5.1 apresentada anteriormente e copiada abaixo, foi atribuído o valor do nível Atual de desempenho para “Apague o compromisso...” como 0 (zero) já que em uma agenda em papel não é esperado que ocorra um erro para a realização desta tarefa. O valor do nível Atual para “Satisfação inicial” foi considerado não aplicável (denotado “??”) já que não interessa avaliar-se esse atributo para agenda em

papel: não se espera um usuário da agenda eletrônica que não conheça uma agenda em papel

Or- dem	Identifi- cador	Req uisit o ou met a?	Atributo de usabili- dade	Ato r	Instrumento de medida	Valor a ser medido	Níveis de desempenho			
							Atual	Pior aceitá vel	Melhor possí- vel	Alvo
1	RU01	R	Desempe- nho inicial	Todos	"Acrescente o compromisso..." - benchmark 1	Tempo de execução na primeira tentativa	5 s	12 s	4 s	7 s
2	RU02	R	Desempe- nho inicial	Todos	"Apague o compromisso..." benchmark 2	Número de erros na primeira tentativa	0 erro	3 erros	0 erro	1 erro
3	RU03	R	Satisfação - inicial	Secretari a	Questionário: questões...	Média das avaliações	??	7,0	9,5	8,5
4	RU04	M								
5	RU06	R								

Tabela 5-2 Tabela ERU

O nível **Pior aceitável** indica o pior nível de desempenho do usuário que seria ainda aceitável para cada atributo de usabilidade; não o pior que pode acontecer. É o nível mínimo de desempenho que os usuários podem alcançar e ainda considerar-se que a interface possui algum crédito em termos de usabilidade. Espera-se que, para todos os atributos avaliados, o "pior nível aceitável", no mínimo, deve ser alcançado.

O nível **Melhor possível** indica o limite superior realístico do estado de arte, o "nível de inspiração" de um atributo de usabilidade. Mostra o potencial de um atributo e serve como referência para futuras versões do sistema. Deve que ser viável, ainda que represente um desafio, atingi-lo.

O nível **Alvo** indica um valor que significa sucesso inquestionável de usabilidade para a interface, isto é, o nível "que você deseja". A expectativa é que o nível Alvo deva ser alcançado para todos os atributos especificados.

Finalmente cabe observar que quando forem realizados os testes de usabilidade com usuários baseado na Tabela ERU, não serão obtidos os valores reais de desempenho observado dos usuários. Os valores obtidos permitem uma comparação rápida entre os níveis especificados e o resultado real dos testes com os usuários. A partir dessa comparação, pode ser feita uma revisão dos valores dos níveis estimados na Tabela ERU.

5.2.6 DIRETRIZES PARA DEFINIÇÃO DA TABELA ERU

Algumas diretrizes relacionadas à definição da Tabela ERU podem ser apontadas.

No dimensionamento da Tabela ERU, é necessário considerar os recursos/prazos disponíveis. O número de atributos a ser medido deve ser razoável na prática. Quando o desenvolvedor não tem muita experiência, não deve ser muito ambicioso em relação ao número de atributos a serem avaliados. Importante, também, todos os membros do projeto devem concordar com os atributos e valores da Tabela ERU; isso é importante para o comprometimento da equipe.

É preciso considerar que cada atributo de usabilidade deve ser mensurável na prática. Por exemplo, é razoável fazer-se uma medição do desempenho do usuário por um longo tempo?

Os papéis-de-usuários aplicáveis devem ser especificados de forma clara. Pode-se usar uma coluna de Atores na Tabela ERU, como mostramos, ou mesmo fazer-se tabelas separadas para cada papel relevante de usuários.

Verificar se os atributos utilizados refletem as prioridades de usabilidade é importante para evitar desperdício de recursos nas avaliações. A escolha de uma tarefa não representativa pode representar investimento em uma função que não será muito utilizada - perda de dinheiro e tempo!

Outro ponto importante é verificar se as metas para os vários níveis são razoáveis. Cabe observar que é comum que o desenvolvedor iniciante seja muito leniente, defina níveis de desempenho que ficam aquém do necessário, o que não é producente.

Quando os resultados do desempenho observado dos usuários participantes na realização dos testes de usabilidade são muito piores do que os estimados na Tabela ERU há duas possibilidades:

- o processo está caminhando normalmente mas há sérios problemas de usabilidade na interface que devem ser resolvidos, **ou**

- os valores estimados na Tabela ERU são irrealísticos. É necessário rever os níveis de desempenho estimados apresentados na Tabela ERU.

Além de diretrizes gerais, algumas diretrizes para se determinar o nível Pior aceitável podem ser arroladas, como se segue.

- Deve, quando possível, estar próximo do valor do nível Atual de desempenho dos usuários.
- Deve ser mais alto, ou seja, indicar melhor desempenho, do que o nível Atual na medida em que o nível Atual não seja considerado satisfatório.
- Como o sistema atual pode ser muito diferente do sistema planejado, como no caso de nosso exemplo, onde temos agenda em papel versus agenda eletrônica, pode acontecer até que nível Atual seja considerado inatingível para ser o nível Alvo. Nesse caso, deve-se fazer uma estimativa ponderada para os outros níveis de desempenho. Por exemplo, tarefas simples como “acrescentar compromisso”, linha 1 da tabela [Tabela 5-2](#), podem ser feitas muito rapidamente em papel. Sendo assim, o desempenho Alvo e o Pior aceitável foram estimados como aquém do nível Atual (pior do que o nível Atual).

O nível Melhor possível é o melhor nível de desempenho que você pode esperar em uma situação ideal, o que nos leva à diretriz que, para estimá-lo, deve-se considerar onde é possível se chegar com os melhores usuários (mais bem treinados), nas melhores condições, com o melhor desenho e com o melhor uso da tecnologia disponível.

Na estimativa do nível Alvo podemos usar as seguintes diretrizes.

- O nível Alvo deve, normalmente, ser mais alto que o nível Atual para o sistema/versão existente, para representar melhoria. É necessário comparar com sistemas concorrentes.
- O nível Alvo deve ser tal que se for alcançado durante os testes com o usuário, podemos estar confiantes de boa qualidade em termos de usabilidade.

5.3 GLOSSÁRIO

BENCHMARK

Em usabilidade, usamos o termo benchmark para denotar uma tarefa usada como referência para avaliação do desempenho do usuário na interação com um sistema.

5.4 BIBLIOGRAFIA

- Gilb, T. *Design by Objectives*, Manuscritos não publicados, 1981.
- Good, M. et al. *User Derived Impact Analysis as a Tool for Usability Engineering*, Proceedings of CHI Conference on Human Factors in Computing Systems, New York: ACM, 241-246, 1986.
- Hix, D.; Hartson, H. R. *Developing User Interfaces: ensuring usability through product & process*, John Wiley and Sons, 1993.
- <http://lap.umd.edu/QUIS> (site QUIS).
- ISO/DIS 13407 *Human-centred design processes for interactive systems*, 1999.
- [QUIS \(University of Maryland\)](#): acessado em <http://lap.umd.edu/quis/> em 20/10/2009
- Nielsen, J. *Usability Engineering*. Chestnut Hill, MA, Academic Press, 1993.
- Wikipedia, em <http://pt.wikipedia.org/wiki>, acessado em 10/2009

6 DESENHO DA INTERAÇÃO

O desenho da interface do usuário envolve aspectos relacionados à interação usuário-computador e aspectos construcionais, relacionados à Engenharia de software. Este capítulo trata de aspectos relacionados ao desenho da interação dentro do escopo da usabilidade; não abordamos aspectos construcionais.

Conforme definido no Praxis-u, a atividade de desenho da interface compreende duas atividades: Definição do estilo de interação e Desenho da interação. As duas atividades podem ser consideradas atividades de desenho da interface do usuário, no entanto, preferimos separá-las para enfatizar para enfatizar os aspectos relacionados a padrões, que estão incluídos na atividade de Definição do estilo de interação. Padrões são importantes no desenho da interação porque além de facilitar a utilização pelos usuários, promovem uma série de benefícios para o desenvolvimento, como por exemplo, promovem a reuso e contribuem para reduzir custos de desenvolvimento.

Neste capítulo, aspectos associados a padrões são tratados dentro da atividade de Definição do estilo de interação e os demais aspectos de desenho da interface do usuário são apresentados dentro da seção relativa ao Desenho da interação.

É importante mencionar que, diferente da situação observada na Análise de contexto de uso, agora vamos considerar que o nosso produto, ou o produto em perspectiva, estará apoiando as atividades realizadas pelos usuários. Sendo assim, as tarefas descritas na Analise de Contexto de uso que forem ser apoiadas pelo sistema automatizado, devem agora ser repensadas para que sejam executadas com o apoio do sistema em consideração. É importante notar, então, que na Análise de Contexto de Uso **registramos** como os usuários realizam suas tarefas, enquanto que no Desenho da Interação, vamos **projetar** como os usuários deverão executar suas tarefas com o apoio do sistema em perspectiva.



Figura 6-1 Garçom anotando pedido. Fonte: acessado na internet em 21/10/2009 em <http://melhoresbaresdorio.com.br/sem-categoria/dicas-de-como-atender-bem-segundo-a-visao-de-um-bohemio>

Por exemplo, vamos imaginar que estamos desenvolvendo um sistema para ser usado pelos garçons no atendimento aos clientes de um restaurante. Na Análise de tarefas podemos registrar que os garçons se utilizam de um bloco para anotar os pedidos dos clientes (Figura 6-1) e descrever como os garçons realizam o atendimento aos clientes. No entanto, no Desenho podemos decidir que queremos uma solução mais “tecnológica” e vamos projetar a utilização de computador de mão (PDA) para os garçons comandarem o pedido do cliente com comunicação direta com a cozinha do restaurante, Figura 6-2.



Figura 6-2 Dispositivo de mão a ser usado para comandar pedidos. Fonte: acessado na internet em 21/10/2009 em
http://www.coolbluehomes.com/home/handheld/choose_your_handheld_model_type.htm

Acima, nos referimos às “tarefas descritas na Analise de Contexto de uso que forem ser apoiadas pelo sistema automatizado” porque pode acontecer que desistamos de ter o software em consideração apoando alguma tarefa anteriormente descrita na Análise de tarefas. Isso pode acontecer devido à priorização, por falta de recurso, tempo ou por outro motivo relevante qualquer.

6.1 ATIVIDADE DE DESENHO DA INTERAÇÃO

A atividade de Desenho da Interação, conforme definida no Praxis-u, é apresentada na Figura 6-3 na forma de um diagrama (de atividades) UML. O Desenho da Interação é composto de três atividades principais: Modelagem conceitual, Modelagem de conteúdo e navegação e Desenho detalhado, que serão apresentadas nas seções seguintes.

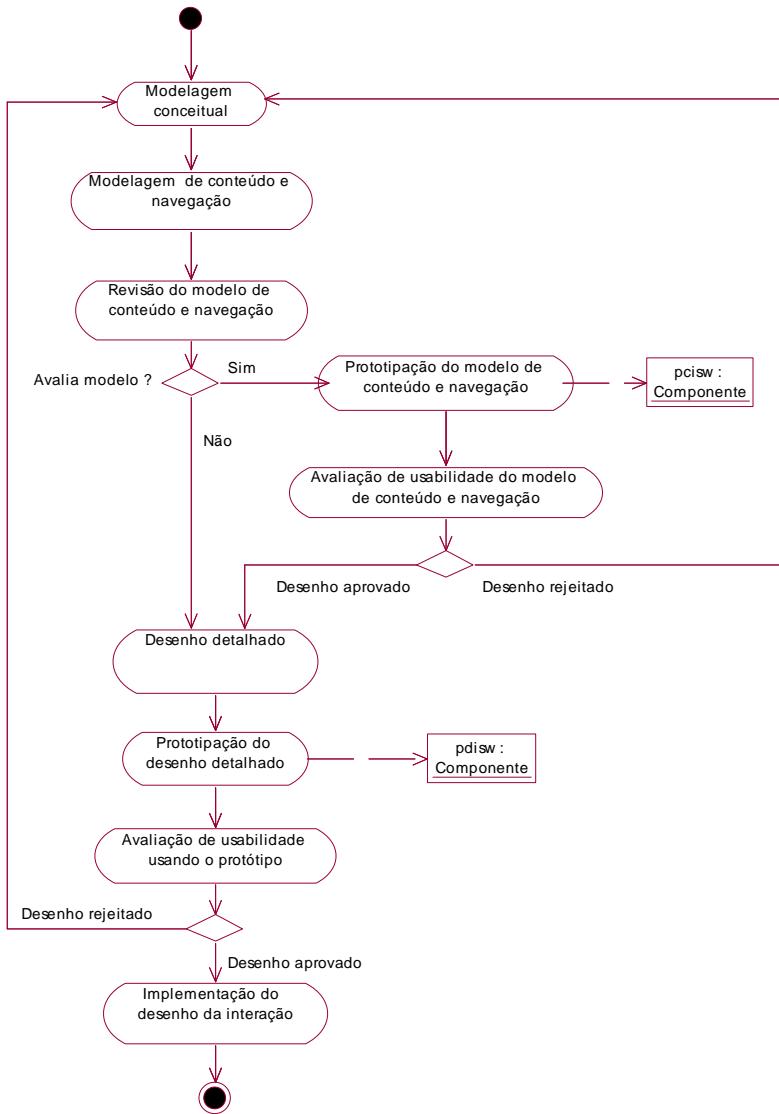


Figura 6-3 Subprocesso de Desenho da interação do Praxis-u

Na Figura 6-3 podemos observar que o processo prescreve a realização das atividades de Modelagem Conceitual seguida da Modelagem de conteúdo e navegação. A seguir, propõe uma atividade de revisão seguida de dois possíveis caminhos. A primeira possibilidade é seguir diretamente para o Desenho detalhado. A alternativa, que consideramos mais indicada, é desenvolver um protótipo correspondente ao Modelo de conteúdo e navegação,

denominado “pcisw”, e utilizar este protótipo em um teste de usabilidade com os usuários visando à validação do modelo. Se, na avaliação com usuários, o Modelo de conteúdo e navegação não for considerado satisfatório, o processo sugere que voltemos à atividade de Modelagem conceitual para a realização de melhorias e o ciclo se repete.

Posteriormente, o processo prescreve outro ciclo onde é feito um protótipo do desenho detalhado, esse protótipo é avaliado com usuários visando à validação, e, caso o desenho seja rejeitado, o processo também sugere uma volta à atividade de Modelagem conceitual para a realização de melhorias no desenho da interação em suas várias etapas.

6.1.1 MODELAGEM CONCEITUAL

A Modelagem Conceitual é um desenho da interação em alto nível de abstração. No nível de abstração considerado não estamos preocupados com a aparência, em termos de design gráfico, dos elementos de interface; isso será considerado em uma etapa posterior do desenho.

A atividade de Modelagem conceitual tem como objetivo definir os principais aspectos relativos à interface do usuário, incluindo a definição de quais vão ser e como vão ser realizadas as atividades que os usuários vão executar com o apoio do produto em consideração. Além das atividades em si, que constituem aspectos dinâmicos, devemos descrever também aspectos estáticos relacionados a essas atividades, por exemplo, os objetos e demais recursos envolvidos.

A Modelagem conceitual envolve três sub-atividades.

- **Definição do modelo conceitual estático:** define os objetos que farão parte do desenho e os relacionamentos entre eles
- **Definição dos modelos mentais:** modelos mentais ou metáforas a serem usados na interação são definidos.
- **Definição dos roteiros de desenho:** define os aspectos dinâmicos da interação, isto é, apresentamos uma definição de como as atividades dos usuários serão realizadas com o apoio do sistema em consideração.

DEFINIÇÃO DO MODELO CONCEITUAL ESTÁTICO

Nesta atividade, definem-se os objetos que atuarão na interação. Os objetos incluem pessoas, equipamentos, artefatos, recursos, etc que estarão envolvidos na interação.

É importante também definir relacionamentos entre os objetos. Na modelagem como utilizada em desenvolvimento de software, é importante descrever não apenas objetos ou conceitos, mas também modelar os relacionamentos ou conexões semânticas relevantes existentes entre eles.

Por exemplo, vamos considerar o sistema para ser usado pelos garçons no atendimento aos clientes de um restaurante, ilustrado na [Figura 6-2](#) acima. Podemos querer representar em um Modelo conceitual estático que a execução da tarefa de “atendimento aos clientes do restaurante” envolve os seguintes “objetos”.

- Garçom: recurso humano, empregado que serve à mesa.
- Dispositivos PDAs: computadores de mão a serem utilizados pelos garçons para comandar os pedidos diretamente à cozinha do restaurante.
- Cozinha: cozinha do restaurante, responsável por receber, executar e avisar quando os pedidos comandados pelos garçons estiverem prontos por meio dos PDAs.
- Pedido: relação de pedidos dos clientes de uma mesa.
- Mesa: mesa do restaurante, corresponde a um grupo de clientes que estão sendo atendidos juntos.
- ...

Os objetos são os listados acima; os relacionamentos entre eles podem ser vínculos como: cada mesa está associada a um grupo de clientes durante um atendimento; cada garçom utiliza-se de um dispositivo PDA do qual faz uso exclusivo; o garçom atende a várias mesas ao mesmo tempo, a Cozinha atende a todas as mesas, etc.

O Modelo conceitual pode ser descrito informalmente, como fizemos no exemplo acima, ou pode-se usar diagramas de classe ou de objetos da UML em uma descrição mais formal (Booch, G. et al, 2005), (Rumbaugh, J. et al, 2004). A decisão do que utilizar vai levar em conta a formação da equipe (conhecimento de UML) e a complexidade do modelo. Pode-se organizar os modelos em diagramas associados a tarefas ou a casos de uso do software em consideração.

DEFINIÇÃO DE MODELOS MENTAIS

A Definição dos modelos mentais visa definir os modelos mentais que serão passados aos usuários através da interface do software, para serem usados na realização de suas atividades suportadas pelo sistema.

Modelos mentais ou metáforas formam um pano de fundo onde se desenvolve o desenho da interação. Em geral, na interface desenhada vamos procurar utilizar os modelos mentais dos usuários levantados na Análise de Contexto. Isso porque os usuários já conhecem e se utilizam desses modelos na realização de suas atividades. No entanto, pode ser que se descubram modelos conceituais que sejam melhores em algum aspecto do que aqueles utilizados no momento atual pelos usuários. Nesse caso, devemos considerar se vale realmente utilizar esses novos modelos em substituição aos modelos usados atualmente, considerando o custo e benefícios. Os custos estão ligados principalmente à necessidade dos usuários de terem que aprender e se adaptar aos novos modelos, os benefícios podem ser em termos de eficiência ou eficácia na interação, que permita um melhor uso dos recursos, humanos ou de outro tipo, envolvidos.

Os modelos mentais, então, serão utilizados pelos usuários na interação. Os modelos mentais serão passados através das imagens ou outros meios que serão colocadas na interface. No entanto, modelos mentais mais complexos podem exigir treinamento para que sejam passados aos usuários.

Como em todo trabalho de modelagem, deve haver uma análise de alternativas, pesando os prós e contras de cada uma delas.

DEFINIÇÃO DOS ROTEIROS DE DESENHO

Esta atividade visa à definição de aspectos dinâmicos associados à interação, isto é, a definição de como as atividades dos usuários serão realizadas com o apoio do sistema em consideração. O resultado dessa atividade é uma visão externa, ou seja, uma visão do ponto de vista dos usuários.

Sugerimos a técnica de Roteiros para ser utilizada para a descrição da interação. Porém, diagramas de atividades, de estado ou de interação ou modelos informais também podem ser utilizados para complementar, ou mesmo para compor, a descrição.

Roteiros são histórias sobre pessoas e suas atividades (Rosson & Carroll, 1990), como já apresentado no Capítulo 4 - Análise de contexto de uso. A técnica de Roteiros usada no Desenho da interação é a mesma utilizada no domínio do Problema, na Modelagem de tarefas. Entretanto, enquanto no domínio do problema os roteiros descrevem a situação corrente onde o sistema em perspectiva ainda não aparece, no Desenho da Interação os Roteiros vão ser usados para descrever o modo **projetoado** de como as atividades vão ser realizadas com o apoio do sistema em consideração.

Os Roteiros utilizados são histórias que descrevem um cenário onde uma tarefa de interesse é realizada com o apoio do sistema em perspectiva. Não se trata de uma simples descrição de tarefas, como vimos, mas de uma descrição enriquecida com aspectos relevantes como a caracterização das pessoas envolvidas, os objetivos da tarefa, as estratégias utilizadas pelas pessoas envolvidas na realização das tarefas, etc. Estes aspectos constituem informações importantes usadas no desenvolvimento da interação, especificamente no Desenho detalhado, para o produto em consideração. Naturalmente, os Roteiros de domínio de problema constituem a principal fonte de informação para a criação dos Roteiros de Desenho da interação.

Podemos dizer que os Roteiros estão para a usabilidade assim como os casos de uso estão para a engenharia de software. Mas os Roteiros envolvem um contexto mais amplo que o representado pelos casos de uso da engenharia de software, já que os casos de uso são utilizados para descrever apenas as partes de uma tarefa que envolvem a utilização do sistema de software. Já os Roteiros são usados também para descrever outras partes das tarefas além de estritamente aquelas em que o sistema está apoiando sua realização.

O conteúdo de um Roteiro de desenho da interação é semelhante ao conteúdo de um Roteiro de domínio de problema apresentado no Capítulo 4 e reproduzido aqui na Tabela 6-1. Em geral, os Roteiros descrevem uma instância da interação, mas alternativas importantes podem também ser descritas.

Aspectos usados no Roteiro	Significado dos aspectos usados no roteiro
Cenário	Detalhes da situação que motivam ou explicam os objetivos, ações, reações dos atores. O software em perspectiva apóia a realização da atividade modelada.
Atores	Papéis de pessoas que interagem com o ambiente do cenário; eventualmente com suas características relevantes.
Objetivos da tarefa	Aspectos da situação que motivam as ações realizadas pelos atores.
Modelos mentais	Modelos conceituais e atividade mental de que os atores se utilizam para converterem objetivos em comportamentos.
Avaliação	Atividade mental dos envolvidos que visa interpretar características da

	situação.
Ações	Comportamento observável das pessoas. Deve envolver o uso do sistema em consideração.
Eventos	Ocorrências externas ou internas que podem influenciar a atividade. Podem não ser percebidas pelos atores, mas são importantes para o roteiro.

Tabela 6-1 Roteiros de desenho da interação

O Roteiro apresentado no exemplo mostrado no Capítulo 4, seção 4.4.3, descreve um cenário onde um cliente (José) de um supermercado deseja comprar um vinho para presentear um amigo e o vendedor (James) do setor de vinhos do supermercado vai apoiar a venda. O quadro abaixo utiliza este mesmo exemplo, só que agora considerando o uso de um software que está sendo desenvolvido para apoiar a tarefa de venda de vinhos.

Observe que as informações contidas no Roteiro de domínio de problema foram utilizadas. Por exemplo, cliente de que o cliente do supermercado ficaria constrangido em deixar transparecer que não conhece de vinho, os desenvolvedores colocaram essa informação como parte do software de modo que o cliente se sinta à vontade para fazer suas consultas. Também a faixa de preço do presente que o cliente tem em mente é informada ao sistema em um contato impessoal.

Roteiros de desenho também devem ser complementados com uma análise de argumentos. Para isso, são identificadas possíveis variações relacionadas aos roteiros identificados e analisados prós e contras. Por exemplo, poderia ser considerada a utilização de um formato de hipertexto na interface. Isso teria a vantagem de permitir ao usuário consultas no nível de detalhamento que julgar conveniente. Além disso, o hipertexto é uma forma eficiente de organizar a informação. No entanto, há o risco de o usuário ter dificuldade na navegação senão tiver familiaridade com esse formato.

James, um vendedor especializado em vinhos, trabalha no setor de vinhos de um supermercado sofisticado. O supermercado utiliza o produto Bem-Vinho para apoiar as vendas no setor de vinhos. José, um cliente, deseja comprar uma garrafa de vinho para presentear um amigo. José está indeciso sobre o que comprar, deseja aperfeiçoar a relação custo-benefício.

José tem em mente um presente na faixa de R\$ 50,00; ele navega pela interface do software Bem-Vinho a procura de sugestões de vinhos para sua compra. José teria reserva em revelar o valor que tem em mente para o presente ao vendedor, mas se sente a vontade para informá-lo ao software Bem-Vinho (esta seria uma das primeiras informações solicitada pelo software). O produto lista 10 sugestões de vinhos na faixa de preço pretendida por José.

O vendedor está posicionado discretamente, pronto para ajudar se solicitado. James sabe que, muitas vezes, o cliente se sente constrangido em transparecer que entende muito pouco (quase nada) de vinhos e que não se sente a vontade para informar (pessoalmente ao vendedor) a faixa de preço do presente pretendida.

Eventualmente, José pede a opinião do vendedor. James, pergunta sobre o gosto do amigo de José. Discretamente, consulta o software Bem-Vinho para conhecer mais detalhes do perfil de vinho desejado pelo cliente, principalmente a faixa de preços que o cliente se dispõe a gastar. James comenta sobre as várias opções de tipos de vinho, fornecidas pelo Bem-Vinho, de acordo com o perfil desejado pelo cliente.

O Bem-Vinho é utilizado pelo cliente ou pelo vendedor para consultas e eventuais detalhamentos de alguma informação. Em particular, com auxílio do Bem-Vinho, pesquisa e apresenta as ofertas de produtos que se encaixem no perfil de vinho desejado pelo cliente, visando oferecer um bom serviço.

De repente, outro cliente pede ajuda ao vendedor. James deixa José "se distrair" utilizando o Sistema e passa a atender ao outro cliente. James fica tranquilo ao deixar o cliente anterior porque sabe que ele pode ir se ocupando com o sistema Bem-Vinho.

6.1.2 MODELAGEM DE CONTEÚDO E NAVEGAÇÃO

Após a modelagem conceitual, chegamos ao ponto de nos preocupar com a interface mais concretamente, fazendo o desenho da arquitetura da interface. A Modelagem de conteúdo e navegação envolve os aspectos estáticos ou estruturais e dinâmicos. O aspecto estrutural corresponde à criação de um modelo simplificado do conteúdo da interface e o aspecto dinâmico corresponde à definição da navegação associada ao modelo estrutural.

O Modelo de conteúdo e navegação define a interface em termo de Espaços de interação, conforme proposto por Constantine e Lockwood (Constantine & Lockwood 1999). Um Espaço de interação é um “espaço de trabalho” onde o usuário tem disponível ferramentas para realizar suas atividades utilizando o software. O Espaço de interação pode ser visto como uma metáfora de um atelier ou um ambiente virtual de trabalho. No software, um Espaço de interação vai corresponder a agrupamento de telas (ou páginas web), telas, ou mesmo partes de telas, ainda sem a preocupação com o design gráfico.

A parte estática do Modelo de conteúdo e navegação descreve a interface em termos de distribuição de conteúdos em espaços de interação. Este modelo deve ser colocado na forma de uma hierarquia de espaços de interação.

Os autores Constantine e Lockwood (1999) sugerem uma maneira informal de representação desses modelos com a utilização de cartões de papel ou de “post-its” em um quadro representando a interface, onde cada cartão representa um Espaço de interação. A topologia dos cartões no quadro está associada à hierarquia. O Post-it é aquele pequeno papel adesivo, em geral amarelo, de fácil remoção, utilizado para se deixar recados ou notas para lembrança de compromissos ou coisas a realizar.

Aqui, nós sugerimos a utilização de textos explicativos e diagramas UML para notação do Modelo de conteúdo e navegação. Esse modelo tem as seguintes características:

- o modelo mostra uma hierarquia de espaços de interação;
- pacotes lógicos da UML são utilizados para representar a organização hierárquica entre espaços de interação;
- relacionamentos de associação, com estereótipos <<web page link> entre Espaços de interação indicam possibilidade de navegação;

- relacionamento de associação, com estereótipo <<affinity>>, indicam semelhança de espaços ;
- relacionamentos de agregação ou composição podem indicar relação parte-todo entre espaços de interação, tipicamente, entre elementos de interação e suas telas (ou páginas);
- relacionamentos de generalização podem representar herança entre espaços de interação.
- diagramas de sequência ou de comunicação com objetos representando telas (ou páginas Web) e mensagens indicando navegação também são utilizados para representar o aspecto dinâmico de navegação.

Um exemplo ilustrativo desse modelo pode ser encontrado no sítio do professor (<http://homepages.dcc.ufmg.br/~clarindo/disciplinas/eu/material/index.htm>).

O Modelo de conteúdo e navegação pode ser implementado na forma de protótipo, o que facilita a realização de avaliações de usabilidade com usuários visando à validação do Modelo, conforme mostrado na Figura 6-3 acima. O modelo baseado em UML que sugerimos pode, inclusive, ser usado diretamente como um protótipo para avaliações com os usuários. Ou então, pode-se criar um protótipo navegável a partir desse modelo, o que pode ser feito facilmente com ferramentas de prototipagem ou mesmo com ferramentas de criação de apresentações.

6.1.3 DESENHO DETALHADO

A atividade de Desenho detalhado visa chegar ao nível do desenho de telas em um processo que envolve refinamentos e melhorias sucessivas. Nessa atividade, o desenho conceitual e o modelo de conteúdo e navegação são utilizados como insumo, e chegamos ao nível de nos preocupar com a aparência de objetos, navegação entre telas, redação de mensagens, rótulos, etc. Como resultado final, temos o desenho da interface pronto para ser implementado no desenvolvimento de um produto de software.

A atividade de Desenho é complexa. Há uma vasta literatura disponível sobre o assunto, sendo assim, vamos preferir somente indicar algumas referências sobre o assunto. Por exemplo, Shneiderman et al. (2009) têm um livro considerado com uma referência, com extensa cobertura do assunto. O livro de Van Harmelen (2001) é um livro de editor que apresenta várias técnicas e métodos para o desenho de interface. Podem ser consultados

também (Hix & Hartson 1993), (Galitz 2007), (Krug & Black 2005), (Rosson & Carroll 2002), dentre outros.

6.2 GUIA DE ESTILO DE USABILIDADE

A utilização de padrões no desenho da interface do usuário traz benefícios tanto para os clientes/usuários como também para a equipe de desenvolvimento do software. Dada a importância que representa a utilização de padrões, o Praxis-u prescreve que, em adição ao Desenho da interação, a atividade de desenho da interface inclua também a atividade de Definição do estilo de interação para tratar de aspectos de desenho ligados a padrões.

A atividade de Desenho do estilo de interação define padrões ou estilos a serem utilizados internamente pela equipe de desenvolvimento da interface com o usuário. Esses padrões são registrados no documento Guia de estilo de usabilidade (geusw).

O objetivo do Guia de estilo de usabilidade é estabelecer diretrizes e padrões para o desenvolvimento de software e, em alguns casos, descrever também métodos para garantir a consistência entre produtos de uma “família” de produtos ou mesmo dentro de um produto.

Um Guia de estilo de usabilidade pode, então, abranger um produto de software ou uma família de produtos com algum vínculo que os une. Por exemplo, o conjunto de produtos de software de um mesmo cliente de uma empresa desenvolvedora de software pode estar submetido a um mesmo padrão estabelecido em um Guia.

A utilização de padrões no desenvolvimento de interfaces traz vários benefícios, como listados a seguir.

- Garante a consistência em uma família de produtos. Uma das principais vantagens que justificam a utilização de padrões é justamente a de estabelecer uma base comum que facilite a consistência em relação a diversos aspectos associados à interface.
- Contém um repositório de diretrizes e padrões, fonte de consulta para os desenvolvedores, principalmente, mas também para outros atores envolvidos no desenvolvimento de software.
- É uma ferramenta para o treinamento de novos desenvolvedores que têm o Guia com fonte de consulta.
- Facilita o trabalho em conjunto de grupos.

Para os usuários, especificamente, podemos listar os seguintes benefícios.

- Redução de erros na utilização do software.
- Aumento da confiança na utilização de um produto cuja interface se utiliza de padrões que lhes são familiares.
- Aumento da eficiência na utilização do software.
- Redução da resistência dos usuários à nova tecnologia que um produto de software pode representar.

A organização desenvolvedora também pode se beneficiar do uso de padrões.

- Reduz tomadas de decisões arbitrárias.
- Minimiza a reinvenção, beneficia o re-uso
- Diminui o tempo de desenvolvimento.
- Redução de testes pela utilização de elementos padronizados confiáveis. Neste aspecto, também é um instrumento que ajuda a reduzir erros encontrados em avaliações de usabilidade.

O Guia de estilo de usabilidade deve ser desenvolvido em um trabalho de grupo para que todos os envolvidos estejam de acordo. O comprometimento da equipe é importante para o sucesso na definição e uso do Guia. Sendo assim, idealmente, um Guia de estilo de usabilidade deve envolver a equipe de usabilidade ou de desenho da interface, outros desenvolvedores da área de engenharia de software, profissionais responsáveis por documentação, design gráfico e marketing, usuários, clientes e especialistas no domínio. Claro que nem sempre todos esses profissionais estão disponíveis ou mesmo participam de um projeto de desenvolvimento de software; nesse caso, faz-se o melhor possível com as pessoas disponíveis.

Com sugestão, um Guia de estilo de usabilidade pode ter a seguinte estrutura que é utilizado no artefato "geusw"do Praxis-u.

1. Introdução: define o contexto de utilização do Guia. Deve incluir os seguintes aspectos ou seções no documento:

- Objetivos do documento de Guia de Estilo;
- audiência esperada do documento;
- como se espera que o Guia seja utilizado;

- um histórico de acontecimentos relevantes para mostrar o contexto onde o Guia será utilizado e o escopo ou abrangência do documento;
- definição de consistência com outros Guias ou documentos;
- referência a outros documentos relevantes para o leitor;
- identificação do cliente e fornecedor;
- dados do projeto;
- plataforma utilizada e outras decisões importantes que forem tomadas nas reuniões entre os grupos envolvidos.
- organização do documento;

2. Conceitos preliminares: dá uma visão geral de usabilidade, apresentando os principais conceitos relacionados ao assunto. Lembre-se que o Guia de estilo pode ser utilizado por usuários e outras pessoas que podem não ter muita familiaridade com a usabilidade.

3. Diretrizes Gerais: diretrizes de usabilidade que se aplicam a qualquer tipo de produto. Essas diretrizes servem com uma espécie de padrão mais “leve”, alertando os desenvolvedores sobre diretrizes importantes para o desenho da interface do usuário.

4. Padrões específicos da família de produtos <nome da família de produtos>: pode ser organizado com o seguinte conteúdo.

1. Aspectos gerais: apresenta padrões em mais alto nível de abstração relativos à organização e formato de telas ou páginas Web e aspectos do ambiente de programação que impõem restrições ao desenho da interface.
2. Padrões de comportamento da interface: apresenta aspectos de comportamento da interface, como navegação, forma de pesquisa, etc.
3. Elementos de interação: deve existir uma seção separada para cada objeto de interação. Dentro de cada seção deve haver figuras ilustrativas de como deve ser o layout do objeto e as diretrizes aplicáveis a ele.
4. Mensagens: essa seção deve apresentar diretrizes relativas às mensagens de erro, mensagens informativas e feedback do progresso de uma tarefa, etc, ou seja, todos os tipos de mensagens, incluindo figuras ilustrativas, devem estar nesta seção.
5. Tratamento da ajuda aos usuários: descreve padrões relativos à ajuda online aos usuários. Padrões de manuais de usuários são normalmente considerados associados aos processos de desenvolvimento.

6. Dispositivos de interface de hardware: descreve os dispositivos de entrada e saída para os quais o Guia se aplica.

5. Padrões específicos de produtos da família: opcional. Os “Padrões específicos de produtos da família” vão descrever padrões específicos de cada produto da família. Os padrões de cada produto específico são vistos como uma especialização (no sentido Orientado a Objeto do termo) dos “Padrões específicos da família de produtos <nome da família de produtos>”. As seções que descrevem cada produto e a seções que descrevem a família de produtos têm a mesma estrutura, e qualquer aspecto definido para a família de produtos é considerado válido para todos os produtos da família e não precisam ser repetidos. Você só vai usar a parte de padrões de cada produto se quiser modificar (sobrescrever) o que já foi definido para a família de produtos, como funciona no conceito de herança em O-O. Observe , ainda, que o Guia permite descrever vários produtos de uma mesma família, que apareceriam em seções 5.1, 5.2, ..., quantos forem necessários!

6. Glossário

7. Bibliografia:

citar bibliografia relevante para os usuários.

Vários tipos de consistência podem ser importantes e devem ser objeto de padronização registrada do Guia de estilo de usabilidade, como se segue.

- Consistência com outros guias de estilo usados. Pode ser interessante criarmos uma hierarquia de guias de estilo. Por exemplo, considerando o chamado “governo eletrônico”, as instâncias de governo federal, estadual e municipal, cada uma, pode definir padrões que se aplicam a qualquer sítio de governo eletrônico. Neste caso, podemos imaginar uma hierarquia, onde o governo municipal deve seguir os padrões estabelecidos em nível estadual e o governo estadual deve seguir os padrões estabelecidos em nível federal.
- Consistência em relação ao aspecto visual da interface.
- Consistência em termos de mensagens de erro.
- Consistência de comportamento dos objetos de interação usados na interface.
- Consistência com as expectativas do usuário. Ou seja, uma interface deve ser desenvolvida considerando os usuários, seus comportamentos e expectativas.

Os seguintes passos são recomendados na criação de Guia de Estilo.

1. Definição das pessoas que vão desenvolver o Guia. Deve incluir pessoas dos vários perfis de profissionais descritos anteriormente.
2. Definição da plataforma para qual o Guia vai ser desenvolvido.
3. Confecção da estrutura do documento.
4. Escolha das diretrizes que serão colocadas.
5. Definição de requisitos que o Guia de Estilo deve satisfazer. Definição dos objetos de interação que o Guia de Estilo deve abordar.
6. Confecção dos gabaritos (*templates*) de telas e dos exemplos dos objetos de interação.
7. Montagem do documento.
8. Revisão.

Vários fatores podem levar ao fracasso na adoção do conceito de Guia de estilo no desenvolvimento da interação. Por exemplo, os gerentes de projeto podem negligenciar os benefícios do Guia de Estilo e não promoverem adequadamente sua utilização. Ou um Guia de Estilo pode ser muito amplo, perdendo muito de sua aplicabilidade; neste caso, deve ter seu foco reduzido. Pode ser também que a equipe de desenvolvimento fique relutante em adotá-lo - aí cabe a atuação da gerência para buscar fomentar sua utilização efetiva.

Outro risco é que o Guia se torne maçante se tiver muita informação textual. Por isso, é muito importante que o Guia de estilos de usabilidade seja rico em exemplos práticos e ilustrações para facilitar e tornar até mais agradável sua utilização. Para facilitar a consulta, é importante que o Guia tenha um sumário. Outro fator que pode colocar a utilização do Guia em risco é a discordância entre membros da equipe que o desenvolve. Sendo assim, é importante o trabalho de uma equipe motivada e comprometida.

Concluindo, a elaboração de um Guia de estilos de usabilidade envolve o trabalho de grupos de diversas áreas da empresa/instituição. Ele sozinho, obviamente, não garante o sucesso do design, mas é um documento que contribui muito para garantia de padrões e agilidade no desenvolvimento.

6.3 GLOSSÁRIO

Sem conteúdo

6.4 BIBLIOGRAFIA

Booch, G.; Rumbaugh, J.; Jacobson, I., *Unified Modeling Language User Guide*, 2nd Edition, Addison Wesley, 2005.

Constantine, L.L. & Lockwood, L.A.D. *Software for Use: a Practical Guide to the Models and Methods of Usage Centered Design*, Addison-Wesley, Reading-MA, 1999.

Galitz, W. O. *The Essential Guide to User Interface Design*. John Wiley, 2007.

Krug, S.; Black, R. *Don't Make Me Think: Common Sense Approach to Web Usability*, 2nd edition, New Riders, 2005

Nielsen, J. *Designing Web Usability*, New Riders Press, 2000.

Rosson, M. B., Carroll, J.M. *Usability Engineering: Scenario Development of Human-computer Interaction*. Morgan kaufmann Publishers, 2002.

Rumbaugh, J.; Jacobson, I.; Booch, G., *The Unified Modeling Language Reference Manual*, Addison Wesley, 2nd edition, 2004.

Shneiderman, B; Plaisant, C.; Cohen, M.; Jacobs, S. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 5 ed. Reading, MA, Addison-Wesley, 2009

Van Harmelen, M. *Object Modeling and User Interface Design: Designing Interactive Systems*, Addison-Wesley, 2001.

Hix, D.; Hartson, H. R. *Developing User Interfaces: ensuring usability through product & process*, John Wiley and Sons, 1993.

7 AVALIAÇÃO DE USABILIDADE

A atividade de Avaliação de usabilidade prescrita pelo Praxis-u visa à avaliação da qualidade da interface como instrumento da interação usuário-computador. Pode ser considerada como uma atividade de garantia da qualidade dentre outras que são utilizadas no processo de desenvolvimento de software.

Diferentes tipos de avaliação, com objetivos e características próprias, podem ser utilizados. Existem técnicas de avaliação, que utilizam o formato de revisões. As avaliações mais confiáveis, no entanto, envolvem experimentações com a participação de representantes dos usuários. Estas avaliações, também chamadas de testes com os usuários, normalmente fazem uso de roteiros de tarefas que são executadas por usuários com a utilização de protótipos ou com o produto final, dependendo do estágio de desenvolvimento do software. Representantes dos usuários são convidados a executar as tarefas previstas no roteiro enquanto um avaliador/observador se coloca ao lado analisando a qualidade da interação e identificando possíveis problemas.

Este capítulo apresenta a atividade de Avaliação de usabilidade prescrita no Praxis-u. O processo de avaliação é semelhante para as várias técnicas de avaliação que podem ser utilizadas, no entanto, neste documento, vamos nos enfatizar a técnica de Teste de usabilidade com o usuário.

7.1 OBJETIVOS

A avaliação de usabilidade pode ter o caráter formativo ou somativo. A avaliação formativa é realizada durante um projeto de desenvolvimento de software com o objetivo de aperfeiçoar o produto. A avaliação somativa é realizada, em geral, com o produto concluído, visando julgá-lo ou compará-lo com produtos semelhantes. A avaliação somativa pode também ser utilizada como critério de aceitação de um produto, ou seja, como parte dos requisitos não funcionais acordados com os usuários.

As avaliações formativas devem ser realizadas o mais cedo possível durante o desenvolvimento do software, em respeito ao princípio que preconiza que quanto mais cedo se detecta um problema, menor o custo de sua solução. Sendo assim, para se fazer avaliações antes do desenvolvimento do produto final, nas avaliações formativas, muitas vezes são utilizados protótipos.

Apesar da necessidade de se avaliar as interfaces o mais cedo possível, isso não significa que deva se avaliar soluções “toscas”, sem muita reflexão, já que o custo das avaliações também precisa ser levado em conta. O custo das avaliações inclui não somente o aspecto financeiro, mas também o possível desgaste dos desenvolvedores/fornecedores com os usuários ou clientes ocasionado por uma solução ruim submetida de maneira intempestiva. No entanto, é preciso não confundir um protótipo simples com uma solução ruim ou inadequada para a

avaliação; um protótipo simples, por exemplo, desenhado a mão, pode ser adequado e útil para uma avaliação de aspectos específicos da interação.

As avaliações de usabilidade são indicadores da qualidade da interação usuário-sistema. A detecção de problemas de usabilidade por meio de uma avaliação permite diagnosticar, em determinados contextos de operação do produto, quais objetivos foram atingidos.

A avaliação de usabilidade de um sistema interativo tem os seguintes objetivos gerais:

- verificar a qualidade da interface em termos de sua adequação para que os principais atributos de usabilidade sejam alcançados;
- validar os requisitos e metas de usabilidade, verificando o desempenho dos usuários face aos objetivos estabelecidos;
- validar a eficácia da interação humano-computador face à efetiva realização das tarefas por parte dos usuários;
- verificar a eficiência dessa interação, face aos recursos empregados.
- obter dados qualitativos que sirvam de insumo para uma melhoria da qualidade da interface;
- obter indícios da satisfação ou insatisfação (efeito subjetivo) que a interface possa trazer ao usuário.

Os objetivos específicos são:

- avaliar o desempenho, constatar, observar e registrar problemas efetivos de usabilidade durante a interação;
- obter métricas objetivas para eficácia, eficiência e produtividade do usuário na interação com o sistema. Por exemplo, podem ser analisadas métricas relacionadas ao tempo gasto, a quantidade de incidentes ou de erros dos usuários, passos desnecessários na realização de tarefas e busca de ajuda, dentre outros;
- diagnosticar as características do desenho que provavelmente atrapalham a interação por estarem em desconformidade com padrões implícitos e explícitos de usabilidade;
- prever dificuldades de aprendizado na operação do sistema;
- avaliar o desempenho dos usuários, na utilização do sistema, em relação aos atributos de usabilidade considerados mais importantes como, por exemplo, produtividade, prevenção de erros, facilidade de aprendizagem e retenção do conhecimento de como utilizar o sistema;

- conhecer a opinião do usuário em relação ao sistema;
- sugerir as ações de re-projeto mais evidentes para melhorias considerando-se os problemas de interação efetivos ou diagnosticados.

7.2 TÉCNICAS

Para realizar a avaliação de um sistema interativo, podem-se empregar várias técnicas que podem ser classificadas, dependendo da estratégia utilizada, em analítica, experimentais e de pesquisa de opinião. As técnicas analíticas são realizadas por especialistas em desenvolvimento de interface através de revisões de protótipos ou do produto final buscando avaliar a qualidade da interação proporcionada pela interface. As técnicas empíricas ou experimentais objetivam detectar problemas de usabilidade por meio da observação do usuário interagindo com os protótipos ou a interface finalizada, através de experimentos controlados. As técnicas de pesquisa de opinião buscam avaliar a satisfação do usuário através de técnicas de questionários e ou entrevistas, com o objetivo de se antecipar à reação do usuário com relação ao produto.

Cabe observar a participação de usuários em técnicas analíticas só faz sentido se eles têm um conhecimento de interface que lhes permita contribuir na análise da qualidade da interação do produto sob avaliação. Normalmente, a participação dos usuários é mais indicada nas técnicas experimentais ou de pesquisa de opinião onde sua contribuição será baseada em sua experiência na utilização do sistema em consideração ou no uso de sistemas semelhantes. Ou seja, nestas técnicas se espera que os participantes sejam representativos de perfis dos usuários e que se comportem nada mais, nada menos, do que como usuários típicos.

Cabe também observar que não somente problemas, mas também aspectos positivos da interface, podem ser interessantes e merecem ser registrados nas avaliações de usabilidade. Isso não só por contribuir para a auto estima da equipe, mas também porque uma boa solução deve ser lembrada para, quem sabe, ser utilizada em outros locais ou situações em que se apliquem.

7.2.1 ANALÍTICAS

As técnicas analíticas em geral são realizadas por projetistas ou especialistas em usabilidade, podendo até dispensar a participação de usuários. As técnicas analíticas mais conhecidas são as avaliações heurísticas e as inspeções através de listas de conferência.

7.2.1.1 Avaliação heurística

A avaliação heurística deve envolver a participação de especialistas componentes da equipe de desenvolvimento da interação ou convidados externos. Esses convidados podem incluir especialistas em usabilidade, convedores do domínio de que trata o software em consideração, desenvolvedores da área de engenharia de software, pessoas da área de marketing ou de comunicação do cliente, ou mesmo usuários. A composição da equipe dependerá do tipo de produto e das condições específicas do projeto.

A avaliação heurística é realizada considerando-se um conjunto de regras ou diretrizes que são observadas para identificar possíveis problemas na interação humano-computador que provavelmente os usuários encontrarão. Esse tipo de avaliação é baseado no conhecimento e na experiência de avaliadores especialistas, que analisando as interfaces de um determinado sistema fazem o levantamento dos possíveis problemas e sugerem soluções. Posteriormente, os resultados da avaliação de cada avaliador são organizados em um único relatório, onde resultados iguais ou similares são agrupados e depois categorizados em função da gravidade do problema. Segundo Nielsen (1993), três a cinco avaliadores são suficientes para identificar a maior parte dos problemas.

Cabe observar que a primeira etapa de análise deve ser efetuada individualmente por cada avaliador para evitar que um influencie outros. A segunda etapa, no entanto, onde os envolvidos se reúnem para analisar e priorizar os problemas encontrados, deve ser um trabalho de equipe.

7.2.1.2 Inspeções com listas de conferência (Checklists)

A avaliação de usabilidade por inspeção com listas de conferência é realizada por meio de vistorias através das quais profissionais diagnosticam rapidamente problemas gerais e repetitivos das interfaces (Bastien & Scapin 2009) e (Cybis, Betiol, e Faust 2007). Esses profissionais podem ser programadores, analistas, ou especialistas em usabilidade. Nesse tipo de técnica, a qualidade das listas determina as possibilidades de avaliação.

A utilização de listas de conferência tem a vantagem de poder suprir a carência de profissionais especializados na equipe avaliadora, considerando que as listas foram desenvolvidas por pessoal altamente capacitado. A utilização de listas de conferência geralmente produz resultados mais uniformes e abrangentes, em termos de identificação de pequenos problemas, visto que os avaliadores são conduzidos no exame da interface por meio de uma lista de questões a responder sobre a usabilidade do produto. No entanto, é importante considerar que a qualidade das listas influencia a qualidade do resultado final.

A avaliação com listas de conferência pode ser combinada com a avaliação heurística para se alcançar as vantagens das duas abordagens. Neste caso, recomenda-se que a avaliação pelos especialistas seja feita primeiramente como na avaliação heurística, em que os especialistas fazem sua avaliação livremente. Somente após uma primeira avaliação mais livre, os especialistas utilizariam as listas de conferência em uma segunda avaliação. O objetivo desta separação em duas avaliações é evitar que os avaliadores sejam dirigidos pelos ou se atenham somente aos itens que constem nas listas de conferência.

7.2.2 AVALIAÇÕES EXPERIMENTAIS

As técnicas experimentais, também chamadas de empíricas, de avaliação de usabilidade contam com a participação direta dos usuários, utilizando-se de experimentos. Compreendem basicamente os testes com usuários através do monitoramento de sessões de uso do produto, ou protótipo, em consideração.

Os teste de usabilidade com a participação dos usuários são consideradas as formas mais confiáveis de avaliação e, geralmente, as mais indicadas (Hix & Hartson 1993). Isso por ser muito difícil modelar ou prever o comportamento do ser humano na utilização de um produto de software. Dada a dificuldade de se prever o comportamento do ser humano, devido a sua enorme complexidade, a solução mais confiável para se validar a qualidade de uma interface em termos de usabilidade envolve a experimentação e observação do usuário realmente utilizando o produto ou um protótipo do produto.

Os testes empíricos avaliam a interface de um determinado produto por meio da simulação de uso do produto com participantes que sejam representantes da população dos usuários que utilizarão o sistema. Para a composição dos cenários de realização dos testes, são elaborados roteiros, cujo conteúdo é baseado no perfil dos usuários e nas suas tarefas típicas, que serão executados durante uma sessão de testes. Essas tarefas geralmente são aquelas já definidas na Especificação de requisitos de usabilidade.

Nesta técnica, os representantes de usuários devem executar as tarefas que constam dos scripts em sessões que são gravadas e acompanhadas por avaliadores no papel de monitores. Os monitores, especialistas em usabilidade e avaliação, têm a responsabilidade de conduzir as sessões de avaliação e coletar dados quantitativos e qualitativos que são posteriormente submetidos à análise visando à identificação de problemas e a indicação de soluções.

É importante salientar que uma das limitações apresentadas pelos testes é relacionada à representatividade da amostra testada. É imprescindível compor um grupo de usuários que incorpore, senão todas, pelo menos as principais características do público alvo do produto.

Outro ponto importante é relativo às condições de aplicação dos testes. Os testes não são situações reais de uso do sistema e por mais transparentes que possam parecer podem causar constrangimentos aos usuários. Desta forma é fundamental esclarecer que o alvo dos testes é o produto e não o usuário participante. Além disso, é dever do monitor que conduz os testes manter um ambiente confortável para que os participantes ajam com mais naturalidade.

7.2.3 PESQUISA DE OPINIÃO

As técnicas de pesquisa de opinião são baseadas na aplicação de questionários ou entrevistas com o usuário para avaliar o seu grau de satisfação em relação ao sistema e sua utilização.

A opinião do usuário ajuda a identificar problemas correntes do sistema em consideração. Esses problemas serão corrigidos ou servirão de referência para algum tipo de análise.

Em termos de qualidade, a satisfação do usuário está estritamente ligada ao atendimento daquilo que ele julga importante em um produto. Os dados resultantes da análise do questionário ajudam na identificação das reais necessidades do usuário, a orientar as revisões de projeto e a validar avaliações analíticas ou experimentais. Nesses casos, os avaliadores podem concentrar suas análises nos pontos problemáticos apontados pelo usuário.

A elaboração de questionários de avaliação requer uma capacitação específica para que sejam confiáveis e, portanto, deve contar com a participação de especialistas nessa área.

7.3 SUBPROCESSO DE AVALIAÇÃO DE USABILIDADE

Para ser eficaz e eficiente, a avaliação de usabilidade deve ser organizada dentro de uma metodologia bem definida. Diversas avaliações de usabilidade, às vezes utilizando-se diferentes métodos, podem ser executadas durante o ciclo de vida de desenvolvimento de um produto de software. Essas avaliações podem ser feitas com protótipos ou com o produto final, dependendo do momento e dos objetivos.

As avaliações devem ser cuidadosamente especificadas, desenhadas e planejadas, antes de serem realizadas. Assim como nos testes funcionais, durante e após a realização das avaliações de usabilidade, os resultados de cada avaliação devem ser analisados minuciosamente, comparando-se os resultados obtidos com as metas estabelecidas.

No Praxis-u, a atividade de Avaliação de Usabilidade do fluxo de usabilidade, que se desdobra em um sub-fluxo, visa uma avaliação da qualidade da interação proporcionada pela interface usuário-computador. Um planejamento inicial da freqüência e tipo de avaliações deve ser feito dentro da atividade de personalização do processo para projetos específicos. Um planejamento detalhado de cada avaliação deve ser realizado dentro do cada ciclo de desenho ou associado a liberações (versões parciais) do produto.

O planejamento das avaliações de usabilidade deverá ser documentado em um documento próprio: "dausw" - Descrição de Avaliações de Usabilidade. Um relatório de cada avaliação também deverá ser registrado em um relatório denominado "rausw" – Relatório de Avaliações de Usabilidade.

As principais referências utilizadas são: o processo de avaliação de produtos de software descritos na norma ISO/IEC 14598, o modelo de qualidade de software descrito na norma ISO/IEC 9126, considerando-se especificamente a usabilidade do produto, uma proposta de metodologia apresentada em (Cybis 2002) e o fluxo de testes de software descrito em (Padua, 2003). Outras referências importantes são (Hix & Hartson 1993), (Nielsen 1993), (Rubin 1994).

7.3.1 VISÃO GERAL

O processo de avaliação de usabilidade é constituído de três macro-atividades bem definidas para cada ciclo de avaliações realizadas: preparação, realização e análise de dados resultantes. Apesar de envolver atividades bem diferentes, as etapas de preparação e

realização das avaliações são semelhantes às atividades dos testes funcionais da engenharia de software (Padua, 2008). Durante a preparação, é elaborado o plano e são desenhadas as especificações de cada tipo de avaliação. Durante a realização, as avaliações são executadas, problemas de usabilidade são identificados e os dados coletados são registrados. Por fim, os dados são analisados: os defeitos encontrados são categorizados e avaliados com relação à sua gravidade, soluções são sugeridas e os relatórios de avaliação são redigidos. Se possível, antes mesmo da liberação final dos relatórios, pode-se corrigir alguns defeitos encontrados, observando-se as soluções sugeridas.

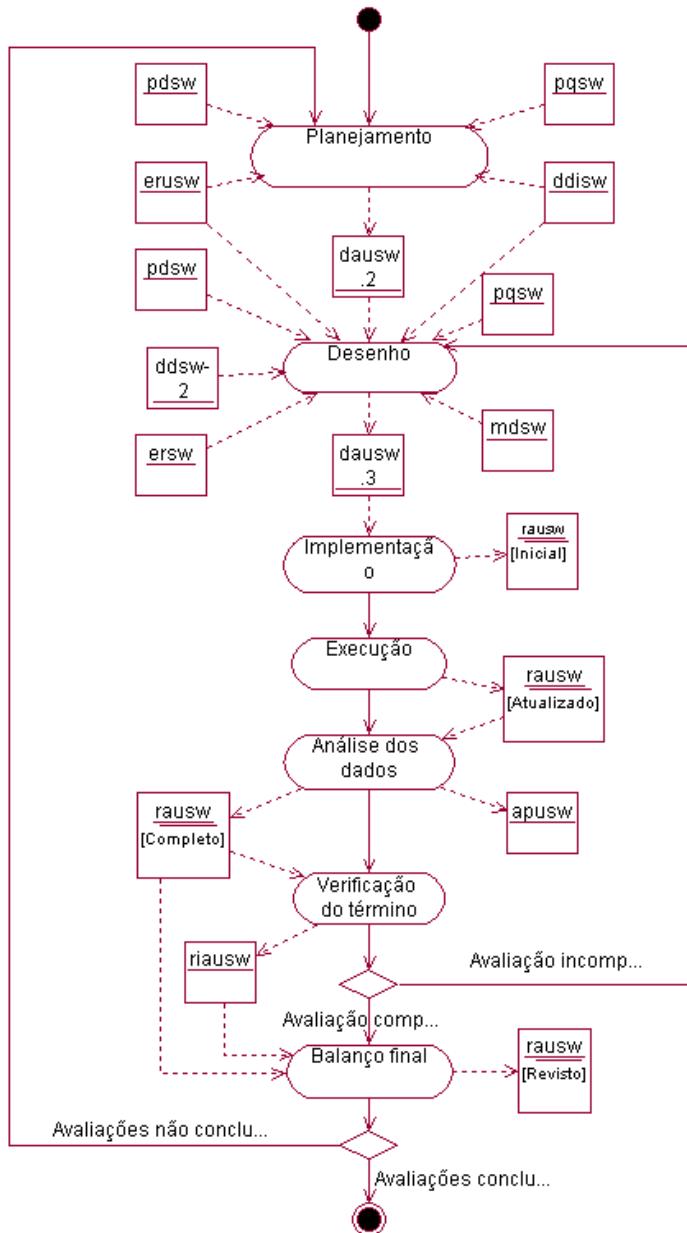


Figura 7-1 Diagrama de atividades do sub-fluxo de Avaliação de Usabilidade

O subprocesso de avaliação aqui proposto, Figura 7-1, prescreve as seguintes atividades.

- **Planejamento:** define as técnicas de avaliação mais adequadas para serem usadas em cada avaliação de usabilidade no ciclo de vida de desenvolvimento do produto em consideração. Para cada técnica, identifica os objetivos, componentes ou unidades a

avaliar, as sub-características de qualidade a serem observadas, a abordagem metodológica a ser adotada na avaliação, critérios de completeza e sucesso, critérios de suspensão e retomada, artefatos e resultados esperados da avaliação, ferramentas e equipamentos, responsabilidades pelas avaliações, agenda das avaliações nas iterações e identificação de riscos e contingências. No documento de Descrição de Avaliação de Usabilidade do Software ("dausw"), o resultado dessa atividade é apresentado nos planos específicos de cada técnica de avaliação.

- **Desenho:** configura as técnicas selecionadas para a avaliação. São detalhados os parâmetros específicos das técnicas utilizadas em cada avaliação específica. Define o protótipo ou produto a ser avaliado e os recursos a serem utilizados - infraestrutura, participantes (avaliadores, especialistas, usuários). Define também os procedimentos para a realização das avaliações e o material para a realização das avaliações. Por exemplo, a definição de roteiros de tarefas e cenários, o número de usuários a participar dos testes com usuários e o número de especialistas que irão realizar uma avaliação heurística devem ser definidos nessa atividade. No "dausw", os resultados dessa atividade são apresentados nas Especificações que detalham as avaliações.
- **Implementação:** prepara o ambiente para as avaliações, incluindo a execução de uma avaliação piloto, se prevista no método escolhido.
- **Execução:** executam-se as avaliações seguindo as indicações de cada técnica e coleta os dados.
- **Análise dos dados:** caracteriza os problemas, que são compilados, categorizados e priorizados; propõe soluções e elabora as recomendações para a implementação de melhorias.
- **Verificação do término:** inspeciona as avaliações, determinando se as condições para sua completeza e sucesso estão satisfeitas.
- **Balanço final:** realiza o balanço final das avaliações de usabilidade, verificando se os requisitos esperados para a atividade foram efetivamente alcançados e registrando as conclusões e lições aprendidas.

A preparação, realização e análise de dados de cada avaliação correspondem a uma passada completa pelo sub-fluxo de avaliação de usabilidade. Para cada etapa (fase ou iteração) do projeto de desenvolvimento do produto, podem-se combinar várias técnicas para testar o desenho ou a implementação da interface, da forma mais abrangente possível, dentro das limitações de recursos humanos e técnicos, custos e prazos.

7.3.2 DETALHAMENTO DAS ATIVIDADES

O resultado das atividades de avaliação de usabilidade que se seguem é documentado em dois artefatos (documentos): o “dausw” (Descrição de Avaliação de Usabilidade) e o “rausw” (Relatório de Avaliação de Usabilidade) que apresentam, respectivamente, toda a documentação gerada antes da avaliação e os resultados das avaliações relacionados com um projeto. Sendo assim, o “dausw” e o “rausw” são únicos para cada projeto e documentam todas as avaliações realizadas em um projeto.

O “dausw” é dividido em duas macroseções: a seção de Planos e a seção de Especificações. A relação entre um Plano e a respectiva Especificação para uma determinada técnica de avaliação corresponde ao conceito de herança na metodologia O-O (Orientada a Objetos), isto é, a Especificação de cada avaliação herda todas as definições feitas no respectivo Plano e eventualmente pode definir novos parâmetros ou alterar os parâmetros já definidos no plano, para uma avaliação específica. Por exemplo, a definição de roteiros de tarefas, cenários e número de usuários a participarem dos testes empíricos pode ser realizada em nível de Plano no “dausw” e valer para todas as avaliações documentadas nas Especificações. No entanto, estes parâmetros também podem ser alterados em uma Especificação correspondente a uma avaliação específica. A utilização do conceito de herança com relação às Especificações e aos Planos (em uma técnica específica de avaliação) permite que os parâmetros de desenho comuns a diversas avaliações sejam documentados somente no respectivo Plano.

O subprocesso de Avaliação de usabilidade mostrado na Figura 7-1 utiliza os seguintes artefatos.

- **pds**: Plano de Desenvolvimento do Software, documento indicado no Praxis que descreve, de forma detalhada, os compromissos que o fornecedor assume em relação ao projeto, quanto a recursos, custos, prazos, riscos e outros aspectos gerenciais.
- **pqs**: Plano da Qualidade do Software, documento do Praxis que descreve, de forma detalhada, os procedimentos de garantia da qualidade que serão adotados no projeto.
- **ers**: Especificação dos Requisitos do Software, documento do Praxis que descreve, de forma detalhada, o conjunto de requisitos especificados para um produto de software.
- **erusw**: Especificação de Requisitos de Usabilidade, documento do Praxis-u que descreve, de forma detalhada, a análise de contexto e os requisitos relacionados com a usabilidade.

- **ddsw:** Descrição do Desenho do Software, documento do Praxis que descreve, de forma detalhada, os aspectos mais importantes do desenho do software.
- **ddisw:** Descrição do Desenho da Interação do Software, documento do Praxis-u que descreve protótipos e aspectos mais importantes relacionados ao desenho da interação com o usuário.
- **dausw:** Descrição das Avaliações de Usabilidade, documento do raxis-u que descreve de forma detalhada o planejamento das avaliações de usabilidade
- **rausw:** Relatório das Avaliações de Usabilidade, relatório do Praxis-u que descreve os resultados das avaliações de usabilidade
- **mdsw:** Modelo de Desenho do Software, modelo do Praxis que detalha a estrutura lógica e física do produto, em termos de seus componentes.
- **apusw:** Modelo de Análise de Problemas de Usabilidade, modelo do Praxis-u utilizado na análise de problemas em avaliações de usabilidade
- **riausw:** Relatório de Inspeção de Avaliação de Usabilidade, relatório do Praxis-u que descreve se estão satisfeitas ou não as condições para o término de uma avaliação de usabilidade.

7.3.2.1 Planejamento

Como as avaliações de usabilidade podem ocorrer em diversos momentos durante o desenvolvimento de um produto de software, a atividade de Planejamento pode ser realizada junto com o Planejamento do processo de usabilidade conforme prescrito no Praxis-u e, posteriormente, pode ser completada com elementos de novas avaliações quando estas forem realizadas. O Planejamento da avaliação é composto das atividades de Identificação inicial dos requisitos da avaliação, Identificação dos itens a testar e Identificação detalhada dos requisitos da avaliação.

Cabe observar que estamos falando de uma avaliação formativa, realizada dentro de um processo de desenvolvimento do software. Sendo assim, todo o contexto que envolve a avaliação já é bem conhecido. Este não sendo o caso, por exemplo, em uma avaliação somativa, seria necessário um trabalho anterior visando um bom conhecimento do produto e a determinação de objetivos, contexto e requisitos de usabilidade envolvidos na avaliação.

7.3.2.1.1 Identificação inicial dos requisitos da avaliação

Essa atividade é realizada por meio de várias tarefas que são apresentadas detalhadamente na **Erro! Fonte de referência não encontrada..**

Descrição	Identificação inicial dos requisitos da avaliação.
Insumos	1 Especificação de Requisitos do Software (ERSw – requisitos de interface) 2 Especificação de Requisitos de Usabilidade (ERUSw – perfil do usuário, atributos e metas de usabilidade). 3 Descrição do Desenho da Interação do Software (DDISw) 4 Plano de Desenvolvimento de Software (PDSw). 5 Plano da Qualidade de Software (PQSw).
Tarefas	1 Levantar recursos existentes, identificando: <ul style="list-style-type: none"> ◦ pessoas (monitores de avaliação, especialistas, usuários, etc); ◦ hardware; ◦ software de sistema; ◦ ferramentas de teste; ◦ histórico de avaliações anteriores; ◦ formulários; ◦ suprimentos. 2 Escolher as técnicas para as avaliações, identificando: <ul style="list-style-type: none"> ◦ necessidades de estruturas provisórias; ◦ áreas de riscos que devem ser avaliadas; ◦ fontes existentes de casos de testes de usabilidade; ◦ etapa do ciclo de vida do produto de software; ◦ cronogramas e avaliação de custo/benefício; ◦ requisitos de recursos existentes ou necessários.
Resultados	1 Requisições de recursos para as avaliações. 2 Seleção de técnicas de avaliação de usabilidade.

Tabela 7-1 Identificação inicial dos requisitos da avaliação

7.3.2.1.2 Identificação dos componentes a testar

Essa atividade é realizada por meio de várias tarefas que são apresentadas detalhadamente na Tabela 7-2.

Descrição	Identificação dos componentes a testar.
Insumos	1 Especificação de requisitos de Software (ERS – casos de uso e desenho das telas de usuários) 2 Plano de Desenvolvimento de Software (PDSw) 3 Descrição do Desenho de Software (DDS – Desenho Arquitetônico e planos de

	liberação) 4 Descrição do Desenho da Interação do Software (DDISw)
Tarefas	1 Identificar: <ul style="list-style-type: none"> ◦ componentes a testar; ◦ atributos e metas de usabilidade dos componentes a testar (testes empíricos); ◦ status (situação de prototipação ou de desenvolvimento no momento da avaliação) dos itens a testar, dentro do projeto; ◦ características dos dados de entrada e saída (nos testes empíricos isso é essencial para a descrição das tarefas)
Resultados	1 Lista de componentes e aspectos a avaliar. 2 Eventuais pedidos de esclarecimentos aos projetistas de interface, relativos aos componentes a testar.

Tabela 7-2 Identificação dos componentes a testar

7.3.2.1.3 Identificação detalhada dos requisitos da avaliação

As tarefas a serem realizadas nessa atividade são apresentadas detalhadamente na Tabela 7-3.

Descrição	Identificação detalhada dos requisitos da avaliação.
Insumos	1 Lista de itens e aspectos a testar. 2 Informação geral da identificação inicial dos requisitos da avaliação.
Tarefas	1 Determinar: <ul style="list-style-type: none"> ◦ requisitos de recursos específicos dos componentes a testar; ◦ detalhes da abordagem; ◦ cronograma detalhado. 2 Especificar condições de completeza das avaliações: <ul style="list-style-type: none"> ◦ itens a serem avaliados; ◦ grau de cobertura por itens.
Resultados	1 Plano de teste completo.

Tabela 7-3: Identificação detalhada dos requisitos da avaliação

7.3.2.2 Desenho das avaliações

Nesta atividade são definidas as Especificações de avaliações que detalham os procedimentos de avaliação. Os procedimentos de avaliação compreendem os protocolos que definem o formato das avaliações, uma caracterização dos usuários participantes e avaliadores e os roteiros de tarefas a serem executadas durante a realização da avaliação.

Cabe observar que, enquanto é feito somente um Plano para cada tipo de técnica, a Especificação é associada a uma avaliação específica, ou seja, pode ser feita uma Especificação para cada avaliação a ser realizada.

Essa atividade é realizada por meio de várias tarefas que são apresentadas detalhadamente na Tabela 7-4.

Descrição	Desenho das avaliações.
Insumos	2 Especificação de requisitos de Software (ERSw – especificação de requisitos de interface, perfil do usuário, atributos e metas de usabilidade). 3 Descrição da Avaliação de Usabilidade ("dausw"). 4 Descrição do Desenho da Interação do Software (DDISw).
Tarefas	1 Desenhar avaliações, considerando: ◦ tipo de avaliação; ◦ objetivos gerais. 2 No desenho, estabelecer: ◦ objetivos específicos; ◦ reutilização de especificações de avaliações anteriores (heurísticas, tarefas do usuário); ◦ ordem de execução se for um requisito do método de avaliação escolhido. 3 Especificar os procedimentos de avaliação, tais como roteiros (scripts), papéis e responsabilidades, número de usuários, participação de especialistas, dentre outros.
Resultados	1 Especificações das avaliações.

Tabela 7-4: Tarefas de desenho das avaliações

7.3.2.2.1 Especificação das avaliações

As Especificações das avaliações são seções do "dausw" que contém um detalhamento das avaliações a serem realizadas. A especificação compreende uma descrição dos procedimentos da avaliação e das responsabilidades dos atores envolvidos, na forma de uma seqüência de ações (roteiros). Esses roteiros devem ser observados e execução das avaliações para efeito de padronização e rigor experimental. Os procedimentos são configurados dependendo do tipo de avaliação.

O padrão adotado para as especificações de avaliações, independentemente do tipo escolhido, prevê a estrutura mostrada na Tabela 7-5. Os Recursos a serem utilizados e os Procedimentos de avaliação são detalhados em subseções a seguir.

Item	Descrição
Identificador da especificação da avaliação	Identificador único para esta avaliação.
Especialização do Planejamento	Pode ser usado para alterar qualquer parâmetro do Planejamento para uma avaliação específica objeto do Desenho da interação.
Recursos a serem utilizados	Alocação dos recursos humanos necessários para a realização desta avaliação.
Agenda detalhada	Agenda detalhada das avaliações.
Procedimentos da avaliação	Identificação e descrição de cada um dos procedimentos desta avaliação.
Material para execução da avaliação	Descreve o material de suporte, como roteiros, listas de conferências, etc, a ser utilizado nas avaliações.

Tabela 7-5: Especificação das avaliações

7.3.2.2.2 Recursos a serem utilizados

Essa atividade consiste principalmente em especificar, em função dos requisitos de recursos humanos determinados previamente, qual o perfil das pessoas que participarão da avaliação (planejamento, realização, análise e validação) e qual a quantidade necessária por perfil.

Para os testes com usuários (experimentais), deve-se descrever o número total de participantes dos testes, o período de realização dos testes, o número de seções de teste por dia, e como os participantes serão distribuídos nestas seções, considerando-se critérios específicos em relação ao perfil, às versões do produto, aos custos e prazos.

Na avaliação heurística, devem-se especificar quantos especialistas e monitores participarão da avaliação. A definição do número de especialistas depende de uma análise de custo/benefício, no entanto, a literatura (Nielsen, 1993) sobre o assunto recomenda de três a cinco para identificar a maior parte dos problemas de usabilidade das interfaces, visto que o diagnóstico é baseado na experiência e competência do especialista no assunto.

Na inspeção por lista de conferência, os próprios programadores e analistas podem fazer a avaliação. Como os inspetores são conduzidos no exame da interface através de uma mesma lista de questões sobre a usabilidade do projeto, os resultados tendem a ser uniforme. Entre três e cinco inspetores também é suficiente para detectar grande parte dos problemas.

7.3.2.2.3 Procedimentos da avaliação

Os procedimentos da avaliação devem descrever a seqüência de passos ou roteiros a serem executados ou observados pelos monitores de avaliação, especialistas em usabilidade,

projetistas de interface e participantes de sessões de testes com usuários, dependendo do tipo de avaliação de usabilidade.

DETALHAMENTO DOS PROCEDIMENTOS DA AVALIAÇÃO HEURÍSTICA

Na avaliação heurística, além dos procedimentos a serem executados ou observados pelo monitor da avaliação, deve-se especificar também quais são os procedimentos para os especialistas em usabilidade **Tabela 7-6**

Tarefas	Descrição
Seleção de heurísticas	Escolha e adaptação das heurísticas ao contexto e tipo de produto a ser avaliado.
Elaboração de roteiros	Elaboração de roteiros para o monitor de avaliação, descrevendo os passos que devem ser seguidos e observados na condução da avaliação e elaboração de roteiros para os especialistas de usabilidade, descrevendo objetivos, aspectos a serem avaliados e como a avaliação será conduzida, em termos de cronograma, por exemplo.

Tabela 7-6: Procedimentos para avaliação heurística

DESENHO DOS PROCEDIMENTOS DOS TESTES EXPERIMENTAIS

Nos testes empíricos, devem-se especificar quais procedimentos serão executados ou observados pelo monitor de teste e equipe, e quais tarefas os usuários devem realizar em cada sessão de teste.

O padrão adotado para descrição dos procedimentos é apresentado na Tabela 7-7.

Tarefa	Descrição
Definição dos aspectos gerais	Descrição sucinta dos objetivos que orientam o desenho do teste e de que forma este se dará (observação direta, por exemplo).
Levantamento das medidas de avaliação	Descreve quais tipos de dados serão medidos na avaliação e como medi-los. Os dados podem ser de natureza quantitativa, que correspondem a resultados numéricos como medidas do desempenho do usuário ou avaliação de opiniões por questionário, ou qualitativos, i.e., resultados não numéricos, como lista de problemas ocorridos durante o uso da interface pelo usuário. No caso de testes empíricos, é importante caracterizar bem os valores a serem medidos – por exemplo, deve ficar bem claro o que será considerado um erro de usabilidade na utilização da interface pelo usuário. Este trabalho deve ser baseado na especificação de usabilidade já realizada, possivelmente após uma revisão.
Procedimentos para orientar os participantes	Recepção Como o participante deve ser recepcionado e quais atividades ele deve fazer inicialmente.
	Explicações sobre o teste Apresentação das explicações que devem ser dadas aos participantes, tais como: finalidade do teste, como será o teste, o que o usuário poderá ou não fazer.
Elaboração das listas de tarefas para os usuários participantes.	Descreve as tarefas que serão realizadas pelos participantes dos testes. Cada descrição de tarefa inclui o estado ou contexto onde é iniciada, a descrição das ações a serem realizadas, os critérios de completeza e

	sucesso e tempo máximo para execução, dentre outros. Esta lista é também chamada de Roteiro de tarefas para os usuários.
Elaboração de roteiros para os monitores da avaliação	Elaboração de roteiros para o monitor de avaliação, descrevendo os passos que devem ser seguidos e observados na condução da avaliação. Cabe observar que a lista de tarefas para os usuários participantes também é utilizada pelos monitores das avaliações.
Etapas de execução das tarefas (cenários)	Descrever os cenários nos quais os participantes estarão sendo observados. Quais são as circunstâncias, equipamentos e estados destes, documentos usados, atitudes que o monitor ou equipe de testes deverão ter.
Procedimentos pós-tarefas	O que deve ser feito após o participante terminar a execução do conjunto de tarefas elaboradas para obtenção de dados. Quais ações o monitor de teste e equipe devem realizar.

Tabela 7-7: Descrição dos procedimentos para testes experimentais

DESENHO DOS PROCEDIMENTOS DAS AVALIAÇÕES COM LISTAS DE CONFERÊNCIA

A avaliação com listas de conferência é semelhante à avaliação heurística Tabela 7-8.

Tarefa	Descrição
Definição e organização do conteúdo da lista de conferência	Definição da organização e conteúdo geral ou específico da lista de conferência a ser utilizada, ou escolha de alguma lista previamente validada.
Elaboração dos scripts	Elaboração dos roteiros constando orientações, atividades e a ordem destas, se for o caso, para os analistas ou projetistas da interface.

Tabela 7-8: Descrição dos procedimentos para listas de conferência

7.3.2.3 Implementação da Avaliação

Nesta atividade é preparado o ambiente para as avaliações, compreendendo a instalação de protótipos ou versão de avaliação do produto, a disponibilização da infra-estrutura necessária e a execução de uma avaliação piloto, se prevista no método escolhido, visando à prevenção de ocorrências de problemas que poderiam vir a comprometer a realização da avaliação posteriormente, Tabela 7-9.

Descrição	Implementação das avaliações.
Insumos	1 Plano das avaliações. 2 Especificação das avaliações. 3 Recursos para as avaliações. 4 Itens a testar. 5 Ferramentas para execução da avaliação. 6 Dados de atividades anteriores de avaliação se houver. 7 Estruturas provisórias ou permanentes de avaliações se houver.
Tarefas	1 Configurar ambiente das avaliações.

	2 Disponibilizar recursos necessários. 3 Instalar itens a testar, ferramentas e estruturas provisórias. 4 Executar uma avaliação piloto, se a técnica escolhida exigir tal atividade.
Resultados	1 Itens a avaliar instalados e configurados. 2 Ferramentas e estruturas provisórias instaladas e configuradas.

Tabela 7-9: Implementação da avaliação

7.3.2.4 Execução da Avaliação

Essa atividade consiste na realização da avaliação propriamente dita. Seguindo-se as indicações de cada técnica, coletam-se os dados e registram-se os problemas de usabilidade identificados, Tabela 7-10. O desenrolar de cada avaliação é controlado e dirigido pelos monitores de teste que devem planejar com antecedência como proceder nos casos de interrupções, retomadas e encerramento precoce da avaliação. Além disso, as normas e os procedimentos descritos no plano de avaliação devem ser observados e seguidos. Se existir documentos anexos ao plano de avaliação, eles devem ser utilizados, em conformidade com o planejamento.

Descrição	Realização das avaliações.
Insumos	1 Especificação das avaliações. 2 Recursos para as avaliações. 3 Itens a testar instalados e configurados. 4 Ferramentas e estruturas provisórias ou permanentes de avaliações instaladas e configuradas. 5 Dados de atividades anteriores de avaliações se houver.
Tarefas	1 Executar as avaliações. 2 Coletar e registrar os dados. 3 Analisar as falhas e tomar as providências adequadas: ◦ em caso de falhas da própria avaliação; ◦ em caso de defeitos de implementação dos protótipos ou produto final; ◦ em caso de defeitos de desenho dos itens.
Resultados	1 Coleção de dados das avaliações. 2 Especificações de avaliações revisadas se for o caso. 3 Solicitação de investigação e correção de defeitos se for o caso.

Tabela 7-10: Realização da avaliação

Na coleta de dados, dependendo do tipo de avaliação sendo realizado, cabe observar que diversos tipos de dados podem ser importantes e, portanto, devem ser registrados. Segundo sua natureza, os dados podem ser classificados em:

- Objetivo: representa medidas observadas, por exemplo, o desempenho do usuário enquanto usa a interface para realizar tarefas de benchmark.
- Subjetivo: representa opiniões, de usuário ou de avaliadores, sobre usabilidade da interface.
- Quantitativo: são dados e resultados numéricos, como medidas do desempenho do usuário ou avaliação de opiniões.
- Qualitativo: dados e resultados não numéricos, como lista de problemas ocorridos durante o uso da interface pelo usuário.

Normalmente, as pessoas associam avaliação objetiva com dados quantitativos e avaliação subjetiva com dados qualitativos. Porém, avaliações subjetivas podem gerar dados quantitativos (com a utilização de questionários com notas para os quesitos colocados) e avaliações objetivas podem gerar dados qualitativos (por exemplo, sugestões de melhorias a partir da observação).

7.3.2.5 Análise de dados

A análise de dados coletados durante uma avaliação de usabilidade visa à análise dos problemas levantados para priorização da solução e investimento em melhorias. Pode ser dividida em dois processos maiores, distintos pela abrangência e resultado produzido, a saber: a análise preliminar e a análise detalhada como sugerido por Rubin e outros (2008). A Tabela 7-11 sumariza a atividade de análise de dados.

Na análise preliminar, os problemas mais críticos são passados para os projetistas de interface antes da liberação final do relatório, com o objetivo de fazer as devidas melhorias antes mesmo de a avaliação ser concluída. Pode-se entregar uma versão resumida do relatório com a descrição dos problemas e soluções iniciais propostas ou fazer solicitações informais.

A análise detalhada é mais exaustiva. Além dos problemas e recomendações apresentadas na análise preliminar, atualizados se necessário, outras análises pormenorizadas e recomendações devem ser implementadas e descritas no relatório final. A duração da análise detalhada pode levar de duas a quatro semanas após a avaliação, dependendo da complexidade e tamanho dos produtos avaliados.

Independentemente dos dois tipos de análise, os passos seguidos geralmente são (Rubin et al. 2008):

- compilação e sumarização dos dados;
- análise detalhada;
- desenvolvimento de soluções;
- produção do relatório de avaliação.

É importante salientar que esses podem interagir entre si, não seguindo, portanto, uma ordem necessariamente linear.

Descrição	Análise de dados.
Insumos	1 Coleção de dados resultantes da avaliação.
Tarefas	1 Compilação e sumarização de dados. 2 Análise detalhada. 3 Desenvolvimento de soluções e recomendações. 4 Produção do relatório de avaliação.
Resultados	1 Relatório da avaliação.

Tabela 7-11: Análise dos dados da avaliação

7.3.2.5.1 Compilação e sumarização dos dados

Compilar os dados significa organizá-los de acordo com padrões pré-definidos para que a uniformização propicie maior facilidade durante a análise de resultados. É importante utilizar durante a avaliação formulários ou algum meio eletrônico padronizado que permitam uma ordenação com menor esforço do conjunto de dados levantados. No Praxis-u recomendamos a planilha “apusw”.

Durante a compilação, dados que são iguais devem ser registrados apenas uma vez, juntamente, se desejado, com o número de ocorrências.

É aconselhável organizar os dados ao final de cada sessão para se obter maior agilidade no processo e esclarecer eventuais dúvidas que com o passar do tempo podem exigir maior esforço para serem esclarecidas devido a possíveis dificuldades de se lembrar de situações específicas.

Uma vez que os dados coletados sejam organizados, sejam ao final do dia de trabalho ou quando todas as sessões de realização das avaliações forem concluídas, o passo seguinte é a classificação dos problemas encontrados segundo critérios específicos, tais como tipo de tarefa, tipo de usuário. Para a avaliação com listas de conferência essa tarefa não precisa ser executada, visto que as questões normalmente já são categorizadas.

A tarefa subsequente é a sumarização ou criação de resumos dos dados organizados previamente. O objetivo é generalizar os resultados para a população de usuários ou versões de produtos. Por exemplo, pode-se optar por fazer uma distribuição de freqüência para os tipos de problemas encontrados, tais como obstáculos e barreiras; ou um resumo específico para cada grupo de usuários. Dependendo da técnica de avaliação escolhida, essa tarefa pode ser opcional.

7.3.2.5.2 Análise detalhada

A análise de dados tem os seguintes objetivos:

- identificar a causa dos problemas levantados;
- analisar dados qualitativos e quantitativos;
- priorizar problemas.

IDENTIFICAR A CAUSA DOS PROBLEMAS LEVANTADOS

Para apontar as causas dos problemas pode ser útil a identificação da fonte e do componente, ou combinação de componentes, responsáveis. Compreendendo-se as causas dos problemas é possível propor recomendações mais precisas e soluções mais eficazes. A causa de um problema pode ser considerada também como uma heurística ou diretriz não atendida.

ANALISAR DADOS QUALITATIVOS E QUANTITATIVOS

A análise de dados qualitativos e quantitativos permite, utilizando-se inferências estatísticas a partir dos dados sumarizados, definir com mais precisão a quantidade e quais tipos de problemas de usabilidade afetam determinados grupos de usuários ou versões do produto. Além disso, auxilia na atribuição de severidade para um problema. Por exemplo, um problema que pode ocorrer para qualquer tipo de usuário do produto é, normalmente, mais grave que outro que se verifica somente para alguns tipos de usuários.

A severidade do problema pode ser determinada pela combinação de vários fatores, incluindo-se a estrutura do problema, o tipo de tarefa em que ele se manifesta ou o tipo de usuário que ele afeta. A severidade pode ser quantificada por meio da seguinte escala (Nielsen 1993), Tabela 7-12:

Severidade	Descrição
------------	-----------

0	Problema cosmético (Irritante): não é preciso corrigi-lo a menos que se tenha tempo extra no cronograma do projeto.
1	Pequeno problema de usabilidade (Moderado): baixa prioridade para corrigi-lo.
2	Grande problema de usabilidade (Severo): é importante corrigi-lo, indica alta prioridade.
3	Catástrofe de usabilidade (Inutilizado): é imperativo corrigi-lo antes de ser liberado para uso. Altíssima prioridade

Tabela 7-12: Escala de severidade de um problema

Diversos fatores podem ser considerados para se determinar a Seriedade dos problemas encontrados. A **freqüência** com que um problema pode ocorrer é influenciada por dois fatores: a porcentagem do total de usuários afetados pelo problema e a probabilidade que um usuário do grupo afetado experimentará o problema. A freqüência pode ser medida com base na seguinte escala (Rubin et al. 2008),

Tabela 7-13.

Freqüência	Descrição
0	Ocorre em 10% ou menos de utilização do produto.
1	Ocorre de 11 a 50% do tempo de utilização do produto.
2	Ocorre de 51 a 89% do tempo de utilização do produto.
3	Ocorre em 90% ou mais do tempo de utilização do produto.

Tabela 7-13: Escala para medir a freqüência de ocorrência de um problema

Além da freqüência de ocorrência, outros fatores podem ser considerados na determinação da severidade de um problema de usabilidade identificado, seguem alguns exemplos.

- **Impacto:** será fácil ou difícil para os usuários superar o problema?
- **Persistência:** é um problema que só será experimentado uma vez (usuários conseguem superá-lo uma vez que sabem sobre ele) ou será problema toda vez que for encontrado?
- **Impacto de mercado:** certos problemas de usabilidade podem ter um efeito devastador sobre a popularidade do produto, mesmo que sejam objetivamente fáceis de superar.

No artefato “apusw” do Praxis-u, sugerimos que os quatro fatores: Frequencia, Impacto, Persistência e Impacto de mercado sejam analisados e a cada um atribuído uma “nota” de gravidade em uma escala de 0 a 3 conforme sugerido acima. Com base na avaliação desses

quatro fatores, sugerimos que uma outra “nota”, também na escala de 0 a 3 conforme indicado na Tabela 7-12, seja atribuída à Estimativa de seriedade considerada globalmente.

Cabe ainda observar que a priorização dos problemas encontrados é um aspecto importante da avaliação de usabilidade e, portanto, deve refletir o julgamento de toda a equipe envolvida na avaliação. Sendo assim, sugerimos que seja usado um esquema de votação, onde cada membro da equipe de avaliação dá uma nota para a Estimativa de seriedade (cada um considerando os quatro fatores) e uma média dessas notas seja considerada como a nota final da Estimativa de seriedade a ser considerada.

PRIORIZAR PROBLEMAS

O conjunto de problemas levantados devem ser priorizados a fim de permitir que a equipe responsável pelo desenho ou projeto da interface realize as melhorias em uma ordem definida com base na seriedade ou criticidade dos problemas. Além disso, por questões de custos e prazos, pode-se definir a ordem das correções ou re-projeto a partir dos problemas prioritários e minimizar o impacto negativo sobre o produto liberado.

Os problemas podem ser priorizados considerando-se somente a severidade do problema ou a combinação desta e a probabilidade de ocorrência, ou se for o caso, o consenso estabelecido pelos especialistas ou usuários em alguma reunião de brainstorming ou JAD. Técnicas baseadas no uso de cartões ou fichas podem ser utilizadas (Constantine & Lockwood 1999) para a busca do consenso em trabalho de equipes.

7.3.2.5.3 Desenvolvimento de soluções e recomendações

Essa atividade consiste em converter as informações obtidas na análise de dados para ações e recomendações a serem executadas e seguidas, respectivamente, pela equipe responsável pelo projeto das interfaces. Para se obter bons resultados nessa atividade, é importante considerar os princípios do projeto centrado no usuário, as diretrizes ou normas para desenho de interface, bem como a forma como as pessoas lidam com a informação (processo cognitivo).

Outro aspecto a ser considerado é a interdisciplinaridade da equipe responsável por essa tarefa. Profissionais da área de comunicação, ergonomia e fatores humanos podem propor soluções a partir de perspectivas diferentes e entrar em consenso sobre as alternativas apresentadas. Uma forma mais simples é a própria equipe de usabilidade juntamente com a equipe de desenho, buscar uma boa solução.

DIRETRIZES PARA ELABORAÇÃO DE SOLUÇÕES E RECOMENDAÇÕES

- Observar princípios do projeto centrado no usuário.
- Considerar normas, padrões e diretrizes para desenho de interface de usuário.
- Procurar soluções simples e eficazes.
- Focalizar primeiramente nas soluções e recomendações que causam maior impacto sobre o produto, por exemplo, uma mudança do estilo de navegação na maioria das telas da interface.
- Definir pequenas e grandes recomendações: nas pequenas recomendações se gasta menos tempo para realizar as mudanças, sem o risco de tropeçar no planejamento. Já nas grandes recomendações, as mudanças requerem mais tempo para serem realizadas.
- Definir quais são as áreas que exigem pesquisa adicional para delimitar melhor o problema e propor soluções mais coerentes.

7.3.2.5.4 Produção do relatório de avaliação

A equipe responsável pelo desenho da interface deve ter contato direto com os resultados, soluções e recomendações propostas. Para isso é importante transcrever esses itens para um relatório. Este deve ser confeccionado após se promover discussões sobre as percepções, opiniões e verificação de possíveis falhas nas recomendações.

O relatório final deve ser completo, constando todos os tipos de problemas identificados, críticos ou não. Os objetivos e revisões também devem ser relatados. O relatório final deve suportar e iniciar mudanças, dirigir ações, prover um registro histórico, ter um papel educacional e ser um importante veículo de comunicação.

A estrutura do relatório, ou seja, as seções que o compõem são descritas em padrões de avaliação de usabilidade. Em linhas gerais, o início do relatório descreve a motivação para a avaliação e como foi preparado, o meio descreve o que aconteceu durante a avaliação e o fim relata as recomendações e sugestões propostas.

7.3.2.6 Verificação do Término

Essa atividade determina se as condições para completeza e sucesso das avaliações estão satisfeitas. Se for necessário para garantir a cobertura desejada, avaliações suplementares podem ser desenhadas, retornando-se à atividade de desenho da avaliação.

Descrição	Verificação do término da avaliação.
------------------	---

Insumos	1 Plano das avaliações. 2 Especificação da avaliação. 3 Relatório da avaliação.
Tarefas	1 Checar término normal das avaliações: ° verificar se há necessidade de mais testes; ° se não houver, registrar o término normal. 2 Checar término anormal das avaliações, documentando o ocorrido. 3 Suplementar o conjunto de avaliações, se necessário. 4 Retornar à execução das avaliações, se necessário.
Resultados	1 Relatório verificado e completado. 2 Especificação de avaliações revisadas se for o caso.

Tabela 7-14: Verificação do término da avaliação

7.3.2.7 Balanço Final

Essa atividade realiza o balanço final das avaliações de usabilidade, verificando se os requisitos esperados para a atividade foram efetivamente alcançados e registrando as conclusões e lições aprendidas. Deve-se também ser feita uma aferição da qualidade da interação proporcionada pela interface através de uma comparação dos resultados obtidos com as metas ou requisitos de usabilidade pré-definidos. Se necessário, pode ser proposto uma realização de um novo ciclo completo do fluxo de usabilidade visando à melhoria da qualidade da interface.

Descrição	Balanço final
Insumos	1 Relatório da avaliação verificado. 2 Especificação da avaliação revisada se for o caso. 3 Dados sumarizados da avaliação revisados se for o caso.
Tarefas	1 Aferir a qualidade da interação com relação aos requisitos ou metas estabelecidos. 2 Descrever o status da avaliação, registrando: ° variações; ° avaliação dos términos anormais da avaliação; ° problemas não-resolvidos. 3 Descrever o status das unidades sob avaliação. 4 Completar o relatório da avaliação. 5 Colocar os artefatos reutilizáveis sob Gestão de Configuração.
Resultados	1 Relatório de resumo das avaliações. 2 Possivelmente, componentes de avaliação reutilizáveis.

Tabela 7-15: Balanço final da avaliação

7.4 PADRÃO DE DESCRIÇÃO DE AVALIAÇÃO DE USABILIDADE

A documentação das avaliações de usabilidade consiste em dois documentos:

- Descrição das Avaliações de Usabilidade do Software ("dausw"): documenta o planejamento, executado antes da realização das avaliações, abrangendo os planos e especificações.
- Relatório das Avaliações de Usabilidade do Software ("rausw"): reúne todos os relatórios de avaliação produzidos em um projeto.

A confecção da Descrição das Avaliações de Usabilidade e dos Relatórios das Avaliações de Usabilidade deve seguir as diretrizes que são apresentadas nas próximas seções.

7.4.1 DESCRIÇÃO DAS AVALIAÇÕES DE USABILIDADE

O documento "dausw" é subdividido em duas partes, uma referente aos planos de avaliação e a outra referente às especificações das avaliações.

Cada plano de avaliação de usabilidade registra os aspectos relativos ao planejamento de uma avaliação de usabilidade utilizando-se uma técnica específica. Cada especificação da avaliação define o desenho de uma avaliação.

O documento de Descrição das Avaliações de Usabilidade do Software deverá utilizar a seguinte estrutura:

1. Planos de avaliações
 - 1.1. Plano de avaliações heurísticas
 - 1.2. Plano de testes de usabilidade
 - 1.3. Plano de avaliações com listas de conferência
2. Especificações de avaliações

7.4.2 RELATÓRIO DE AVALIAÇÕES DE USABILIDADE

Os vários Relatórios das Avaliações de Usabilidade do Software registram os dados relativos às realizações das avaliações e as recomendações desenvolvidas baseadas nos acontecimentos. A seguir é apresentada uma sugestão para a organização de relatórios:

1. Relatórios de Avaliações Heurísticas

- 1.1. Relatórios de Avaliações Heurísticas do módulo A
- 1.2. Relatórios de Avaliações Heurísticas do módulo B
- 1.3. ...
2. Relatórios de Avaliações com Listas de Conferência
 - 2.1. Relatórios de Avaliações com Listas de Conferência do módulo B
 - 2.2. Relatórios de Avaliações com Listas de Conferência do módulo C
 - 2.3. ...
3. Relatórios de Testes Empíricos
 - 3.1. Relatórios de Testes Empíricos do módulo B
 - 3.2. Relatórios de Testes Empíricos do módulo C
 - 3.3. ...

7.5 GLOSSÁRIO

Sem conteúdo

7.6 BIBLIOGRAFIA

Bastien & Scapin, links [Critérios Ergonômicos](#) e [ErgoList](#) em página de site web acessado em <http://www.labiutil.inf.ufsc.br/> em outubro 2009.

Cybis, W. Betiol, A. H. Faust, R. *Ergonomia e Usabilidade – Conhecimentos, Métodos e Aplicações*, Novatec, 2007

Constantine, L.L. & Lockwood, L.A.D. *Software for Use: a Practical Guide to the Models and Methods of Usage Centered Design*, Addison-Wesley, Reading-MA, 1999.

Hix, D.; Hartson, H. R. *Developing User Interfaces: ensuring usability through product & process*, John Wiley and Sons, 1993.

ISO/IEC 14598-2 Software Product evaluation, 1999.

ISO/IEC 9126 Information technology – software product quality- part 1: quality model 1999, (FDIS).

Nielsen, J. *Usability Engineering*. Chestnut Hill, MA, Academic Press, 1993.

PADUA, W. *Engenharia de Software: Fundamentos, Métodos e Padrões*. 2. ed. Rio de Janeiro: Editora LTC, 2003. v. 1. 604 p.

Rubin, J. Chisnell, D. Spool, J. *Handbook of Usability Testing: how to plan, design, and conduct effective tests*, John Wiley and Sons, 2nd edition, 2008.

8 DIRETRIZES DE USABILIDADE

Neste capítulo apresentamos de forma sucinta uma compilação de diretrizes para o desenvolvimento da interação. Esse tipo de diretriz em geral pode ser encontrado em livros e artigos sobre usabilidade. As diretrizes enfatizam a atividade de desenho da interação, mas não se limitam a essas.

As diretrizes que se seguem estão organizadas em três níveis, fundamentos, princípios e diretivas, de acordo com o nível de abstração que abordam. As diretrizes são de várias fontes como Constantine e Lockwood (1999) e Hix (1993).

8.1 FUNDAMENTOS DE DESENHO

Apresentamos aspectos de fundamentos relacionados ao desenho de interfaces do usuário, colocados em nível mais alto de abstração.

1. **Apoio:** o sistema deve apoiar as atividades ou tarefas reais que os usuários precisam executar, tornando-as mais fácil, simples, rápido ou mais divertido ou tornando coisas novas possíveis. O sistema deve, de alguma forma, contribuir e agregar valor às atividades realizadas pelos usuários.
2. **Acesso:** o sistema deve ser acessível sem help ou documentação para o usuário que tem conhecimento do domínio da aplicação. Ou seja, a não ser por motivo de falta de conhecimento do domínio, o usuário não deve ter dificuldades em utilizar um sistema.
3. **Progressão:** o sistema deve facilitar o avanço contínuo em conhecimento e habilidade, e acomodar mudanças progressivas à medida que o usuário ganha experiência com seu uso. O sistema deve acompanhar a evolução do usuário, atendendo às necessidades que vêm surgindo à medida que o usuário ganha domínio sobre o uso do sistema.
4. **Consistência:** consistência está associada ao uso de padrões. Diversos tipos de consistência devem ser observados no desenho da interação. Conforme mencionado no texto referente a Guia de Estilos, consistência é sempre importante para propiciar a boa qualidade da interação.

5. **Contexto:** o sistema deve ser adequado para as condições reais e presentes no contexto operacional onde o sistema é utilizado. O desenho da interface do sistema deve levar em conta o ambiente físico, social e cultural em que vai ser utilizado.
6. **Eficiência:** o sistema não deve interferir (ou impedir) no uso eficiente por usuários habilidosos e com experiência no sistema. Ou seja, o sistema deve prover mecanismos como atalhos ou comandos mais poderosos que permitam maior desempenho aos usuários experientes.

8.2 PRINCÍPIOS

No segundo nível de abstração estão os princípios de desenho de interface do usuário. Vários deles estão relacionados aos princípios básicos de *design* de modo geral, já apresentados no capítulo 2 deste documento.

1. **Informação:** a disponibilização adequada de informação na interface reduz a necessidade de memorização do usuário. Ao invés de ter que memorizar as diversas opções disponíveis, o usuário necessita apenas escolher dentre as opções que lhes são oferecidas. Como mencionado na seção 2.2, a colocação adequada de informação na interface reduz a necessidade de memorização do usuário. Figuras e ícones quando significativas podem levar muita informação aos usuários de maneira direta.



Figura 8-1 Informação sobre funções disponíveis.

2. **Estrutura:** organize a interface de modo que seja útil e faça sentido, consistente com modelos mentais dos usuários, colocando coisas relacionadas juntas e separando coisas não relacionadas, diferenciando coisas diferentes e fazendo coisas semelhantes lembrar uma às outras. A estrutura de uma interface deve fazer sentido para os usuários, deve

estar consistente com seus modelos mentais, não necessariamente vai refletir a visão dos desenvolvedores.

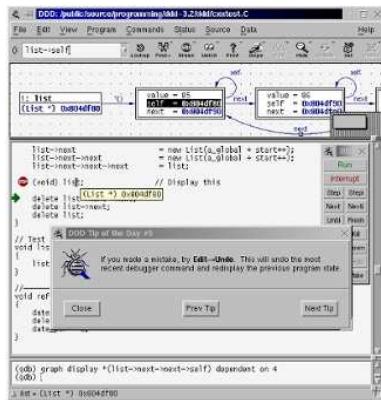


Figura 8-2 Imagem de uma estrutura confusa para o usuário.

3. **Visibilidade:** mantenha visíveis as opções necessárias para a realização de uma tarefa, sem distrair o usuário com informação redundante e fora de contexto. As ferramentas e materiais devem estar disponíveis onde e quando forem necessárias, todas as opções devem ser explícitas e aparentes ao usuário.



Figura 8-3 Permita que usuário identifique facilmente as ferramentas disponíveis.

4. **Realimentação (Feedback):** mantenha os usuários informados de ações e interpretações, mudanças de estado ou condição, erros ou exceções que sejam de seu interesse , através de uma linguagem clara, concisa, não ambígua e familiar ao usuário. O estado do sistema é uma informação importante para o usuário se situar na interação. Em muitas situações, o feedback sutil, eficiente mas sem chamar a atenção, não intrusivo, é recomendado. Um bom exemplo disso é um cursor de mouse que assume diferentes formas dependendo do elemento da interface que está sendo apontado.

5. Modelo mental: garanta que o usuário forme um modelo mental compatível com o modelo real. Isso pode ser conseguido por um desenho adequado da interface ou até mesmo por meio de treinamentos.



Figura 8-5 Modelos Mentais são utilizados pelas pessoas para compreender o mundo.

6. Simplicidade: faça que tarefas simples e comuns possam ser realizadas de forma simples, comunicando de forma clara e objetiva na linguagem do usuário.

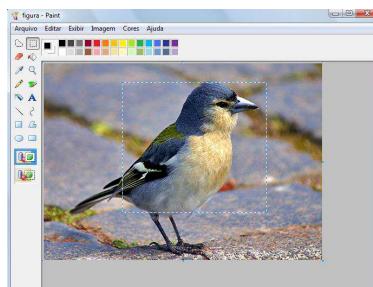


Figura 8-6 Ferramenta de seleção do PAINT, simples, mas eficiente.

7. Tolerância: seja flexível e tolerante, reduzindo os custos de erros e mau uso através da disponibilização de “desfazer” (undos) e “refazer” (redos), prevenindo erros, tolerando uma variedade de entradas e seqüências e interpretando todas ações razoáveis de uma maneira razoável. O sistema deve ser flexível, para, quando possível, buscar a interação com o usuário de forma razoável.

Software should at least not do something stupid when confronted with unexpected input or actions.

Constantine and Lockwood



Figura 8-7 Exemplos de Tolerância não adequada.

8. **Reuso:** reuse objetos e comportamentos internos e externos, mantendo consistência ao invés de fazê-la arbitrariamente, desse modo reduzindo a necessidade do usuário pensar e ter que se lembrar. A reutilização ajuda não só os usuários como também os programadores, porém a reutilização deve ser racional e utilizar consistências soluções bem elaboradas como fonte do reuso.

CTRL + C CTRL + V



Figura 8-8 Um bom exemplo de reutilização; é mantido em diversas versões de programas.

8.3 DIRETRIZES

Em terceiro nível de abstração, seguem algumas diretrizes de usabilidade.

1. Cuidado com diretrizes:

As diretrizes podem até ser conflitantes, por exemplo, uma diretriz pode indicar que se deve dar ao controle ao usuário, deixá-lo fazer o que desejar. Isso pode ir de encontro a outra diretriz que prescreve que se deve prevenir erros dos usuários. Nesses casos, avaliando a situação específica, com bom senso, pode-se decidir sobre qual diretriz seguir. Seguem dois exemplos dessa situação.

Apresentação e remoção de mensagens na tela: deve-se manter sob o controle do usuário ou fazer a mensagem desaparecer após um tempo (em nome da celeridade ou facilidade para o usuário)? Se não houver risco de dano, pode-se utilizar remoção automática, mas com tempo de exibição apropriado. Havendo risco de danos, deve-se requerer ação do usuário para remoção da mensagem.

Exemplo de caixa de mensagens com os botões “Help”, “Cancel” e “OK”. Qual seria a opção default, aquela que o sistema utilizaria em resposta a um “Enter”, por exemplo? Poderia ser o mais utilizado, ex. “OK”, ou o mais seguro, ex. “Cancel”.



Figura 8-10 Exemplo onde o botão “Sim” é tido como padrão à resposta a um ‘Enter’ do usuário.

2. Desenho centrado no usuário:

Nem sempre o que é bom para o desenvolvedor é bom para o usuário. Conheça o usuário final da aplicação. É importante conhecer seu comportamento relacionando estas características com aspectos do sistema a ser desenvolvido.



Figura 8-11 Usuários e desenvolvedores apresentam visões diferentes do sistema.

3. Envolve o usuário:

No desenvolvimento participativo, a participação do usuário desde o início é recomendável, pois o conhecimento e o comprometimento do usuário são importantes. Saiba escolher bem o usuário e identificar juntamente com ele suas necessidades.



Figura 8-12 Muitas organizações já adotam a idéia do desenvolvimento participativo com o usuário.

4. Mantenha o usuário no controle da situação:

O Usuário deve poder fazer o que julgar necessário, a menos que haja uma situação de risco ou de dano, que lhe deve ser comunicado.

O sistema deve ser previsível e as mensagens devem ser consistentes. Cuidado com o 'decurso de prazo': aquelas mensagens indicando ações que são efetivadas se o usuário não se manifesta dentro de certo tempo.

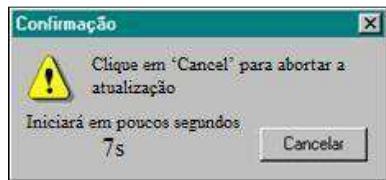


Figura 8-13 Deixe o usuário exercer controle da situação.

5. Previna erros do usuário:

Em situações de risco de dano resultante da ação do usuário é importante antecipar as suas reações evitando a geração de erros. Torne indisponíveis funções não pertinentes ao sistema e peça confirmação de ações de riscos.



Figura 8-14 Exemplo de solicitação de confirmação aos usuários.

6. Invista tempo em desenho:

Se por um lado buscamos produzir protótipos e testá-los o mais rapidamente possível, o tempo investido em melhorar um desenho representa economia de tempo em interação com o usuário. Além disso, a avaliação de um desenho mal elaborado com o usuário, de maneira intempestiva, pode levar ao descrédito em relação ao produto em desenvolvimento. A produção do protótipo ou de soluções de desenho não deve ser feita de maneira açodada.



Figura 8-15 um bom desenho requer dedicação

7. Aperfeiçoe as operações do usuário:

Permita o uso de atalhos e imponha processos otimizados via seqüências de ações aos usuários do sistema. Isso facilita e aumenta o aprendizado por parte dos usuários.



Figura 8-16 Use teclas de atalho, aumentando a eficiência por parte dos usuários.

8. Ajude o usuário a iniciar-se no sistema:

Considere as necessidades do usuário novo em relação ao sistema que ele ou ela estarão aprendendo a utilizar. Por exemplo, use demonstrações de como utilizar o sistema; isso facilita o aprendizado por parte de usuários inexperientes. Podemos também utilizar tutoriais e um modo de ajuda bem claro e consistente. Lembrem-se que tudo deve ser bem claro, pois muito formalismo pode confundir os usuários.

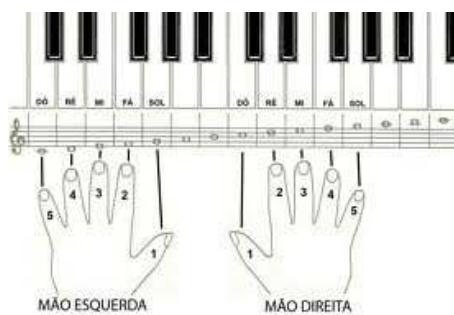


Figura 8-17 Auxilie o usuário na iniciação ao uso do sistema.

9. Considere limitação humana de memória:

Muitos sistemas apresentam inúmeras opções de funcionalidade e operações, não é interessante fazer com que o usuário saiba todas de cor. Utilize listas de opções, de modo que o sistema auxilie na sua memorização e utilização. O uso de fechamentos também facilita a controle da realização de uma tarefa pelos usuários. Fechamento é uma técnica

que consiste em se dividir uma tarefa complexa em subtarefas e indicar claramente a conclusão de cada subtarefa para que usuário se situe mais facilmente com relação ao estado do sistema.



Figura 8-18 Facilite a memorização por parte dos usuários.

10. Considere questões de cognição:

Considere questões de cognição, isto é, questões relacionadas ao conjunto dos processos mentais usados no pensamento humano, por exemplo, a memorização, a percepção, a classificação e o reconhecimento. Use pistas cognitivas, por exemplo, **ctrl + C** para copiar. Tente usar analogias com mundo real, com cuidado porque pessoas diferentes podem entender de maneira diferente as analogias utilizadas.

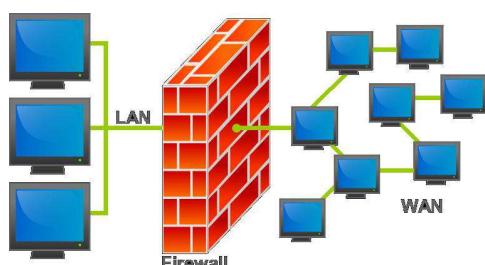


Figura 8-19 Exemplo de analogia com o mundo real, Firewall -> Proteção.

11. Use realimentação informativa:

Na interação com uma interface o usuário tem a necessidade de uma realimentação (*feedback*) constante sobre o que está sendo realizado para que este possa fazer uma avaliação de suas ações. Esse princípio foi apresentado na seção 2.7 acima. Podemos considerar dois tipos de realimentação, a articulatória e a semântica. A realimentação articulatória refere-se ao mapeamento de movimentos ou posicionamentos associados às

ações dos usuários. A realimentação semântica é aquela em que se informa ao usuário sobre os efeitos de suas ações em termos semânticos, para que este possa avaliar se seus objetivos na interação foram atingidos.



Figura 8-20 Exemplo de realimentação articulatória

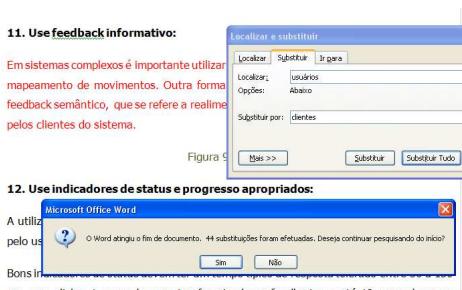


Figura 8-20 Exemplo de realimentação semântica

12. Use indicadores de progresso apropriados:

Ações simples realizadas pelo usuário na utilização de um sistema normalmente não requerem indicadores de progresso. As pessoas têm um entendimento, que é subjetivo e baseada em suas experiências, de quanto tempo será necessário para a realização de ações na interação com um sistema. Tipicamente, para ações que um usuário considere simples, o tempo de resposta tolerado pelos está em torno de:

- 50 a 150 ms para clicks,
- 1 segundo para tarefas simples e freqüentes,
- E até 12 segundos para tarefas complexas.

Esse valores foram obtidos em experimentos e refletem a expectativa dos usuários em relação ao tempo em que suas ações serão processada pelo sistema que está utilizando.

Em tarefas mais complexas, com duração de mais de 2 a 4 segundos, é interessante a utilização de indicadores de progresso. Os indicadores de progresso distraem os usuários e ajudam na redução do tempo por eles percebido, dando-lhes uma estimativa do tempo que a operação irá demorar. A partir de um indicador de progresso, o usuário forma uma expectativa de quanto falta para o término da execução de uma tarefa. Sendo assim, é importante que o indicador de progresso dê uma noção realista do tempo que ainda falta para a realização de uma tarefa para que suas expectativas não sejam frustradas.

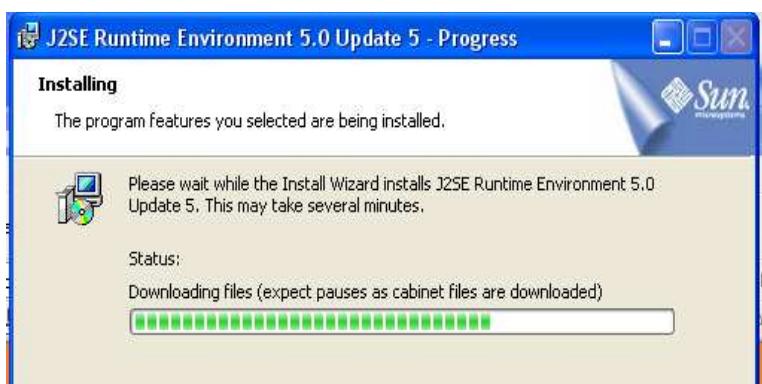


Figura 8-21 Indicador de progresso criando expectativa adequada para o usuário.

13. Use mensagens apropriadas:

Use mensagens centradas no usuário e nas tarefas e que possam ser entendidas com facilidade. Tente utilizar mensagens positivas, nunca ameaçadoras, principalmente mensagens como “erro fatal, execução abortada”.

Evite ser “espirituoso”. Claro que, dependendo do contexto e do tipo de produto, o humor pode ajudar na interação. Procure usar termos específicos apropriados: por exemplo, ao invés de “dado ilegal” use “dado fora do limite permitido de 0 a 99”.

Ponha a “culpa” de erros no sistema: por exemplo, ao invés de “comando ilegal” use “o sistema não reconhece este comando”. São coisas sutis, mas que fazem a diferença.



Figura 8-22 Mensagem de erro que não é compreendida pelos usuários.

14. Cuidado com antropomorfismo:

Antropomorfismo é a aplicação, em algum domínio, em nossa caso na interface do usuário, de linguagem ou de conceitos próprios do homem ou do seu comportamento. Dependendo do contexto, um “bom dia” ou “obrigado” pode ser interessante, mas, no fundo, todos sabem que “o computador não é gente nem amigo da gente”! Sendo assim, o antropomorfismo só deve ser usado com cautela.



Figura 8-23 Cuidado ao tentar colocar sentimentos em sistemas computacionais.

15. Use diálogos modais com cuidado:

A utilização de elementos modais em sistemas computacionais exige atenção e impedem outras ações do usuário até que alguma tarefa seja completada. Sua utilização deve ser feita quando realmente necessário, já que tira do usuário, de certa forma, o controle das ações do sistema.



Figura 8-24 Exemplo da utilização de uma tela modal.

16. Permita o usuário reverter ações facilmente:

O mecanismo de UNDO e REDO (fazer e desfazer) proporciona vários benefícios para a interação dos usuários:

- permitem a recuperação em caso de erros ou falhas causadas por uma ação indesejada executada distraidamente;
- representa comodidade, facilitando a recuperação de erros;
- provê uma ferramenta que pode ser usada para experimentação, proporcionando segurança para o usuário. Por exemplo, em uma planilha, permite ao usuário testar cenários do tipo “e se eu fizer tal coisa...”.

Claro que sempre o custo e benefício devem ser avaliados, mas sempre que viável deve-se implementar funções de *UNDO* e *REDO* nos sistemas.



Figura 8-25 Os botões undo/redo melhoram o desempenho dos usuários em suas operações.

17. Prudência ao exigir a atenção do usuário:

Lembre que o sistema deve ser claro mas cauteloso em relação a chamar a atenção dos usuários. Evite pisca-pisca e alertas sonoros; cuidado com negritos e sublinhados. Isso pode confundir o usuário e tirar seu foco na realização de suas atividades.

Evite utilizar caixas altas para chamar a atenção dos usuários, elas aumentam o tempo de leitura em até 10%. Cuidado também com o uso de cores, especialmente com seu significado em uma dada cultura e contexto.



Figura 8-26 Cores fortes e fora do contexto podem tirar o foco dos usuários.

18. Uso apropriado de telas:

A disposição de telas em um sistema computacional é muito importante. Tente manter a inércia de telas, reduzindo mudanças drásticas entre telas que são apresentadas em sequência aos usuários. Siga padrões associados a telas. Além disso, a posição de objetos usados internamente deve ser consistente entre telas.

Evite telas muito carregadas, pois a produtividade de leitura cai muito quando se usa menos de 25% de espaço branco; 50% seria recomendado para tela mais textuais.

Organize telas para gerenciar a complexidade. Observa-se ganhos de até 40% em produtividade do usuário conseguido com uma melhor organização da informação em uma tela.

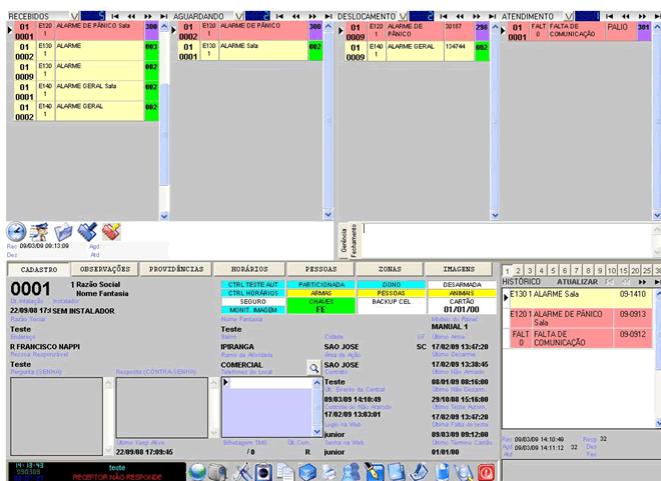


Figura 8-27 Evite utilizar varias páginas para representar dados diferentes ao mesmo tempo.

19. Considere diferenças de usuários:

Permita personalização, que deve manter-se mesmo entre sessões de uso. Sistemas computacionais podem apresentar tipos diferentes de usuários, podemos destacar, entre eles, usuários novícos, intermitentes e frequentes.



Figura 8-28 O sistema pode ser utilizado por diversos tipos de usuários.

É preciso considerar o conhecimento sintático e o conhecimento semântico necessários à realização de uma tarefa pelos usuários. O conhecimento sintático é relativo à estrutura e organização dos elementos de um interface; o conhecimento semântico é relativo ao entendimento do significado desses elementos.

Usuários noviços têm desconhecimento sintático e semântico do sistema. Eles necessitam de clareza, simplicidade, pequeno número de funções mais úteis (facilitando a memorização), mensagens claras, feedback informativo e manuais completos, tutoriais e demonstrações.

Já os usuários intermitentes possuem um conhecimento semântico, mas perdem mais facilmente o conhecimento sintático. Suas principais necessidades são em relação a comandos simples e consistentes, seqüência lógica de passos, funções fáceis de se lembrar, assistência e help on-line e manuais concisos.

Por outro lado, os usuários freqüentes têm um bom conhecimento sintático e semântico. Para um uso eficiente do sistema eles necessitem de uma interação rápida, comandos poderosos, digitação reduzida, mensagens breves com acesso a detalhes só se necessário, feedback conciso e uma possibilidade de personalização da interface.

8.4 GLOSSÁRIO

Sem conteúdo

8.5 BIBLIOGRAFIA

Hix, D.; Hartson, H. R. *Developing User Interfaces: ensuring usability through product & process*, John Wiley and Sons, 1993.

Constantine, L.L. & Lockwood, L.A.D. *Software for Use: a Practical Guide to the Models and Methods of Usage Centered Design*, Addison-Wesley, Reading-MA, 1999.

9 ELEMENTOS DE INTERAÇÃO

Neste capítulo apresentamos de forma geral e sucinta elementos de interação usados em interfaces do usuário. Podemos definir elementos de interação como coleções de objetos de interface, linguagens de comando e técnicas associadas, os quais podem ser utilizados para desenhar os componentes de interação de uma interface de um software.

As diversas formas de interação podem ser classificadas de acordo com o tipo de elementos de linguagem utilizados, como de manipulação direta (com objetos) e linguagens de comando. A maioria dos estilos de interação discutidos neste capítulo é usada em interfaces de manipulação direta. Neste tipo de interface, o usuário executa as ações desejadas através de interação imediata com os objetos de interação, ao invés de descrever as ações a serem executadas, indiretamente. O objetivo das interfaces com manipulação direta é minimizar os possíveis erros do usuário e aumentar o sentimento de primeira pessoa (Hutchins et al., 1986). Já na interação por meio de linguagens de comandos, como o nome indica, a interface disponibiliza comandos, geralmente apresentados em forma textual, que são utilizados pelos usuários para comandar a execução de funções do sistema.

Na literatura podem ser encontradas descrições e análises de diversos tipos de elementos de interação. Os elementos de interação aqui apresentados são baseados principalmente nos trabalhos de Hartson e Hix (1993). Cada estilo de interação é apresentado independentemente de uma plataforma específica de hardware ou software. Os elementos de interação mais utilizados e disseminados são as interfaces gráficas, as interfaces gráficas em sentido amplo e as linguagens de comando. Estes elementos são apresentados nas próximas seções.

9.1 INTERFACES GRÁFICAS

Interfaces gráficas podem ser definidas como qualquer estilo de interação que proveja janelas, botões, ícones, e outros. É geralmente chamada de interface gráfica do usuário, ou GUI (Graphical User Interface), utilizada em um estilo de manipulação direta de objetos.

O principal objetivo dessas interfaces é permitir a interação com a utilização de elementos gráficos como ícones e outros indicadores visuais. Essa interação é feita geralmente através de um *mouse* ou um teclado, com os quais o usuário é capaz de selecionar símbolos e manipulá-los de forma a obter algum resultado prático.

Assim como existem vários tipos de pessoas e gostos, existem também interfaces gráficas que vão ser mais ou menos apreciadas de acordo com o gosto pessoal. No entanto, a decisão sobre qual tipo de elemento de interação a ser utilizado vai depender muito das características do tipo de tarefa a ser executada. Cada interface gráfica possui um modo particular de utilização e de gerência de sua aparência; as principais formas de interfaces gráficas podem ser sub-divididas em:

- Janelas (windows)
- Menus
- Formulários (Forms)
- Caixas (Box)

9.1.1 JANELAS (WINDOWS)

As janelas (windows) podem ser considerados objetos de tela que fornecem uma arena para apresentação e interação com outros objetos de interação. Basicamente elas podem ser classificadas como janelas primárias e janelas secundárias. A janela primária é a janela inicial do sistema; é dela que podemos acessar ou gerar as demais janelas. Uma janela secundária é originada da janela primária ou de outras janelas secundárias.

Caixas de diálogo ou até mesmo caixas de mensagens do sistema podem ser consideradas como janelas secundárias. Quando várias janelas são abertas na tela simultaneamente, apenas uma fica ativa, isto é, pode aceitar entradas do usuário. Esta ativação geralmente é feita colocando-se o ponteiro do mouse em qualquer posição na janela.

As janelas podem ser movidas horizontalmente e verticalmente pela área de trabalho, de acordo com o gosto do usuário. Outra propriedade importante é o tamanho, que pode ser redimensionado horizontalmente, verticalmente ou em ambos os sentidos.

Outra propriedade das janelas é a modalidade. Uma janela modal requer foco do usuário enquanto estiver aberta, o utilizador deve necessariamente interagir com ela a fim de fechá-la, para então voltar a usar o sistema. Um exemplo é um editor de texto, que geralmente abre uma janela modal para confirmar o salvamento de um arquivo aberto. Enquanto a confirmação não é respondida, a janela do documento aberto não pode ser fechada.

Para a utilização de janelas alguns cuidados devem ser tomados. Não é conveniente o uso de muitas janelas, isso pode confundir o usuário. A organização e a ordem das janelas devem ser importantes, faça com que a seqüência seja feita de forma lógica facilitando os usuários em suas tarefas. Todas as janelas devem seguir um padrão, como tamanho, cores, títulos entre outras características. Evite utilizar a mesma janela para atividades diferentes, como por exemplo, a mesma janela lista os funcionários e seus relatórios de venda. Faça a distinção de cada janela em referência a sua funcionalidade específica.

9.1.2 MENUS (MENU)

Menus podem ser considerados lista de itens na qual uma ou mais seleções podem ser feitas pelo usuário. É um dos mais populares estilos de interação, pois reduz a necessidade de memorizações pelos usuários. O usuário simplesmente seleciona itens de uma lista diretamente, sem necessidade de memorização desses itens, reduzindo o número de erros possíveis. Entretanto, usuários experientes geralmente acham lenta a operação de ativação/desativação de menus. É preciso também considerar o espaço na tela requerido pelos menus.

A função básica de qualquer menu é oferecer escolhas ao usuário. Existem várias formas de menus, como por exemplo, menus push-buttons, menus pull-down, menus radio-button, menus check-button, menus dinâmicos, menus pop-up, menu de opção (Combo-box), menus de alternativa (toggle menu), menus de cascata, menus em “pizza” (pie menu), menu de paleta (menu de ícones) e menu embutido.

9.1.2.1 Menus push-buttons

Em um menu push-button, as escolhas são distribuídas sobre botões separados e os botões são visíveis a todo tempo em uma determinada janela. Alguns dos comandos mais comuns push-buttons que aparecem em muitas interfaces são 'ok', 'quit', 'cancel', 'help' e comandos básicos específicos da aplicação. Geralmente um dos botões é escolhido como default; sua aparência é diferente dos outros botões: contorno em negrito ou uma borda extra em volta do botão. Os push-buttons geralmente são ativados via mouse ou teclas de função ou combinação de teclas, sendo destacados quando escolhidos pelo usuário.

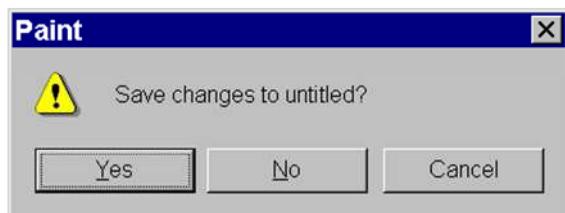


Figura 9-1 Menu push-button

9.1.2.2 Menus pull-down

É o estilo de interação mais popular. Geralmente são ativados no topo da janela; somente o título do menu ocupa espaço permanente na tela. São utilizados para se acessar as funções mais importantes do sistema.



Figura 9-2 Menu pull-down

9.1.2.3 Menus radio-button

Os menus raio-button oferecem escolhas que são exaustivas e mutuamente exclusivas.

Na figura abaixo, as opções "Fonte das Listas" e "Função de Cópia" são menus radio-button.



Figura 9-3 Menu radio-button

9.1.2.4 Menus check-button

Esses menus oferecem escolhas que podem não ser mutuamente exclusivas. O usuário pode fazer uma ou mais escolhas de um conjunto, via mouse. As opções escolhidas geralmente são marcadas por um indicador visual.



Figura 9-4 Menu check-button

9.1.2.5 Menus dinâmicos

Contêm opções que são dependentes da execução. Um exemplo simples consiste em menus que podem ter opções meio apagadas para indicar que não estão disponíveis naquele momento. Outro exemplo seria um menu de possíveis vôos entre duas cidades, que só pode ser determinado quando o usuário especifica as cidades origem e destino em tempo de execução.

Tornou-se comum incluir os nomes dos últimos arquivos dos objetos recuperados no menu File, de forma que os arquivos mais recentemente utilizados possam ser reabertos com facilidade (esta seção é chamada de MRU - Most Recently Used). Esse também é um bom exemplo de menu dinâmico.

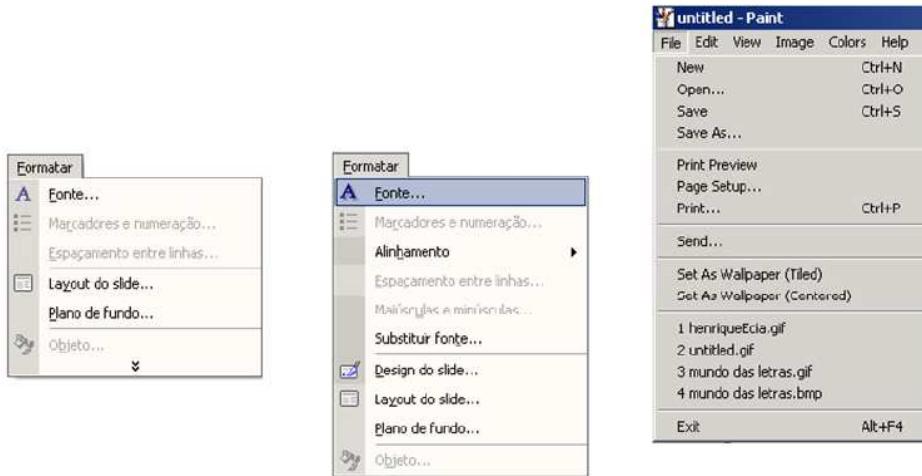


Figura 9-5 Menus dinâmicos

9.1.2.6 Menus pop-up

Os menus Pop-up podem aparecer em diferentes lugares na tela, determinado pela posição corrente do cursor, quando o usuário clica um botão específico do mouse. Geralmente não há indicação de existência do menu pop-up. Sua principal vantagem é em relação à utilização de espaço, pois não utiliza espaço permanente da tela. Por outro lado, este tipo de menu apresenta a limitação de baixa visibilidade, o que pode levar ao desconhecimento de sua existência, principalmente por usuários noviços.

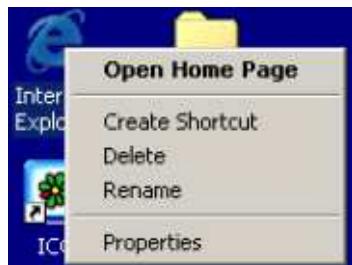


Figura 9-6 Menu pop-up

9.1.2.7 Menu de opção (Combo-box)

A origem do nome Combo é que este tipo de menu pode ser considerado uma combinação de menus de botão com menu pop-up, agregando, também, suas vantagens de visibilidade aliado à pouca ocupação de espaço de tela.



Figura 9-7 Menu de opção (combo)

O menu de opção se parece muito com um campo (em um formulário), sendo a descrição da opção escolhida é sempre visível. Outras opções aparecem quando o usuário ativa o menu. Depois de feita a escolha, o menu desaparece e o valor da nova opção escolhida passa a ser visível na janela.

O menu de opções funciona como os menus radio-button, devido a possibilidade de fazer apenas uma escolha na lista. Apesar de serem funcionalmente semelhantes, deve-se utilizar o menus de opções se o número opções for grande, se variar consideravelmente, ou se o layout de uma caixa de diálogo o exigir.

9.1.2.8 Menus de alternativas (toggle menu)

Nos menus de alternativas ou *toggle* o valor da opção corrente também é mostrado na janela, porém, as possíveis escolhas são apresentadas uma de cada vez a cada acionamento do usuário. Um exemplo típico deste tipo de menu é o utilizado em TVs que têm um botão no controle remoto utilizado para selecionar o sinal de entrada (que pode ser, por exemplo, antena, DVD, jogo, etc). A cada toque do usuário uma nova entrada é selecionada, todas consideradas em uma sequência circular.



Figura 9-8 Menus de alternativas

O menu de alternativas é mais utilizado para um pequeno número de opções, por exemplo, para escolhas binárias do tipo on/off. Como nos menus pop-up, o menu toggle é indicado quando o espaço na tela é limitado. A desvantagem, comparando com o menu de opções, é que as escolhas não podem ser todas visíveis simultaneamente.

9.1.2.9 Menus de cascata

Os menus em cascata, também conhecidos como menus hierárquicos ou menus pull-right, se parecem com uma seqüência de menus pull-down. Quando o usuário seleciona uma das opções do primeiro nível de menu na sequência, um outro menu aparece a partir da opção selecionada, e assim por diante até o final da sequência. As opções em um menu que levam a outro nível de menus podem ter um indicador visual (por exemplo, ">") para indicar que um outro menu irá aparecer.

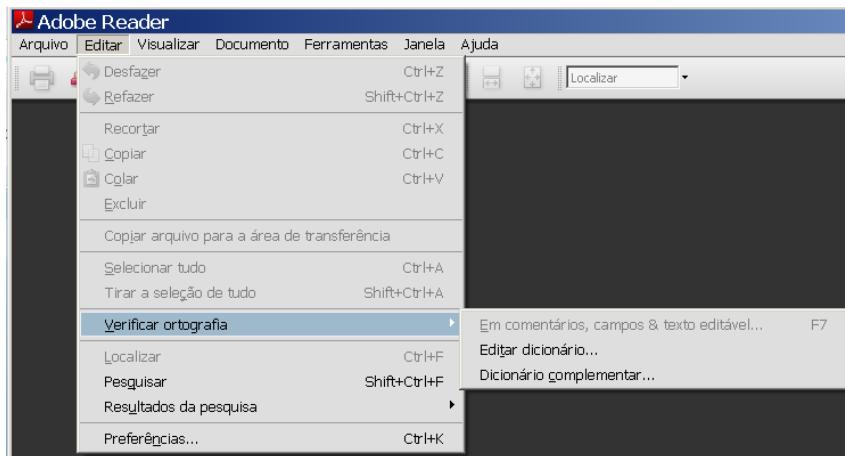


Figura 9-9 Menu de cascata

9.1.2.10 Menus em “pizza” (pie menu)

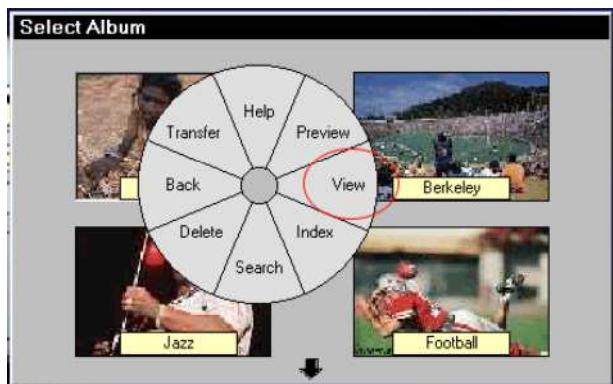


Figura 9-10 Menu em pizza

Os menus tipo pizza *pie* apresentam as opções arranjadas num círculo ou semi-círculo, procurando minimizar os movimentos do mouse feitos pelo usuário. Após um tempo de uso, a seleção nesse tipo de menu tende a se tornar mais rápida, e pode ser feita sem muita atenção visual.

9.1.2.11 Menu de paleta (menu de ícones)

Os menus de paleta ou menus icônicos são aqueles nos quais as opções são representadas por ícones gráficos que podem conter desenho e/ou texto. As opções geralmente são mutuamente exclusivas e dispostas no lado esquerdo da janela. São, em essência, push-buttons agrupados juntos; as escolhas geralmente são mutuamente exclusivas e podem implicar mudanças de modo. Muito usados em editores gráficos.

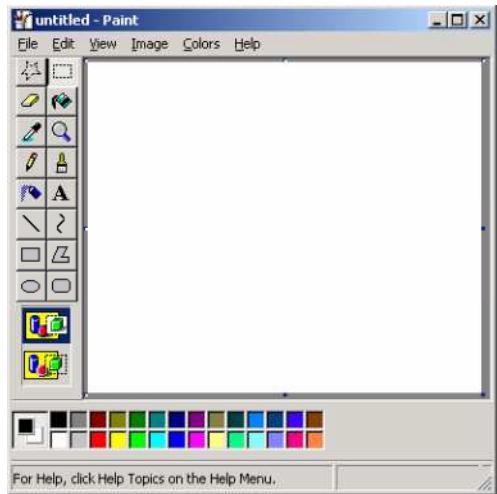


Figura 9-11 Menu de Paleta

9.1.2.12 Menu embutido

Menus embutidos são encontrados em hipertextos ou hipermídia. Numa tela com texto e/ou gráficos, alguns objetos são selecionáveis via mouse ou touchscreen. Depois de feita a seleção do objeto em destaque, esse objeto é instanciado e o usuário interagir com ele. São indicados em sistemas de armazenamento e recuperação de informação online, onde o usuário pode selecionar um objeto numa tela cheia de informação para encontrar maiores detalhes sobre o objeto selecionado.

9.1.3 FORMULÁRIOS (FORMS)

Um formulário é análogo aos formulários de papel a que estamos acostumados e consiste em uma tela contendo campos rotulados a serem preenchidos pelo usuário, geralmente através de linha de comando ou fazendo escolhas em menus. Um formulário é uma janela secundária que geralmente possui informações associadas a uma base de dados.

Figura 9-12 Formulário

É interessante que os formulários contenham um visual atraente e conteúdo consistente. Evite assumir que os formulários de papel possam ser convertidos diretamente para o desenho na tela, pois a tela tem dimensões diferentes e outras características a mais.

Use indicadores apropriados para campos no formulário, como indicadores visuais, e organize os formulários em seções, como por exemplo, campos opcionais, campos obrigatórios, entre outros.

Aproveite e utilize de maneira consciente abreviações, como por exemplo, CPF, CEP. Use uma navegação lógica entre os campos. Use mensagens de erros informativas e consistentes para caracteres e valores inválidos juntamente com mensagens que auxiliem o preenchimento de campos, evitando erros por dúvidas dos usuários.

Existem vários tipos de valores para os campos de um formulário, como por exemplo, os valores digitados; lista de escolhas; valores default; valores obrigatórios e valores opcionais; e valores dependentes.

9.1.3.1 Valores digitados

Os valores digitados pelo usuário podem ser validados ou não validados. Num campo não validado o usuário pode digitar qualquer cadeia de caracteres, que será aceita pelo sistema. Num campo validado, o usuário deve digitar a cadeia numa sintaxe pré-definida, por

exemplo, data e hora. Se o usuário não seguir o formato prescrito, a cadeia não é aceita pelo sistema e o usuário deve tentar novamente.

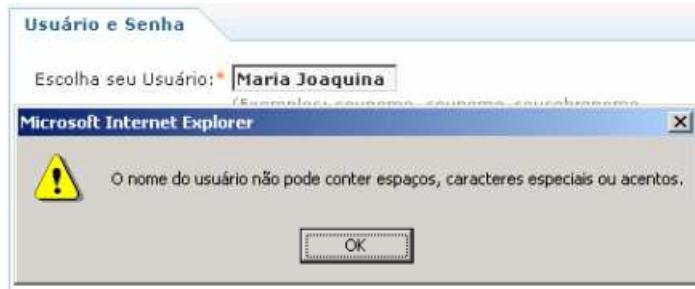


Figura 9-13 Valor digitado

9.1.3.2 Lista de escolhas

Nas listas de escolhas, todas as opções permitidas são apresentadas ao usuário, geralmente através de um menu toggle ou menu de opções. Os possíveis erros de digitação são reduzidos e o usuário não precisa recordar-se de todas as opções.

A screenshot of a form with several input fields. On the left, there are radio buttons for 'Sexo' with options 'Masculino' and 'Feminino', and a red asterisk. To the right is a dropdown menu labeled 'Estado'. Below these, there are radio buttons for 'Estado Civil' with options 'Solteiro', 'Casado', 'Viúvo', and 'Separado', also with a red asterisk.

Figura 9-14 Lista de escolhas

9.1.3.3 Valores default

Alguns campos possuem valores default, tal como data corrente, que pode ser facilmente obtida do sistema. O usuário pode alterar este valor, se o desejar.

A screenshot of a form with a dropdown menu labeled 'País'. The selected value is 'Brasil', indicated by a blue background.

Figura 9-15 Valores default

9.1.3.4 Valores obrigatórios e valores opcionais

Campos com valores opcionais, ao contrário dos obrigatórios, podem permanecer vazios. Em um formulário, campos com valores obrigatórios devem ser distinguidos dos campos com valores opcionais; se possível agrupados em diferentes posições da tela.

Os campos marcados com (*) são de preenchimento obrigatório.

Dados Pessoais	
Nome: Maria	Sobrenome: Joaquina
E-mail: maria@joaquina.ert	
Sexo: <input checked="" type="radio"/> Masculino <input type="radio"/> Feminino	
Estado Civil: <input checked="" type="radio"/> Solteiro <input type="radio"/> Casado <input type="radio"/> Viúvo <input type="radio"/> Separado	
Data de Nascimento: * <input type="text"/> / <input type="text"/> / <input type="text"/>	(Exemplo: 01/01/2001)

Figura 9-16 Valores obrigatórios e opcionais

9.1.3.5 Valores dependentes

Campos com valores dependentes são aqueles que devem ser preenchidos somente se outro valor for digitado. Por exemplo, se o campo estado civil foi preenchido com a opção casado, o campo com o nome do cônjuge deve ser completado. O sistema pode forçar esta dependência entre campos automaticamente.

9.1.4 CAIXAS (BOX)

Uma caixa consiste em uma área da tela usada para mensagens, entrada de texto, comandos, seleção e controle. Podem ser vistas como uma janela que não possui as opções de minimizar/maximizar e redimensionar. Alguns tipos de caixas aparecem como resultado de ações dos usuários, enquanto outras são apresentadas pelo sistema para informar alguma situação corrente.

As caixas suportam agrupamento visual de diferentes tipos de objetos como botões, listas e *scroll-bars*. Se não forem cuidadosamente projetadas, as caixas (principalmente as caixas de diálogo) podem parecer bastante confusas ao usuário. Existem vários tipos de caixas, como por exemplo, as caixas de listas, as caixas de entrada, as caixas de mensagem e as caixas de diálogo.

9.1.4.1 Caixas de lista

Uma Caixa de lista consiste em uma sequência de opções, geralmente com *scroll-bars* horizontal e vertical. Podem ter um pequeno campo para entrada de texto, geralmente na base da caixa, onde o usuário digita uma cadeia de caracteres, o sistema percorre as opções procurando por aquela que casa com a cadeia digitada. Sua utilização evita erros por parte do usuário, e não requer que o usuário se lembre de todas as opções.

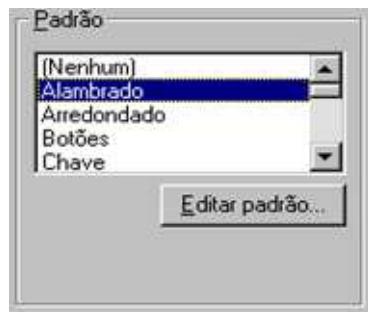


Figura 9-17 Caixa de lista

9.1.4.2 Caixas de entrada

Uma Caixa de entrada permite ao usuário entrada de texto, geralmente possui funções básicas de edição (como inserir, apagar e quebrar linhas (*wraparound*)).



Figura 9-18 Caixa de entrada

9.1.4.3 Caixas de mensagem

Uma Caixa de mensagens é um objeto de interação que apresenta informação ao usuário, mostrando o progresso de uma operação, fazendo perguntas, dando algum aviso ou requerendo algum tipo de informação. Geralmente, a caixa de mensagens é apresentada pela aplicação sem que o usuário tenha requerido diretamente, muito utilizada para apresentar mensagens de saída (ex.: mensagem de erro, mensagens de confirmação) para o usuário.



Figura 9-19 Caixa de mensagem

9.1.4.4 Caixas de diálogo

Uma Caixa de diálogo é um objeto de interação que contém outros objetos, como: listas, botões, caixas e campos para entrada de texto. Geralmente é apresentada como parte de uma tarefa, como resposta a uma escolha em um menu ou ação de uma tecla aceleradora. Caixas de diálogo desaparecem como resultado de uma ação explícita do usuário, como pressionar o botão ok ou cancel dentro da própria caixa. Elas podem ser classificadas como caixas de diálogo não-modais ou caixas de diálogo modal.

CAIXAS DE DIÁLOGO NÃO-MODAIS

Este tipo de caixa meramente informa ou aguarda entradas. Uma barra de ferramentas do Windows é um exemplo de uma caixa de diálogo não-modal. Ela apenas fica na tela e responderá se for clicada. Outro exemplo de caixa de diálogo típica não-modal é do corretor ortográfico exibido pelo Word for Windows, que permite ao usuário continuar a editar o documento enquanto a caixa de diálogo permanece exibida na tela.



Figura 9-20 Caixa de diálogo não modal

CAIXAS DE DIÁLOGO MODAL

Algumas vezes o aplicativo não pode continuar sem o auxílio do usuário. Por exemplo, um erro durante a impressão como uma mensagem do tipo “acabou o papel” não deve ser do tipo que o aplicativo ignore com facilidade. Em outro exemplo, alguns aplicativos podem não permitir a edição de um documento enquanto o aplicativo está imprimindo. Isto é chamado caixa de diálogo modal, pois não se pode fazer nada enquanto a caixa está sendo exibida, mas, no entanto, é possível passar para um outro aplicativo (Ctrl-Esc ou Alt-Tab no Windows).

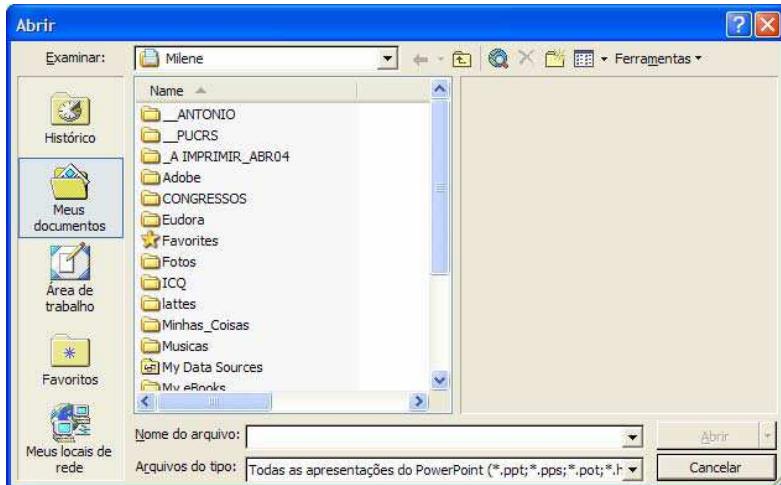


Figura 9-21 Caixa de diálogo modal

9.2 INTERFACES GRÁFICAS EM SENTIDO AMPLO

Neste capítulo usamos o termo interface gráfica em um sentido mais amplo, vinculada ao uso de representação visual da informação em contraste com o uso de textos ou números - mais amplo que o conceito WIMP (Windows, Icons, Menus and Pointers) ou interfaces VUI (Visual User Interfaces).

As interfaces gráficas usam manipulação direta, interação point-and-click, seguindo o paradigma objeto-ação: aponta e clica um objeto para selecioná-lo, depois executa uma ação sobre ele.

As interfaces gráficas fazem também uso de representações visuais, ao invés de simples representações alfanuméricas, para a comunicação com o usuário. Entretanto, alguns tipos são mais difíceis de serem projetadas e implementadas do que as interfaces alfanuméricas, e o hardware para elas também pode ser bem mais caro. A seguir apresentamos algumas interfaces gráficas muito utilizadas atualmente, são elas: a visualização de dados, bancos de dados visuais, animações, vídeo (e áudio) e realidade virtual.

9.2.1 VISUALIZAÇÃO DE DADOS

A Visualização de dados representa uma das primeiras aplicações de gráficos na interação com o usuário, incluindo grafos, histogramas, e vários tipos de imagens mais bem elaboradas. Estas técnicas de visualização têm sido largamente utilizadas nas mais diversas áreas, por exemplo, Química, Medicina, Geografia, Geologia, Astronomia. É usado muito em

diagnósticos médicos usando ultra-som, simulação de fluxo de fluido, transferência de calor, entre outros.

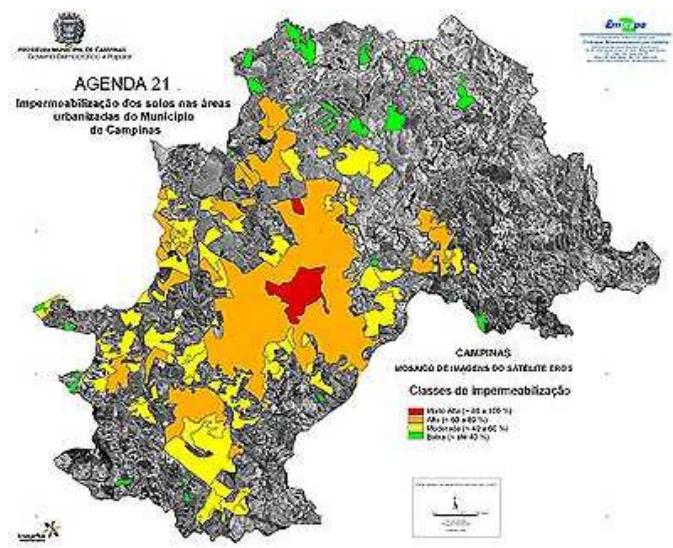


Figura 9-22 Visualização de dados

9.2.2 BANCOS DE DADOS VISUAIS

Os Bancos de Dados visuais consistem de bancos de dados com técnicas de navegação através de representações visuais como imagens digitalizadas (por exemplo, manuscritos num museu) ou imagens geradas pela aplicação (por exemplo, um poço de petróleo e a geologia da vizinhança).

As técnicas para navegação através de uma representação visual para banco de dados estão se tornando mais populares, principalmente em relevo de terreno, documentos históricos, entre outros. As imagens podem também ser geradas a partir de bancos de dados com informação numérica, como por exemplo, imagens geológicas utilizadas em exploração de petróleo.

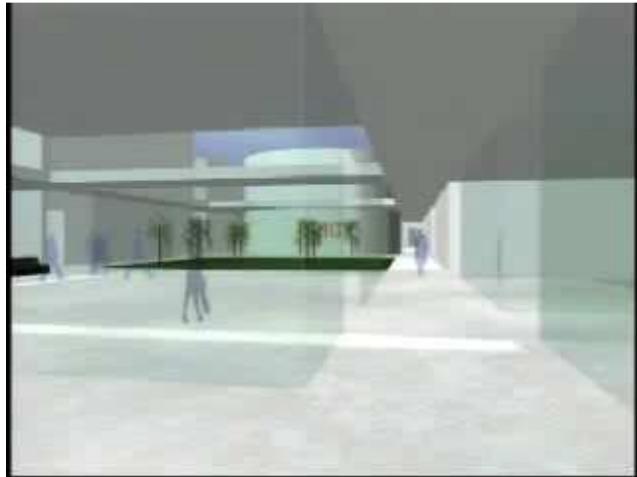


Figura 9-23 Banco de dados virtuais

9.2.3 ANIMAÇÕES

A Animação visa a criação de imagens em movimento em uma interface. Uma Animação pode ser, por exemplo, a representação de saída de uma simulação à medida que ela muda no tempo. São muito utilizadas em aplicações militares para treinamento em áreas perigosas e inacessíveis. Pode incluir, além de imagens visuais, sons, vibrações e temperaturas de um ambiente real. Muito utilizada também em jogos. Um exemplo pode ser visto em:

http://pt.wikipedia.org/wiki/Ficheiro:Newtons_cradle_animation_book.gif

9.2.4 VÍDEO (E ÁUDIO)

O Vídeo como animação realística, e o áudio que tipicamente o acompanha, pode ser combinado com outros elementos de interação em uma interface - por exemplo, em uma janela.

O vídeo ilustra de forma mais realista as atividades e, como na animação, pode ser muito bem empregado em softwares de treinamento. A produção de vídeo de alta qualidade envolve várias atividades que requerem técnicas especiais e equipamentos caros, com altos custos e nem sempre disponíveis para os projetistas da interação com o usuário.

9.2.5 REALIDADE VIRTUAL

O termo Realidade virtual é freqüentemente utilizado para se referir a qualquer simulação interativa onde geralmente são explorados vários sentidos do usuário (visão, audição, tato e até mesmo olfato). Ao invés de usar o mouse para manipular objetos na tela, os usuários podem projetar-se em qualquer ambiente tridimensional. A realidade virtual representa a experiência máxima de primeira pessoa. Através do uso de equipamentos especiais como displays montados na cabeça e luvas de dados (*data gloves*), os usuários podem entrar no mundo virtual e manipular objetos fazendo movimentos com algumas partes do corpo.



Figura 9-24 Realidade Virtual

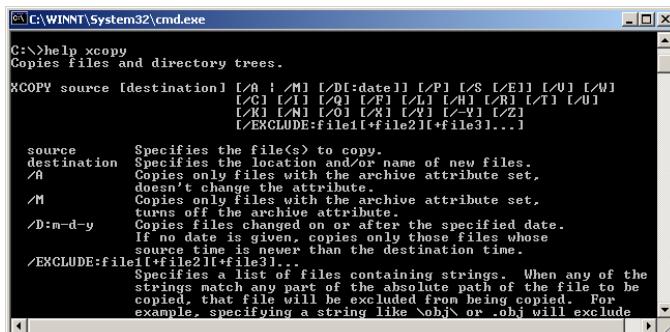
Estes sistemas podem ser extremamente caros, entretanto, avanços tecnológicos atuais prometem torná-los disponíveis brevemente.

Pioneiros na área prometem que os sistemas com realidade virtual irão mudar a maneira como as pessoas usam computadores, a maneira como aprendem e até mesmo a maneira como se relacionam umas com as outras. O objetivo principal da realidade virtual é projetar sistemas de computação que se comuniquem com pessoas de maneira mais próxima da humana, ao invés de forçar os usuários a se conformarem com as restrições de comunicação do computador.

9.3 LINGUAGEM DE COMANDOS

A Linguagem de comando constitui um dos primeiros estilos de interação encontrados em interfaces com o usuário. Consistem em cadeias alfanuméricas representando comandos, parâmetros e/ou opções que são digitadas pelo usuário. Compreendem um estilo de

comunicação rápido e poderoso que muitas vezes é preferido pelos usuários com habilidade de digitação. Entretanto, estes comandos requerem treinamento e boa capacidade de memória do usuário para lembrar a sintaxe precisamente. A linguagem de comandos pode ser mais difícil de se implementar, dependendo da análise sintática a ser feita. As linguagens de comando também podem ser utilizadas em associação com entradas de voz.



A screenshot of a Windows Command Prompt window titled 'C:\WINNT\System32\cmd.exe'. The window displays the help output for the 'xcopy' command. The text is as follows:

```
C:\>help xcopy
Copies files and directory trees.

XCOPY source [destination] [/A | /M] [/D[<date>]] [/P] [/S [/E]] [/V] [/W]
  [/C] [/I] [/Q] [/F] [/L] [/H] [/R] [/T] [/U]
  [/K] [/N] [/O] [/X] [/Y] [/Z]
  [/EXCLUDE:<file1>[+<file2>][+<file3>]...]
```

source Specifies the file(s) to copy.
destination Specifies the location and/or name of new files.
/A Copies only files with the archive attribute set,
 doesn't change the attribute.
/M Copies only files with the archive attribute set,
 turns off the archive attribute.
/D:<m-d-y> Copies files changed on or after the specified date.
 If no date is given, copies only those files whose
 source time is newer than the destination time.
/EXCLUDE:<file1>[+<file2>][+<file3>]...
 Specifies a list of files containing strings. When any of the
 strings match any part of the absolute path of the file to be
 copied, that file will be excluded from being copied. For
 example, specifying a string like \obj\ or .obj will exclude

Figura 9-25 Linguagem de comandos

Com o aparecimento das interfaces de manipulação direta, as linguagens de comando se tornaram menos comuns, mas ainda são utilizadas em várias interfaces, principalmente por profissionais da área de informática. Algumas aplicações mais antigas permitiam ao usuário mais de uma forma de comunicação, por exemplo, através da digitação de comandos curtos ou ativando menus ou teclas de função para selecionar operações.

Para que uma linguagem de comando possa ser usada de forma eficiente pelos usuários, deve-se levar em conta algumas considerações para facilitar o seu uso. Procure usar regras consistentes de formação para comandos de entrada, usando uma sintaxe consistente e escolha nomes de comandos significativos, específicos que facilitem o entendimento e a memorização. Aplique de forma consciente abreviações de palavras, permitindo uma correlação fácil em relação aos erros de digitação.

9.4 OUTROS ELEMENTOS DE INTERAÇÃO

Embora existam muitos outros estilos de interação que podem ser apresentados, serão mencionados apenas alguns daqueles que estão se tornando cada vez mais populares.

9.4.1 TELA DE TOQUE

Touchscreens estão entre os dispositivos de entrada de dados mais duráveis e robustos de todos os dispositivos de entrada. São bastante utilizados em ambientes hostis e em sistemas de acesso público, como no Epcot Center. São ideais para usuários novatos devido à simplicidade de uso.



Figura 9-26 Tela de toque

São muito úteis também quando há pouco espaço para o teclado e/ou mouse e situações em que os usuários têm que ser guiados através de seqüências complexas de tarefas. Recentemente, vêm sendo muito utilizados como tela em notebooks e dispositivos portáteis como o Iphone e similares.

9.4.2 SÍNTESE DA FALA

Os Sistemas que utilizam síntese da fala (criação de sons e palavras no computador) são ideais para usuários deficientes que não podem ver a tela ou manipular o teclado ou o mouse efetivamente. A geração de sons audíveis e palavras com o computador é uma tecnologia relativamente bem dominada, conseguindo-se atualmente que soe mais natural. Pode ser usado de forma redundante, associado a textos e imagens.

9.4.3 RECONHECIMENTO DA FALA E LINGUAGEM NATURAL

São aplicações que permitem ao usuário se expressar em linguagem natural, ou seja, utilizando a língua com que ele se comunica com outros seres humanos, seja português, inglês, entre outras.

A interação em linguagem natural é bastante atrativa para usuário com pouco ou nenhum conhecimento em computação. Entretanto, ela não se aplica a todos os tipos de sistemas. Sistemas de consulta a informações e sistemas baseados em conhecimento são exemplos onde a utilização de interfaces em linguagem natural é bastante interessante. No primeiro caso, por possibilitar que usuário não especialistas possam fazer consultas em sua própria língua. No segundo caso, para que o sistema gere explicações a partir da sua base de conhecimento, uma vez que a linguagem natural é expressiva o suficiente para a descrição do “raciocínio” artificial do programa, o que não seria possível com outros estilos de interação.

Uma aplicação que oferece interface em linguagem natural precisa lidar com construções vagas, ambíguas, e até gramaticalmente incorretas. Ainda não é possível desenvolver sistemas que compreendam qualquer expressão em linguagem natural, mas diversos tipos de sistemas especialistas utilizam com sucesso algum subconjunto de uma linguagem natural, nos quais o usuário deve se expressar de forma inequívoca e tendo em vista as frases que tais sistemas possam interpretar.

Além da fala, pode-se permitir que um usuário interaja com aplicações em linguagem natural por meio de uma interface textual onde ele deve digitar as frases que expressem seus comandos ou questionamentos. Outra alternativa são as interfaces orientadas por menus, através dos quais ele pode selecionar cada palavra ou expressão até compor a frase desejada. Muita pesquisa ainda será necessária antes do desenvolvimento de um computador que possa reconhecer e entender qualquer tipo de frase dita pelo usuário. As principais dificuldades no reconhecimento da fala estão relacionadas com as ambigüidades léxicas, sintáticas e semânticas inerentes a qualquer idioma.

9.5 GLOSSÁRIO

Animação digital é a arte de criar imagens em movimento utilizando computadores. É um subcampo da computação gráfica e da animação. São criados cada vez mais trabalhos com o uso de gráficos de computador a 3D, mas ainda se usam bastante os gráficos de computador a 2D. Por vezes, o destino da animação é o próprio computador, mas por vezes é outro meio, como filmes dedicados parapropaganda e cinema. (Wikipedia, n.d.).

9.6 BIBLIOGRAFIA

Hartson, H. R. and Hix, D., *Developing User Interfaces: ensuring usability through product and process*, John Wiley & Sons, 1983.

Hutchins, E. L., Hollan, J. D. & Norman, D. A., *Direct Manipulation Interfaces*, In Norman, D. A. & Draper, S. W., *User Centered System Design*, Lawrence Erlbaum Associates, 1986.

Wikipedia, http://pt.wikipedia.org/wiki/Animação_digital em 4/2010

10 GLOSSÁRIO

PROCESSO

Processo é um conjunto de passos parcialmente ordenados, cujo objetivo é atingir uma meta. No caso de processo de desenvolvimento de software a meta é entregar um produto de software de maneira eficiente, previsível e que atinja as necessidades de negócio (Wikipedia, n.d.).

11 BIBLIOGRAFIA

Booch, G.; Rumbaugh, J.; Jacobson, I., *Unified Modeling Language User Guide*, 2nd Edition, Addison Wesley, 2005.

Chapman, C.N. & Milham, R. The personas' new clothes. Human Factors and Ergonomics Society (HFES) 2006, San Francisco, CA. October 2006.

Cheng, L., Scapin, C. A. at al. QFD: *Planejamento da Qualidade. Escola de Engenharia da UFMG*. Fundação Cristiano Ottoni, Belo Horizonte, MG, 1995.

Constantine, L.L. & Lockwood, L.A.D. *Software for Use: a Practical Guide to the Models and Methods of Usage Centered Design*, Addison-Wesley, Reading-MA, 1999.

Cooper, A. The Inmates are Running the Asylum: Why High Tech Products Drive Us Crazy and How To Restore the Sanity. Macmillan Publishing Co., 2000.

Cooper, M. Evaluating accessibility and usability of web pages. In Proceedings of 2nd Int. Conference on Computer-Aided Design of User Interfaces, Kluwer Academic, 33-42, 1999.

Cooper, A. e Reimann, R. *About face 2.0: The essentials of interaction design*. Information Visualization. 2004.

Galitz, W. O. *The Essential Guide to User Interface Design*. John Wiley, 2007.

Grudin, J. and Pruitt, J. Personas, participatory design and product development: an infrastructure for engagement. Paper presented at Participatory Design Conference 2002, Malmo, Sweden. June 2002.

Hackos, JT & Redish, JC 1998, *User and Task Analysis for Interface Design*, John Wiley & Sons.

Hix, D.; Hartson, H. R. *Developing User Interfaces: ensuring usability through product & process*, John Wiley and Sons, 1993.

<http://webmail.faac.unesp.br/~paula/Paula/everydaythings.pdf>

ISO 9241-11 Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability 1998.

ISO/IEC 9126 Information technology – software product quality- part 1: quality model 1999, (FDIS).

ISO/IEC 14598-2 Software Product evaluation, 1999

Krug, S.; Black, R. *Don't Make Me Think: Common Sense Approach to Web Usability*, 2nd edition, New Riders, 2005

Mulder, S. and Yaar, Z. The user is Always Right: A practical Guide to Creating and Using Personas for the Web. New Riders Publishing Thousand Oaks, CA, USA, 2006.

Nielsen, J. Usability Engineering. Chestnut Hill, MA, Academic Press, 1993. Livro clássico, precursor, sobre usabilidade.

Nielsen, J. *Designing Web Usability*, New Riders Press, 2000.

Norman, D. A. The Design of Everyday Things. New York, Basic Books, reimpresso em 1998.

PADUA, W. *Engenharia de Software: Fundamentos, Métodos e Padrões*. 2. ed. Rio de Janeiro: Editora LTC, 2003. v. 1. 604 p.

Rosson, M. B., Carroll, J.M. *Usability Engineering: Scenario Development of Human-computer Interaction*. Morgan kaufmann Publishers, 2002.

Rubin, J. Chisnell, D. Spool, J. *Handbook of Usability Testing: how to plan, design, and conduct effective tests*, John Wiley and Sons, 2nd edition, 2008.

Rumbaugh, J.; Jacobson, I.; Booch, G., *The Unified Modeling Language Reference Manual*, Addison Wesley, 2nd edition, 2004.

Shneiderman, B; Plaisant, C.; Cohen, M.; Jacobs, S. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 5 ed. Reading, MA, Addison-Wesley, 2009

Van Harmelen, M. *Object Modeling and User Interface Design: Designing Interactive Systems*, Addison-Wesley, 2001.

Wikipedia, em <http://pt.wikipedia.org/wiki>, acessado em 10/2009