

3

Avaliação de Usabilidade de Sites Web

Marco Winckler¹

Marcelo Soares Pimenta²

Resumo:

Desde a invenção da Web, a tecnologia para construção de sites vem sendo progressivamente incrementada permitindo o desenvolvimento de aplicações cada vez mais complexas que utilizam a Web não apenas para troca de informações, mas como plataforma para aplicações distribuídas tais como comércio eletrônico e intranets. Durante este processo evolutivo, o número de usuários e de sites Web cresceram exponencialmente. A Web tornou-se acessível a todas as pessoas, e conta com uma grande variedade de aplicações. Contudo, observa-se que tal popularidade não implica necessariamente em satisfação dos usuários, que confrontam-se muito frequentemente com problemas de usabilidade.

Usabilidade é o termo técnico usado para descrever a qualidade de uso de uma interface. Quando a usabilidade é levada em conta durante o processo de desenvolvimento de interfaces Web, vários problemas podem ser eliminados como, por exemplo, pode-se reduzir o tempo de acesso à informação, tornar informações facilmente disponíveis aos usuários e evitar a frustração de não encontrar informações no site.

A tendência atual em avaliação é tentar identificar os problemas de usabilidade tão logo eles possam ser detectados na interface. Uma vez identificado, o problema pode ser solucionado ou, ao menos, seus efeitos podem ser minimizados. Existe uma série de métodos de avaliação que podem ser utilizados em diferentes etapas do desenvolvimento de interfaces Web.

O objetivo principal deste capítulo é descrever o processo de avaliação de usabilidade de interfaces Web e alguns métodos e ferramentas que podem ser utilizados neste processo. Para contextualizar este assunto, na parte inicial do capítulo são apresentados alguns tópicos importantes tais como a caracterização de problemas de usabilidade, as peculiaridades do ambiente Web e suas aplicações assim como uma visão panorâmica sobre concepção de interfaces Web, com a intenção de permitir uma compreensão melhor dos objetivos e procedimentos envolvidos na avaliação.

¹ e-mail: winckler@irit.fr

LIHS -IRIT (Institut de Recherche en Informatique de Toulouse)
118, route de Narbonne 31062 - Toulouse Cedex 4 - França

² e-mail: mpimenta@inf.ufrgs.br

Instituto de Informática - UFRGS
Av. Bento Gonçalves, 9500 - Caixa Postal: 15064
CEP 91501-970 - Porto Alegre - RS -Brasil

3.1 Introdução

A World Wide Web (WWW) ou simplesmente Web, foi criada por Tim Berners-Lee em 1989, nos laboratórios do CERN como uma solução para os problemas de intercâmbio de informações entre os pesquisadores. Desde então, a tecnologia para construção de interfaces Web vem sendo progressivamente incrementada permitindo o desenvolvimento de aplicações cada vez mais complexas. No início, a Web era apenas um ambiente para publicação de documentos no formato texto e HTML (*Hypertext Markup Language*) e, portanto, a interação dos usuários era limitada a ler/imprimir texto e a selecionar links para outros documentos. Na sequência, vieram os formulários e programas CGIs (*Common Gateway Interface*) que permitem a entrada de dados do usuário e a integração com aplicações de banco de dados. Como consequência desta inovação, surgem aplicações complexas que utilizam a Web não apenas para troca de informações, mas como plataforma para aplicações distribuídas como, por exemplo, o comércio eletrônico e intranets (em que empresas usam a Web para gerenciar informações internas). Atualmente, outras tecnologias de interface como por exemplo, ActiveX, Java e JavaScript, entre outras, permitem um tipo de interação do usuário próximo ao encontrado nas tradicionais interfaces WIMP (*Windows, Icons, Mouse and Pointers*, como MS Windows, por exemplo).

Durante este processo evolutivo, o número de usuários e de sites Web cresceram exponencialmente. A Web tornou-se acessível a todas as pessoas, e conta com uma grande variedade de aplicações. Contudo, observa-se que tal popularidade não implica necessariamente em usuários satisfeitos. Muitos sites Web são visitados uma única vez pelos usuários. Em muitos casos, isso ocorre não porque o conteúdo do site não os interessa mas sim porque eles foram incapazes de encontrar a informação desejada. A dificuldade em encontrar informações no site é, sem dúvida, o problema de usabilidade mais relatado por usuários.

Usabilidade é o termo técnico usado para descrever a qualidade de uso de uma interface (Bevan, 1995). Essa é uma qualidade importante pois interfaces com usabilidade aumentam a produtividade dos usuários, diminuem a ocorrência e erros (ou as sua importância) e, não menos importante, contribuem para a satisfação dos usuários. A satisfação é um critério importante, embora não o único, para determinação da qualidade global da aplicação. De um modo geral, este é um critério final para que o usuário adquira um software ou visite regularmente um site.

Quando a usabilidade é levada em conta durante o processo de desenvolvimento de interfaces Web, vários problemas podem ser eliminados como, por exemplo, pode-se reduzir o tempo de acesso à informação, tornar informações facilmente disponíveis aos usuários e evitar a frustração de não encontrar informações no site. Se o site em questão for uma aplicação de comércio eletrônico, por exemplo, tais problemas podem significar redução nas vendas ou mesmo venda nenhuma. Se o objetivo for, por exemplo, ensino à distância, alunos podem se sentir frustados, desmotivados e apresentar baixo desempenho devido a ocorrência frequente de problemas de usabilidade.

É necessário reconhecer que a usabilidade está relacionada ao tipo de aplicação em questão, perfil dos usuários, contextos de utilização, etc., que são variáveis. Além disso, tais valores podem se modificar em função do tempo com o crescimento da população de visitantes, mudança dos requisitos e recursos da aplicação e mesmo atualização da tecnologia. Assim, a determinação da usabilidade pode variar em função

destes critérios e, assim, não se pode garantir que um projeto terá 100% de usabilidade, por exemplo.

A tendência atual em avaliação é tentar identificar os problemas de usabilidade tão cedo eles possam ser detectados na interface. Uma vez identificado, o problema pode ser solucionado ou, ao menos, seus efeitos podem ser minimizados. Esta abordagem é utilizada também para o desenvolvimento de aplicações tradicionais, não apenas para Web. Contudo, observa-se que em aplicações Web as atualizações são muito mais frequentes que em outros tipos de interface. Além disso, o caráter distribuído das aplicações Web distancia os usuários dos desenvolvedores e, por vezes, torna-se difícil identificar quem são de fato os reais usuários da aplicação e quais são suas expectativas.

Existe uma série de métodos de avaliação que podem ser utilizados em diferentes etapas do desenvolvimento de interfaces Web. Alguns desses métodos vêm sendo empregados durante vários anos em outros tipos de interfaces como, por exemplo, o método de análise de interação de usuários em laboratórios de usabilidade. Outros métodos são adaptações de métodos tradicionais para a Web, como por exemplo o uso de questionários para avaliar satisfação dos usuários (WAMMI). Outros ainda, são métodos especialmente desenvolvidos para a Web, como a análise de arquivos de log. Cada método apresentado aqui auxilia na identificação de um gama específica de problemas de usabilidade. Não existe um único método capaz de identificar todos os problemas de usabilidade possíveis em uma interface, embora o método de análise de interação do usuário seja um dos mais completos e, algumas vezes, considerado um método de referência. Em resumo, mais de um método deve ser utilizado para cobrir um espectro maior de problemas de usabilidade. Além disso, devem ser aplicados regularmente para que possam acompanhar as alterações de conteúdo, mudanças na população de usuários e inclusões de problemas de usabilidade às interfaces durante a manutenção do site.

O objetivo principal deste capítulo é descrever o processo de avaliação de usabilidade de interfaces Web e alguns métodos e ferramentas que podem ser utilizados neste processo. Para contextualizar este assunto, são discutidos na parte inicial do capítulo os problemas de usabilidade mais frequentes em interfaces Web, como identificá-los e algumas soluções possíveis (seção 3.2), são apresentadas as características do ambiente Web (seção 3.3) assim como uma visão panorâmica sobre concepção de interfaces Web (seção 3.4), de modo a permitir uma compreensão melhor dos objetivos e procedimentos envolvidos na avaliação (seção 3.5) e as ferramentas existentes (seção 3.6). Finalmente, a seção 3.7 contém as considerações finais deste capítulo incluindo um roteiro para realização de avaliações de usabilidade e um guia para selecionar os métodos mais adequados, de acordo com a etapa de desenvolvimento do site e do orçamento disponível.

3.2. Usabilidade

Nesta seção definiremos usabilidade, discutiremos suas características e apresentaremos alguns exemplos de problemas de usabilidade. Abordaremos também o conceito de acessibilidade, cada vez mais considerado devido à universalização da informação e à necessidade de inclusão digital. Um ciclo de projeto que contemple usabilidade (e também acessibilidade) é também discutido.

3.2.1. O que é usabilidade?

Usabilidade é o termo usado para descrever a qualidade da interação dos usuário com uma determinada interface (Bevan, 1995). Esta qualidade está associada, segundo (Nielsen, 1993), aos seguintes princípios:

- facilidade de aprendizado;
- facilidade de lembrar como realizar uma tarefa após algum tempo;
- rapidez no desenvolvimento de tarefas;
- baixa taxa de erros;
- satisfação subjetiva do usuário.

Considera-se que a interface tem um problema de usabilidade se um determinado usuário ou um grupo de usuários encontra dificuldades para realizar uma tarefa com a interface. Tais dificuldades podem ter origens variadas e ocasionar perda de dados, diminuição da produtividade e mesmo a total rejeição do software por parte dos usuários.

Uma grande parte dos problemas relacionados à interfaces Web diz respeito a navegação, ou seja, os usuários têm dificuldade para encontrar a informação desejada no site ou não sabem como retornar a uma página anteriormente visitada. Outros problemas são ocasionados pelo uso de recursos multimídia de maneira inadequada como, por exemplo o uso abusivo de muitas cores numa mesma página. A Figura 3.1 apresenta um exemplo negativo (contra-exemplo) de interface que utiliza inadequadamente muitas cores, frames e textos em destaque.

Embora seja discutível o gosto por cores, sabe-se que o uso excessivo delas em uma mesma página causa fadiga visual, desvia a atenção do usuário do conteúdo e pode tornar a página ilegível se as cores de fundo e texto não são escolhidas adequadamente. No entanto os problemas de usabilidade em interfaces Web não se resumem apenas à links não disponíveis e uso de cores. Alguns problemas podem ocorrer apenas durante a utilização em contextos especiais de uso.

A Figura 3.2 a apresenta um problema de usabilidade resultado de um funcionamento inadequado de um componente da interface. A aplicação em questão tem por objetivo ensinar inglês através de exercícios. No exemplo mostrado, o aluno deve preencher o campo de formulário com uma palavra que completa uma frase, usualmente apresentada entre aspas duplas (""). No entanto, a frase a ser completada não aparece durante a realização do exercício e, portanto, o aluno não pode responder o teste e sua avaliação será prejudicada por este problema de usabilidade.

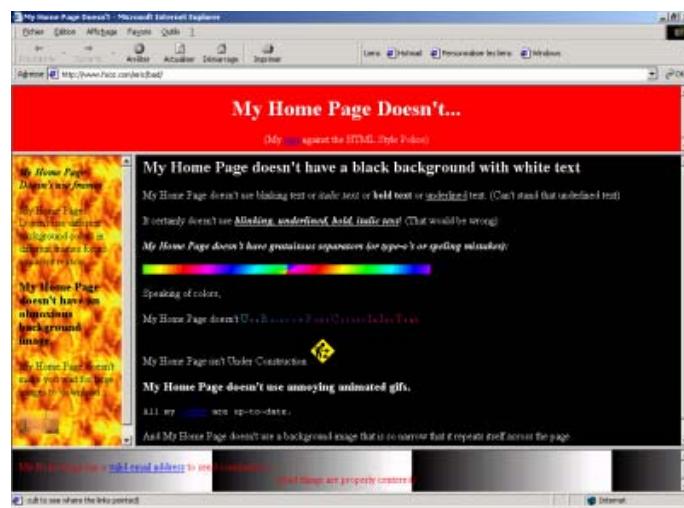


Figura 3.1: Exemplo negativo disponível em: <http://www.fooz.com/eric/bad/>

a) Ausência de informação

b) Inadequação ao contexto

Figura 3.2: Exemplos de problemas de usabilidade encontrados em <http://www.englishpractice.com/>

Alguns problemas de usabilidade ocorrem apenas quando fatores culturais estão envolvidos. Considere, por exemplo, um site Web visitado por usuários do mundo inteiro. É possível que algumas pessoas encontrem dificuldades para usar o site simplesmente porque as referências culturais utilizadas não são as mesmas. Considerando o exemplo da figura 3.2.b que apresenta um outro exercício do site *englishpractice.com*. A questão apresentada é “Qual é a cor de um limão?” e a resposta óbvia para uma criança no Brasil é “verde” (*green*) simplesmente porque esta é a cor mais frequente dos limões no país. Este é um problema de usabilidade apenas para usuários que não estão habituados a comprar limões amarelos (*yellow*) na feira. O problema maior está no fato de que nenhuma informação adicional é fornecida ao aluno, que fica sem saber porque a resposta correta é amarelo.

A interpretação do que é um problema de usabilidade pode variar, e portanto o quê representa um problema para um usuário pode não ser um problema para outro. Por exemplo, considere dois usuários sendo que o primeiro tem uma conexão internet de

rápido acesso (tipo ISDN) e o segundo usa uma conexão modem de 56 kb; provavelmente eles tem opiniões diferentes sobre a velocidade de apresentação de uma página Web que contém muitas imagens. Outro exemplo de origem de problemas é a incompatibilidade entre *browsers* que não suportam da mesma maneira as diferentes tecnologias para construção de interfaces Web; assim, um usuário pode visualizar sem problemas uma interface com o *browser* Internet Explorer enquanto a mesma interface pode apresentar uma série de problemas sobre o *browser* Netscape, e vice-versa. De uma maneira geral, espera-se contentar e eliminar os problemas de usabilidades graves, frequentes durante a utilização da aplicação e que ocorrem com a maior parte do seu público-alvo. Sendo assim, um dos aspectos mais importantes para determinação do que é um problema de usabilidade é conhecer bem os usuários da aplicação.

Embora alguns problemas de usabilidade possam ser específicos à um grupo de usuários, outros podem ser reconhecidos como problemas comuns à grande maioria. Um dos problemas de usabilidade mais frequentes em interfaces Web é a ocorrência de links que contém URLs (*Uniform Resource Locator*) inválidas. Outro problema comum a todos os usuários é a dificuldade de encontrar a informação desejada dentro de um site, embora as razões pelas quais isso ocorre possam ser diversas. É difícil generalizar e descrever todos os tipos possíveis de problemas de usabilidade que podem ser encontrados. Contudo, pode-se identificar algumas métricas ou fatores a serem observados para a determinação de um problema de usabilidade tais como:

- **Desempenho do usuário durante a realização de tarefas:** a observação (direta ou indireta) da realização de tarefas por usuários permite a verificação das seguintes métricas:
 - *Conclusão de tarefas* (com sucesso, parcialmente concluída, não-concluída): tarefas que não são concluídas ou o são apenas parcialmente são um forte indício de que existe algum problema de usabilidade;
 - *Tempo de realização da tarefa*: mesmo se concluída com sucesso, um tempo excessivamente longo pode indicar um esforço desnecessário sendo exigido do usuário;
 - *Ocorrência de erros*: vários tipos de erros podem ocorrer durante a realização de uma tarefa. Se o erro é causado por uma operação do usuário por exemplo, deve-se investigar se a interface não induz ao erro através de comandos complexos ou ausência de mensagem adequadas. Se o erro é produzido por uma atividade do sistema, deve-se verificar como o usuário é advertido da ocorrência e que suporte é oferecido pela interface para efetuar a recuperação deste erro;
- **Satisfação subjetiva do usuário:** a usabilidade é também uma qualidade subjetiva que compreende a opinião do usuário da interface; se o usuários estão satisfeitos com a interface, o efeito de eventuais problemas é minimizado;
- **Correspondência com os objetivos do usuário:** independente das tarefas suportadas pelo sistema, verifica-se se os objetivos do usuários foram alcançados. Esta é uma métrica que pode ser quantitativa ou qualitativa, de acordo com o que é considerado como objetivo final dos usuários;

- **Adequação à padrões** (normas, recomendações ergonômicas, etc.): Grande parte do conhecimento sobre usabilidade é organizado na forma de normas e recomendações ergonômicas tais como as definidas pela ISO9241. Tais recomendações descrevem padrões conhecidos de problemas e, em alguns casos, propõem soluções ou alternativas para evitá-los. A aplicação de tais recomendações durante o desenvolvimento da interface pode realmente evitar ou reduzir vários problemas de usabilidade. Pode-se verificar a usabilidade inspecionando uma interface em relação a tais recomendações. Se a interface as segue pode-se estimar que os problemas de usabilidade foram evitados.

3.2.2 Acessibilidade

Acessibilidade (*accessibility*) é o termo usado para descrever problemas de usabilidade encontrados por usuários com necessidades especiais como, por exemplo, usuários que tem algum tipo de dificuldade auditiva ou visual. Acessibilidade implica em tornar utilizável a interface por qualquer pessoa, independente de alguma deficiência física, sensorial, cognitiva, condição de trabalho ou barreiras tecnológicas.

A maioria das recomendações ergonômicas e recomendação para acessibilidade não limita a utilização da interface apenas à pessoas com necessidades especiais. Na verdade, algumas das recomendações podem ser mesmo úteis por qualquer usuário, como os exemplos a seguir³:

- **Imagens e animações:** use o atributo *alt* para descrever a função de cada imagem;
- **Mapas clicáveis:** use mapas clicáveis do tipo *map* e texto nos pontos com links;
- **Multimídia:** incluir transcrição de áudio e descrição de vídeos;
- **Links hipertexto:** use texto com significado quando lidos fora do contexto e evite “clique aqui”, por exemplo;
- **Organização das páginas:** use cabeçalho, listas e estruturas consistentes. Use *CSS (Cascade Style Sheets)* se possível;
- **Frames:** use elementos *noframe* e títulos com significado;
- **Tabelas:** faça linha por linha regível. Resuma a tabela.

Accessibilidade e usabilidade são conceitos fortemente relacionadas pois ambos buscam melhora a satisfação e eficiência de uso da interface. Contudo acessibilidade diz respeito a uma população muito mais ampla e genérica. Neste documento a usabilidade será o tema principal e recomenda-se aos interessados em acessibilidade que visitem os sites <http://www.hcibib.org/accessibility/> e <http://www.w3.org/WAI/> para maiores informações. Existem várias organizações que se preocupam com a acessibilidade de sites Web, aqui porém, serão feitas referências apenas a iniciativa WAI da W3C.

³ As regras acima são apenas um resumo de regras para acessibilidade. Para um conjunto completo de regras ergonômicas para acessibilidade verifique o site <http://www.w3.org/WAI/>

3.2.3. Ciclo de vida do projeto com usabilidade

Na maioria dos casos, problemas de usabilidade somente são identificados durante a utilização da interface, em situações e contextos de utilização especiais. A maioria dos autores concorda que o processo de desenvolvimento de interfaces com usabilidade, Web ou não, é um ciclo contínuo de design e avaliações de usabilidade, como o representado na figura 3.3. Neste ciclo, inicia-se com a identificação de usuários, tarefas e requisitos para a aplicação. Tais requisitos são utilizados como entrada para a construção de um protótipo que, em seguida, é avaliado com relação a sua usabilidade. Os problemas de usabilidade identificados na avaliação são solucionados na versão seguinte e uma nova avaliação de usabilidade se segue. O ciclo termina quando nenhum problema de usabilidade for identificado ou, pelo menos, os problemas mais graves tenham sido solucionados na interface.

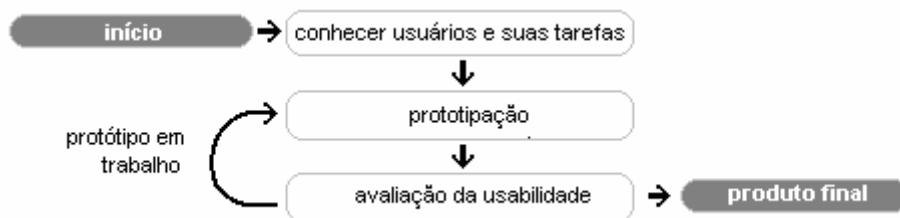


Figura 3.3: Ciclo de vida do projeto com usabilidade

O ciclo de vida acima é uma representação simplificada. Outros autores propõem etapas intermediárias de design e especificação. Contudo, o processo cíclico apresentado aqui é suficiente para descrever o princípio.

O grau de desenvolvimento de um projeto pode ser avaliado em função da complexidade dos protótipos utilizados. Protótipos nada mais são do que versões simplificadas da interface mas que descrevem algumas funções de maneira que possa ser possível analisar a futura interface. Os primeiros protótipos podem ser simples croquis ou desenhos feitos sobre papel. Este tipo de protótipos têm um baixo custo de realização e são ideais para testar diferentes possibilidades de organização da informação em um site. Na sequência pode-se construir protótipos em HTML que descrevem a estrutura de apresentação, embora o conteúdo esteja incompleto.

Cada protótipo deve ser avaliado no sentido de tentar identificar problemas de usabilidade, pois o custo da solução de problemas é inversamente proporcional ao estágio de desenvolvimento da aplicação. Deve-se observar que, dependendo do estágio de desenvolvimento, pode-se empregar um método de avaliação de usabilidade diferente. Por exemplo, pode-se utilizar o método de avaliação heurística mesmo com protótipos em papel, mas nesta etapa de desenvolvimento, a análise de arquivos de *log* é inviável sendo este método mais adequado quando a interface encontra-se em uso.

Deve-se considerar que, devido às frequentes atualizações necessárias à manutenção de um site Web, uma interface Web será sempre um protótipo avançado dentro do ciclo de vida de projeto acima. As modificações no site podem incluir, e com freqüência incluem, problemas de usabilidade devidos a remoção de links ou inclusão de URLs inválidas, modificações na estrutura do site que alteram profundamente a navegação, entre outras. Assim, a cada modificação no site, deve-se considerar a sua avaliação para assegurar que a modificação não alterou a usabilidade do site.

Uma vez identificado um problema de usabilidade, o passo seguinte é procurar uma solução. O custo da solução de um problema pode ser apenas algumas linhas de código corrigidas em alguns minutos ou transformações profundas na estrutura do site, o que podem levar semanas. Além disso, com freqüência, identifica-se não um problema de cada vez mas vários em uma mesma avaliação. É necessário dar prioridade para aqueles que são mais importantes e que exigem uma solução imediata. Por isto, é extremamente importante atribuir a cada problema de usabilidade um grau de severidade.

O grau de severidade pode ser avaliado como relação ao impacto (grave, importante ou impacto menor) sobre a realização de tarefas e freqüência com o qual o problema ocorre. Embora outras escalas para avaliar severidade possam ser utilizadas, sugere-se a escala de severidade proposta por Woolrych e Cockton (2001). Assim, com relação ao impacto, os problemas podem ser classificados como:

- **Grave:** usuários precisam de mais de 2 minutos sem progresso na realização da tarefa. Usuários abandonam a tarefa ou demonstram *stress* na realização da mesma. Usuários não conluem a tarefa.
- **Importante:** usuários gastam até 2 minutos e obtêm êxito na realização da tarefa. Usuários podem demonstrar stress visível ou perda de qualidade de interação.
- **Impacto menor:** usuários encontram o problema mas conseguem contorná-lo sem prejuízo importante para a qualidade de realização da tarefa.

A segunda dimensão para determinar severidade é a freqüência com que um problema ocorre. Assim temos:

- **Grande freqüência:** problemas ocorrem com mais de três usuários.
- **Média freqüência:** problemas ocorrem com dois ou três usuários.
- **Baixa freqüência:** problemas ocorrem com um usuário.

Essa escala de freqüência considera que apenas 5 usuários participam da avaliação. Deve-se observar que ela deve ser ajustada para para um número maior ou menor de participantes. Outra observação importante é que em alguns métodos de avaliação, como no caso da avaliação heurística, não há a participação de usuários. Neste caso, deve-se considerar a freqüência de problemas identificados pelos avaliadores.

Deve-se salientar que a importância e a freqüência dos problemas de usabilidade varia em função da representatividade dos usuários participantes. Se todos os usuários tem um perfil muito próximo, provavelmente os mesmos problemas serão identificados por todos os participantes e darão uma grande importância a todos os problemas identificados, sejam eles graves ou não. Por outro lado, pode ocorrer que usuários diferentes encontrem problemas distintos que não se repetem com outros usuários, dando a falsa impressão de que não se trata de problemas graves. Para resolver tais questões, deve-se usar da experiência em outras avaliações e bom senso.

3.3. O Ambiente Web

Para melhor entender os problemas de usabilidade no ambiente Web, é necessário compreender como a Web funciona, o que são aplicações Web e quais são as diferentes tecnologias que podem ser utilizadas para construir interfaces neste ambiente.

3.3.1. A arquitetura cliente-servidor

O primeiro esboço da Web foi publicado no artigo “*Information Management: A Proposal*” (Berners-Lee, 1989; 1994)⁴ como uma solução aos problemas do CERN de gerência das informações sobre os projetos realizados em seus laboratórios. Desde então, o projeto evoluiu e são várias as razões para o seu atual sucesso, mas duas merecem ser destacadas: sua arquitetura simples mas eficiente e uma interface igualmente simples, originalmente baseada no paradigma de hipertextos. A arquitetura é basicamente um cliente/servidor instalado sobre uma rede de computadores heterogênea. Do lado do cliente está um programa, chamado *browser* ou navegador, que intermedia a solicitação de informações ao servidor e as apresenta para o usuário. O servidor atende os diferentes clientes bem como outros servidores indistintamente, como mostra a figura 3.4.

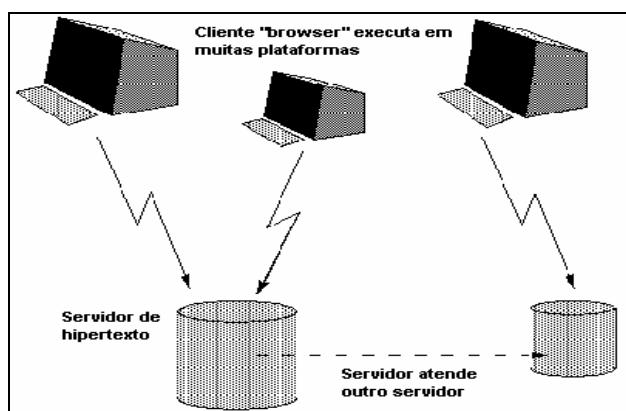


Figura 3.4: Modelo cliente/servidor da WWW

Embora o modelo refencie uma arquitetura cliente/servidor, talvez fosse mais adequado chamá-la de *request/respost* (pedido/resposta) porque, de fato, este é o tipo de comunicação entre as partes; o cliente *browser* solicita um documento ao servidor que processa a chamada, envia o documento ao cliente e encerra a comunicação. Assim, as comunicações entre o cliente e o servidor ocorrem de um modo descontínuo entre as chamadas. Esta arquitetura tem 3 componentes principais: o protocolo de comunicação *HyperText Transfer Protocol* (HTTP), o sistema de endereçamento *Uniform Resource Locator* (URL) e a linguagem *HyperText Markup Language* (HTML).

O protocolo HTTP (Berners-Lee, 1994) é um meio de transporte de arquivos na Web que executa sobre a camada TCP/IP da Internet. O protocolo consiste basicamente

⁴ O artigo escrito em 1989 circulou inicialmente apenas dentro do CERN e posteriormente foi publicado pela sua importância histórica. A primeira demonstração funcional do projeto ocorreu apenas em fevereiro de 1991.

da transição de 4 estados: conexão (o cliente estabelece uma conexão com o servidor); requisição (o cliente envia um pedido ao servidor); resposta (o servidor devolve uma resposta ao cliente); e encerramento (a conexão é desfeita por ambas as partes). Quando um documento ou um objeto (como uma imagem, por exemplo) é enviado para o cliente, é anexado um cabeçalho com a informação necessária para que o *browser* possa interpretá-lo e apresentá-lo adequadamente. Isto torna o protocolo independente do *browser*, que ignora o conteúdo de objetos cujo cabeçalho não comprehende.

URL é a forma conhecida de endereçamentos de objetos na Web. Consiste na identificação do esquema utilizado (HTTP, FTP, etc.⁵) seguido do caminho até o objeto ou documento como, por exemplo: **http://www.inf.ufrgs.br/homepage.html**; onde **http** é o esquema, **://** são caracteres de separação, **www.inf.ufrgs.br** é o nome do domínio Internet e **homepage.html** o documento solicitado.

O terceiro componente é a linguagem baseada em hiperdocumentos HTML. Esta linguagem especifica a estrutura e a formatação para documentos do tipo hipertexto através de marcas (ou *tags*) que indicam a forma como este deve ser visualizado. Pode-se destacar duas grandes vantagens na escolha de uma interface baseada em hipertextos como HTML: i) a facilidade para associar informações através de elos ou *links* permitindo criar grandes redes de informações; e ii) o mecanismo de navegação uniforme com a simples seleção de objetos associados a *links*. Uma vantagem secundária, é o fácil aprendizado, mesmo por usuários não especializados.

A Web foi concebida como um meio para o intercâmbio de informações, geralmente no formato de documentos; por isto é usual tratar as informações publicadas neste ambiente indistintamente como documentos, hipertextos ou hiperdocumentos.

3.3.2. Aplicações Web

Podemos definir uma aplicação Web como uma aplicação de software que utiliza a Web como ambiente de execução. Aplicações Web envolvem Sites Web ou sistemas Web (Conallem, 2000). Sites Web é a forma original de sistema hipermídia distribuído, criado por Tim Bernes-Lee, com o propósito de permitir pesquisa e acesso direto a documentos e informações publicadas em vários computadores que formam a Internet. Documentos, arquivos armazenados em um computador servidor, são acessados e visualizados através de um software chamado browser, instalado no computador cliente, utilizando-se da infraestrutura da internet, através do protocolo HTTP (Hyper Text Transfer Protocol).

A figura 3.5 mostra a arquitetura básica de um Site Web, onde um servidor Web recebe uma solicitação de um browser, localiza o documento em um sistema de arquivos local, e envia o documento de volta ao browser. Os recursos no sistema são interligados entre si através de *links*, que são a forma usual de navegação no sistema. Estes recursos podem ser não só documentos textuais, mas também imagens, vídeo e áudio.

Uma aplicação Web amplia o conceito de Web Site, ao adicionar funcionalidade ao sistema. Em outras palavras, aplicação Web é um sistema Web que permite aos

⁵ Os esquemas mais utilizados são HTTP, TELNET, *File Transfer Protocol* (FTP), *Gopher Protocol* (GOPHER), *Eletronic Mail Address*(MAILTO), *Usenet News* (NEWS) e *Wide Area Information Server* (WAIS).

usuários executarem lógica de negócio com um browser Web. Uma aplicação Web deve ser entendida como uma forma de uso de software acessando dados persistentes através do serviço Web, permitindo a construção dinâmica de páginas para manipular estes dados. Diferente de Sites Web estáticos, onde o conteúdo é um arquivo ou documento pré-formatado (usando um editor HTML, por exemplo), aplicativos Web devem construir dinamicamente o conteúdo em função da interação do usuário com as páginas, via browser. A arquitetura básica é mostrada na figura 3.6.

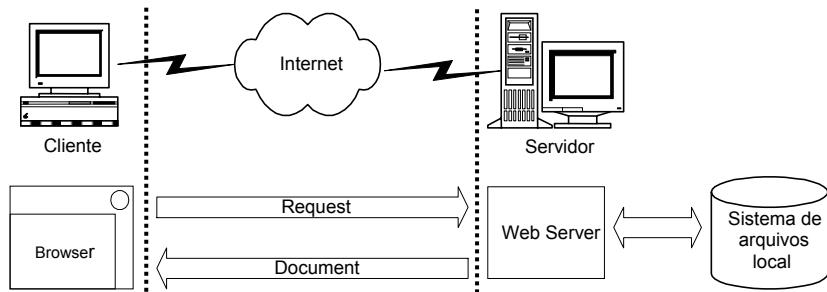


Figura 3.5: Arquitetura básica de Site Web

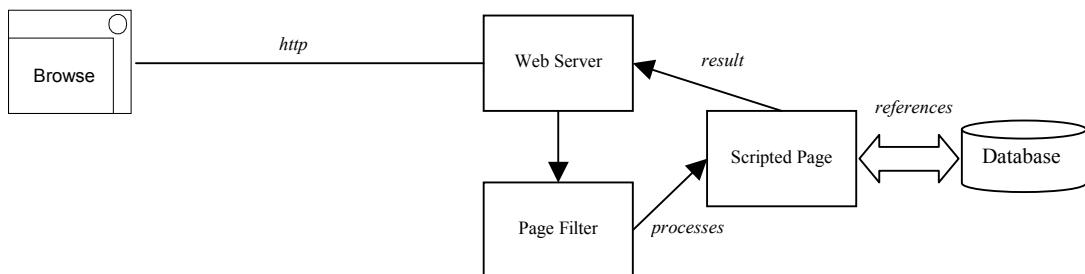


Figura 3.6: Arquitetura de um site web dinâmico

3.3.3. Diversidade de tecnologias para interfaces Web

Inicialmente as interfaces Web nada mais eram do que documentos HTML contendo texto e imagens interligados por *links* que permitiam navegar de um documento a outro. Estes recursos suportados pela primeira versão de HTML 1.0 são suficientes apenas para um número limitado de aplicações. A necessidade de utilizar a Web como um ambiente de base para aplicações mais complexas como, por exemplo transações com banco de dados, motivou o desenvolvimento de novas tecnologias. Tecnologias para especificação de interfaces Web tais como CGI, CCS, JavaScript, etc., foram criadas como soluções alternativas para diferentes problemas considerando o desenvolvimento de aplicações Web. Do ponto de vista da interface, cada tecnologia suporta um estilo de interação diferente e, potencialmente, pode incluir problemas de usabilidade diferentes.

Não é o objetivo aqui descrever exaustivamente todo o potencial de cada uma das tecnologias, mas segue abaixo uma rápida descrição seguida de algumas referências úteis sobre as tecnologias mais utilizadas para construção de interfaces Web. É importante que o avaliador esteja familiarizado com a tecnologia utilizada na construção da interface a ser avaliada, pois isto vai lhe permitir dar melhores sugestões para os problemas encontrados.

HTML

HTML é um formato não proprietário, baseado na *Standart Generalized Markup Language* (SGML) para criação de hipertextos na Web. HTML utiliza marcas ou *tags* tais como, e , para estruturar ou formatar o documento em formato texto. A mais recente versão, HTML 4.0 ou HTML dinâmico (DHTML), tornou-se uma recomendação W3C em dezembro de 1997. Esta nova versão incorpora os padrões criados pelos fabricantes de *browser*, tais como, CSS, frames, JavaScript, além de aperfeiçoar as tabelas e melhorar a resolução impressa dos documentos.

Referências úteis:

- <http://www.icmsc.sc.usp.br/manuals/HTML/> (em português)
- <http://www.w3.org/MarkUp/>
- <http://www.echoecho.com/html.htm>
- <http://nscience.chungbuk.ac.kr/references/html/>

CGI

O protocolo HTTP permite a chamada de programas executáveis hospedados no servidor. Estes programas são chamados *Common Gateway Interface* (CGI) e podem ser escritos em qualquer linguagem de programação tal como C ou Perl. Quando um programa CGI é executado, o resultado do seu processamento pode ser direcionado para o *browser* cliente. Assim pode-se gerar aplicações complexas com o acesso a banco de dados ou criação interativa de documentos. Todo programa CGI é executado no servidor e a interface gerada como saída para o usuário continua estática na sua essência.

ASP e PHP são variações da tecnologia CGI. A diferença principal é que o código a ser executado no servidor é incluído dentro do código HTML ao invés de ser armazenado no servidor. Do lado do servidor um programa interpreta o código ASP ou PHP e processa as instruções devolvendo ao cliente o resultado.

Referências úteis:

- CGI e PERL - <http://www.w3.org/CGI/>
- CGI - <http://hoohoo.ncsa.uiuc.edu/docs/tutorials/cgi.html>
- PHP: http://www.webdevelopersjournal.com/articles/why_php.html,
<http://www.php.net/>, <http://www.phpbuilder.com/>
- ASP: <http://www.aspdeveloper.net/>,
<http://www.advantage.co.nz/ur/aspquiz.htm>

CSS

Cascade Style Sheets permitem fazer com que a apresentação de páginas Web seja determinada por um conjunto de especificações de formatação de página e pela especificação de preferências tipográficas e outras características do dispositivo do cliente, de forma a garantir a continuidade visual do site. Anexando *Style Sheets* à documentos HTML é possível mudar a apresentação de documentos sem adicionar novas marcas ou comprometer os mecanismos de independência de plataforma. O nome *Cascade* implica em que diferentes estilos podem ser combinados em um mesmo documento. A sintaxe da linguagem é um pouco diferente de HTML e requer um

esforço de aprendizagem. Talvez, a principal vantagem de CSS seja barrar a criação de novas marcas em HTML para formatação dos documentos.

Referências úteis:

- <http://www.w3.org/Style/>
- <http://www.awl.com/cseng/titles/0-201-41998-X/liebos/>

JavaScript

JavaScript é uma linguagem interpretada embutida dentro de arquivos HTML. O funcionamento é simples: quando um destes arquivos é carregado, o próprio *browser* interpreta o *script* e realiza as operações especificadas. Embora seja bastante limitada (não suporta aplicações complexas nem transacionais que requerem comunicação via rede), JavaScript permite a criação de animações de imagens, botões e funções para validação de campos em formulários HTML, por exemplo. O diferencial é a possibilidade de criação daquilo que pode ser chamado documentos dinâmicos, ou seja, documentos que reagem imediatamente às interações do usuário.

Uma última nota a respeito da tecnologia de *Scripts* é a versão Jscript, um clone do JavaScript desenvolvido pela Microsoft para executar sobre o *browser* Internet Explorer. Os recursos e objetivos são semelhantes; contudo, JavaScript e Jscript não são totalmente compatíveis.

Referências úteis:

- <http://msdn.microsoft.com/scripting/default.htm?/scripting/jscript/default.htm>
- http://www.webdevelopersjournal.com/articles/jsintro1/js_begin1.html

Java

Java é uma linguagem orientada a objetos desenvolvida pela Sun Microsystems e cuja principal característica é ser uma linguagem interpretada que requer uma “*java virtual machine*” (JVM - máquina virtual Java) para executar os programas nela escritos. De fato, Java também é o nome de uma marca usada para designar as tecnologias Java *Applications*, *Applets* e Java Beans. *Applets* são pequenos programas Java que rodam apenas dentro de um *browser* cliente, que implementam uma JVM.

Diferente de um *script* JavaScript, uma *applet* tem apenas uma chamada dentro do código HTML. A partir desta chamada, o código Java pré-compilado é carregado e interpretado pelo *browser*. A arquitetura da linguagem garante independência de plataforma com a implementação da *Abstract Windows Toolkit* (AWT), que encapsula os mecanismos de apresentação da interface. Contudo, por questões de segurança, todos os recursos de acesso I/O foram retirados. Com isso, *applets* Java podem ter grande expressão computacional, porém, não podem interagir no ambiente do computador do usuário. Como compensação a esta limitação, existem facilidades para comunicação em rede com servidores WWW ou diretamente com outras aplicações.

Referências úteis:

- <http://java.sun.com/>

Plug-ins

Plug-ins é a denominação utilizada para designar componentes que são adicionados aos *browsers* incorporando capacidade de manipulação de dados em outros formatos que HTML. É uma opção adotada por diversas empresas de software para permitir que seus produtos se integrassem ao ambiente WWW sem a necessidade de desenvolver seus próprios *browsers*. As principais aplicações para *plug-ins* são: visualizador de imagens, visualizadores de documentos, animações, aplicações 3-D e VRML, áudio, vídeo, entre outras. Exemplos:

- Shockwave, animações em flash: <http://www.macromedia.com/shockwave>
- Real Player, arquivos de audio/video: <http://www.real.com>
- QuickTime, video: <http://www.apple.com/quicktime/>

Referências úteis:

- <http://www.learnthenet.com/english/html/56plugins.htm>

3.3.4. Diferenças entre *browsers*

A arquitetura cliente-servidor da Web é baseada no princípio fundamental de que usuários devem visualizar a mesma informação independente da plataforma de software e hardware utilizado. Contudo, para visualizar as informações é necessário que cada usuário instale em seu computador um *browser* para interpretar o código HTML. Embora o código HTML seja o mesmo, independente da plataforma ou do sistema operacional utilizado, o *browser* em si é específico à cada plataforma.

Até 1993, um único *browser* (Mosaic) era disponível nas plataformas Windows, OS/Mac e Unix. Apesar de algumas pequenas diferenças, pode-se dizer que todos os usuários tinham acesso ao mesmo conteúdo de informação e a mesma apresentação. Este panorama começa a mudar com o surgimento de concorrentes do Mosaic, como o Lynx, Netscape e Internet Explorer, entre outros. Os fabricantes destes *browsers* começam a incluir características diferentes em seus *browser* como, por exemplo, *tags* proprietárias que apenas executam em determinado *browser*. Para evitar a proliferação de inúmeros dialetos de HTML, novas versões de HTML incluem a novas *tags* padronizando-as. Surgem assim as versões HTML 2.0, 3.2 e 4.0.

Contudo, a concorrência entre os *browsers* não se restringe à padronização da linguagem HTML mas continua com o suporte a outras tecnologias tais como *JavaScript*, *Cascade Style Sheet* (CSS) e XML, entre outras. Apesar dos esforços realizados pelo W3C (órgão que regulamenta padrões para tecnologia Web – <http://www.w3.org/>) para uniformizar a tecnologia para Web, alguns *browsers* continuam adotando padrões proprietários ou suportando apenas parcialmente algumas características. A tabela 3.1 resume algumas características dos *browsers* mais utilizados atualmente na plataforma MS Windows⁶. Observe as diferenças entre o suporte a tecnologias e recursos de formatação de texto.

⁶ Tais características são válidas apenas para os *browsers* citados quando executando sob a plataforma MS Windows. Veja detalhes sobre outras plataformas em: http://hotwired.lycos.com/webmonkey/reference/browser_chart/index.html.

Browser	Java	frames	tabelas	Plug-ins	Tamanho de texto	Cor de texto	Javascript	CSS	Gif89	DHTML	Cor em Tabelas	XML
Explorer 6.0	X	X	X	X	X	X	X	X	X	X	X	X
Explorer 5.5	X	X	X	X	X	X	X	X	X	X	X	X
Explorer 5.0	X	X	X	X	X	X	X	X	X	X	X	S
Explorer 4.0	X	X	X	X	X	X	X	X	X	X	X	
Explorer 3.0	X	X	X	X	X	X	X	X	X		X	
Explorer 2.0			X		X	X						
Netscape 6.1	X	X	X	X	X	X	X	X	X	X	X	X
Netscape 6.0	X	X	X	X	X	X	X	X	X	X	X	X
Navigator 4.7	X	X	X	X	X	X	X	X	X	X	X	
Navigator 4.5	X	X	X	X	X	X	X	X	X	X	X	X
Navigator 3.0	X	X	X	X	X	X	X		X		X	
Navigator 2.0	X	X	X	X	X	X	S		X			
Navigator 1.1			X		X							
Mosaic 3.0		X	X		X							
Mosaic 1.0												
Mozilla	X	X	X	X	X	X	X	X	X	X	X	X
AOL browser 5.0		X	X		X	X	X				X	
AOL browser 4.0		X	X		X	X					X	
Opera 5.11	X	X	X	S	X	X	X	X	X	X	X	X
Opera 4.02	X	X	X	S	X	X	X	X	X	X	X	X
Lynx		X	X									

Legenda: X Suporte S Suporte parcial Nenhum Suporte

Tabela 3.1: Diferenças entre web browsers para plataforma MS Windows

As diferenças de apresentação e suporte à tecnologia podem afetar diretamente a usabilidade das interfaces. Por exemplo, considere um site Web cuja navegação é baseada em menu JavaScript e que 30% dos usuários não utilizam um *browser* com suporte a essa tecnologia. Isto significa que 30% dos usuários deste site estão experimentando problemas de usabilidade relativos à navegação!

Atualmente estima-se que 65% dos usuários Web utilizam o *browser* MS Internet Explorer, sendo que o Netscape ocupa o segundo lugar, com pouco mais de 25%. Ainda que o Internet Explorer, aparentemente, domine o mercado, desenvolver interfaces baseando-se apenas para este *browser* pode não ser a melhor estratégia, pois assim se estaria excluindo uma grande parcela de usuários. Existem duas soluções possíveis: a primeira consiste em utilizar um conjunto mínimo de funcionalidades que seja suportado por todos os browser; a segunda é construir diferentes interfaces, cada uma adaptada a um browser diferente e incluir um mecanismo de detecção que identifica o *browser* e direciona a interface mais adequada a cada caso.

É importante que os avaliadores estejam cientes das diferenças entre os *browsers* e levem este aspecto em conta durante a avaliação. Primeiro deve-se avaliar uma interface Web com mais de um browser e em mais de uma plataforma. Além disso, deve-se considerar diferentes versões do mesmo *browser*. Sabe-se que, de maneira geral, usuários são relutantes em atualizar seus *browsers*. Assim, não é raro encontrar usuários utilizando o primeiro *browser* instalado no seu computador.

A avaliação da interface considerando diferentes *browsers*, versões e plataformas, multiplica o esforço de avaliação; mas é o preço a pagar para garantir usabilidade da interface para um público maior de usuários. Assim, recomenda-se que sempre que possível sejam realizadas avaliações com diferentes *browsers*.

3.3.5. Diferenças entre interfaces Web e Wimp

Interfaces Web apresentam algumas diferenças com relação interfaces WIMP (*Windows, Icons, Mouse and Pointer devices*) tais como interfaces encontradas sob as plataformas MS Windows e OS/Mac. Estas diferenças são principalmente relativas ao modo de utilização destas interfaces e restrições do ambiente Web. Entre as principais podem ser citadas:

- Usuários podem utilizar atalhos para acessar partes de uma aplicação Web sem passar obrigatoriamente pela *homepage* do site. Em interfaces Wimp, o designer tem controle total sobre a apresentação da interface e meios de impedir que usuários explorem a interface de maneira outra que aquela predefinida. Mecanismos como *bookmarks*, histórico de navegações e acesso direto a URL usuários para explorar interfaces Web.
- Interfaces Wimp suportam, em geral, interações mais complexas que interfaces Web. Interfaces Web são caracterizadas principalmente pela interação do tipo *browsing* que consiste em navegar entre páginas selecionando links. Usuários Web passam mais tempo navegado entre páginas do que interagindo com objetos contidos na página. Na prática se observa o fenômeno de *scanning* segundo o qual usuários “passam a vista”, isto é, olham rapidamente páginas Web em busca de links ou palavra-chaves que possam identificar o conteúdo procurado e, na maior parte do tempo, não lêem cuidadosamente o conteúdo apresentado.
- Interfaces Web são relativamente fáceis de construir mesmo por pessoas com pouca formação. Em geral, pessoas sem formação específica estão mais suscetíveis a desenvolver interfaces com problemas de usabilidade.
- Interfaces Web são atualizadas muito mais rapidamente que interfaces Wimp. O que significa que a cada modificação é necessário avaliar a usabilidade da interface para garantir que problemas não sejam incluídos. Em interfaces Wimp, um vez concluído e distribuído o software, problemas de usabilidade somente serão resolvidos na futura versão.
- Finalmente, o uso de um *browser* é obrigatório para a Web. Isto implica que uma série de ferramentas e funções são adicionadas à interface e podem interferir no design final.

Estas diferenças devem se levadas em consideração durante todo o processo de avaliação de interfaces Web. Isto implica na adaptação de métodos de avaliação de usabilidade que sejam adequados ao ambiente Web. Em geral, são necessários métodos de baixo custo para que se torne viável a avaliação freqüente. Também são necessários métodos de fácil realização que possam ser aplicados mesmo por pessoas sem uma formação específica em IHC (até porque não existiriam especialistas em número suficiente para avaliar todas as interfaces Web atualmente ativas).

3.4. Concepção de interfaces Web

Antes de nos concentrarmos em como avaliar interfaces Web, convém abordar - mesmo que superficialmente - alguns conceitos, modelos, metodologias e ferramentas propostos para a concepção de aplicações Web, cuja meta principal é justamente auxiliar o processo de design de sites Web de forma a termos um processo mais efetivo resultando em sites Web de maior qualidade. De fato, hoje em dia é amplamente reconhecido pela comunidade de Interação Humano-Computador⁷ (IHC), que é preferível dedicar mais esforço para desenvolver aplicações de modo que critérios de usabilidade sejam considerados desde o início e durante todo o seu processo de construção do que para uma fase de avaliação *a posteriori* ao fim deste processo. No entanto, vale a pena reforçar que a avaliação continua a ser essencial sobretudo para validar as escolhas da concepção e confirmar o nível de satisfação dos usuários e deve ser sempre realizada por melhor que seja o processo de concepção que a antecede.

O objetivo desta seção é apresentar algumas das idéias pertinentes e relevantes à concepção de interfaces Web assim como resumir as características de algumas técnicas e ferramentas de desenvolvimento, sejam elas de cunho comercial disponíveis no mercado ou de cunho acadêmico resultantes de trabalhos de pesquisa.

Para isto, primeiramente (seção 3.4.1), alguns fundamentos sobre concepção de interfaces Web são apresentados. A seção 3.4.2 descreve um ciclo de desenvolvimento de aplicações Web, explicando as características de cada etapa do ciclo. A seção 3.4.3 aborda mais detalhadamente o processo de design, enumerando na subseção 3.4.3.1 as diferentes abordagens sistemáticas para concepção de interfaces Web propostos para este processo e na subseção 3.4.3.2 as variadas ferramentas existentes para suportar total ou parcialmente este processo.

3.4.1. Fundamentos

Atualmente muitos aspectos considerados fundamentais na concepção de interfaces Web têm sido estudados, muitos livros sobre o assunto têm sido publicados abordando vários deles e muitas recomendações (guidelines) para concepção de sites têm sido compiladas. Mais do que isto, muitos sites têm sido construídos nos últimos anos e a necessidade cada vez maior de desenvolvimento leva à esta aceleração do interesse sobre o assunto. No entanto, mesmo com toda esta efervescência, não se tem um consenso sobre qual as melhores características de um site, qual o melhor processo para construí-lo, quais os modelos a usar no seu processo de desenvolvimento, quais as melhores formas de estruturá-lo, de permitir navegação e de usar a tecnologia disponível para tornar seu uso mais eficiente e agradável. Cada autor apresenta o seu conjunto de requisitos, recomendações, técnicas, modelos e ferramentas que considera mais essencial. Por isto, concordamos com Ben Shneiderman, um dos mais renomados autores da área de IHC, que comentou em 1997: "...it will take a decade until sufficient experience, experimentation, and hypothesis testing clarify issues". Desta forma, apresentaremos aqui um conjunto de tópicos que, embora não sejam eventualmente consensuais, julgamos importantes para quem for conceber sites e que servirão para a

⁷ Tradução adotada atualmente pela comunidade brasileira para a expressão inglesa Human-Computer Interaction (HCI).

compreensão de alguns dos tópicos discutidos posteriormente na parte de avaliação. Como não é intenção desta seção discutir com muita amplitude e profundidade o processo de concepção, estes tópicos incluem as dimensões de concepção de sites Web (seção 3.4.1.1), o ciclo de concepção de aplicações Web (seção 3.4.1.2), algumas características (seção 3.4.2) e recomendações (seção 3.4.2.1), metodologias (seção 3.4.2.2) e ferramentas (seção 3.4.2.3) para projeto de interfaces Web.

3.4.1.1. Dimensões de concepção

As aplicações Web podem ser convenientemente descritas como híbridas entre aplicações hipermídia e sistemas de informação. Como uma aplicação hipermídia, uma aplicação Web é acessada de forma exploratória, não linear, e portanto as suas formas de apresentação e navegação são de grande importância. Como um sistema de informação, a estrutura, o tamanho e a dinamicidade dos seus dados exigem soluções metodológicas (como modelos conceituais, métodos de mapeamento entre estruturas, abstrações) e soluções tecnológicas consolidadas e eficazes (como SGBDs e arquiteturas cliente-servidor) que auxiliem a gerenciar esta complexidade e que permitam fácil interoperabilidade, evolução e manutenção. Devido a sua natureza híbrida, há fatores complicadores como a necessidade de manipular tanto dados estruturados (tuplas e registros) quanto dados não-estruturados (multimídia) e de compatibilizar esta variedade de informações a diferentes estilos de apresentação e navegação para usuários com diferentes níveis de competência.

Por isto, segundo (Fraterno, 1999), uma aplicação Web é caracterizada por três dimensões de projeto principais:

- **Estrutural (ou Conceitual):** que descreve a organização da informação a ser gerenciada pela aplicação, suas propriedades estruturais, e os relacionamentos entre si; tipicamente, a dimensão estrutural descreve os objetos, as associações entre eles e seus objetos componentes; modelos para esta dimensão incluem modelos de dados (p.ex. Entidade-Relacionamento) ou de objetos (p.ex. diagramas de classes UML ou OMT) ou quaisquer outros modelos tipicamente usados para modelagem conceitual de sistemas de informação;
- **Navegacional:** que focaliza as facilidades para o acesso e movimentação em relação às informações da aplicação, especificando as ações disponíveis para movimentação direta de um objeto a outro (navegação contextual) e os caminhos de acesso para alcançar os objetos da aplicação, independente de movimentação (navegação não-contextual) ; dado um esquema estrutural, podem haver vários diferentes esquemas navegacionais representando diferentes maneiras de acessar e de se mover através da mesma informação; modelos para esta dimensão são geralmente baseados em máquinas de estados (p.ex. statecharts) ou redes de Petri;
- **Apresentação:** trata como o conteúdo e os aspectos de navegação serão apresentados para o usuário; especificando o layout e o conteúdo de um conjunto de elementos (páginas) similares da aplicação; dado um par (esquema de estrutura, esquema de navegação) podem haver várias diferentes esquemas de apresentação representando diferentes modos de

graficamente exibir a mesma aplicação; entre modelos usados para esta dimensão incluem-se storyboards, Abstract Data Views (Cowan, 1995), croquis, maquetes, protótipos ou quaisquer outros tipicamente usados para representar o componente de apresentação de interfaces.

3.4.2. Ciclo de concepção de interfaces Web

O ciclo de concepção de interfaces Web é uma espiral contínua, notadamente marcado por sucessivas modificações, que são muito mais freqüentes em aplicações Web que em outros tipos de interfaces. Dentro deste ciclo espiral, várias etapas se sucedem, embora o número e a importância de cada uma delas variem em função da abordagem utilizada. O ciclo proposto por (Scapin, 2001) – ver figura 3.7 – compreende as seguintes etapas:

Etapa 1: Engenharia de requisitos: a estrutura do site e o contexto de utilização são identificados;

Etapa 2: Especificação do site: modelos da interface são construídos a partir dos requisitos obtidos durante a análise de requisitos;

Etapa 3: Design do site: modelos são refinados e o site é implementado de acordo com o seu conteúdo;

Etapa 4: Implementação do site: corresponde a criação de páginas HTML e objetos de som/imagem necessários à aplicação;

Etapa 5: Utilização do site e avaliação: são avaliadas a usabilidade da interface e a coerência da interface com relação aos requisitos iniciais;

Etapa 6: Manutenção do site: envolve um ciclo de maior duração que envolve a coleta de novos requisitos e planejamento das modificações identificadas durante a etapa de avaliação.

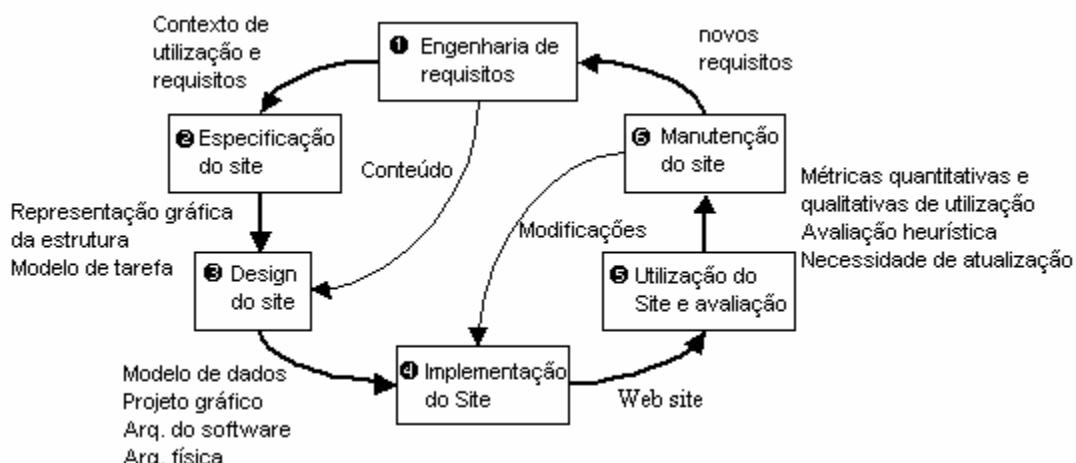


Figura 3.7: Ciclo de desenvolvimento de um website

Este ciclo de desenvolvimento para Web deve ser fortemente baseado em um processo sistemático e cíclico onde o passo de engenharia de requisitos tradicional deve ser seguido pela especificação do site (veja figura 3.7). Uma descrição explícita do site pode ajudar de várias formas no desenvolvimento: formalizando requisitos do usuário, guiando o projeto e a construção do site, documentando um conjunto de informações útil no decorrer das atividades de avaliação e do site. Durante a fase de especificação do

site, são produzidos todos modelos descrevendo requisitos do(s) usuário(s), tarefas e estrutura que será usada para implementar o site.

Observa-se que, neste ciclo, um atalho possível permite a implementação logo após a análise de requisitos sem passar pela etapa de especificação, o que é freqüentemente observado na prática. Contudo, isto dificulta a construção de sites com maior usabilidade pois, devido a necessidade de modificações freqüentes no site, cada vez que um desenvolvedor altera manualmente uma interface ele está sujeito a incluir um problema de usabilidade. Estes problemas podem estar associados a um erro relacionado a operação em si (p.ex. um erro de digitação que torna um *link* inacessível) ou uma modificação na estrutura de navegação que elimina um caminho a um ramo da estrutura antes acessível.

A abordagem de desenvolvimento orientado a modelos visa evitar os atalhos e seguir todas as etapas do ciclo de desenvolvimento. Assim - sempre referenciando a fig. 3.7 - cada alteração da interface (novos requisitos como entrada da etapa 1) deve ser especificada (etapa 2) antes de executada sobre a interface (etapas 3 a 5). Embora a manutenção torne-se mais complexa e demorada, é possível verificar cada modelo a cada alteração. Além disso, a especificação da interface documenta a evolução do site.

3.4.3. Concepção de sites Web

Nos primeiros sites desenvolvidos para a Web, pouca importância era dada aos requisitos dos usuários e à usabilidade das suas interfaces e, por consequência, muitas interfaces Web não preenchiam as necessidades de seus usuários. Esta falha do processo de desenvolvimento foi mais tarde parcialmente investigada quando a preocupação com critérios de usabilidade e o uso de técnicas centradas no usuário (User Centered Design ou brevemente UCD) permitiu incluir os requisitos dos usuários no projeto do site. Por exemplo, o uso combinado ou não de técnicas de *card sorting* e questionários foi popularizado na forma de ferramentas para identificar os requisitos do(s) usuário(s) e estruturar a informação de forma “natural” segundo o ponto de vista deste(s) usuário(s). Embora simples, isto foi reconhecidamente um progresso de atitude orientada a aspectos de interação humano-computador dentro do desenvolvimento no ambiente Web, notadamente ainda muito orientado a tecnologia.

Entretanto, esta mudança de atitude é insuficiente para a obtenção de sites Web de qualidade pois seu design é ainda realizado de maneira muito informal. Atualmente, os projetistas utilizam intensivamente algumas ferramentas para autoria e/ou edição de páginas ou mais raramente para identificação de requisitos mas estas ferramentas (ver seção 3.4.2.3) em sua maioria não fornecem suporte a muitas fases do ciclo de concepção como especificação do site através de modelos e tradução (assistida ou automática) destes modelos visando um projeto das páginas e de elementos do site.

De fato, há surpreendentemente pouco esforço em prover suporte conceitual a modelagem de interfaces WWW. Idealmente, um modelo para uso no desenvolvimento de aplicações Web e suas interfaces deve:

- a) ter uma base rigorosa, sendo que o nível de rigor máximo é o nível de um formalismo, que possui uma fundamentação matemática consolidada e permite verificações formais;
- b) ser completo na sua habilidade de representar os aspectos estáticos e dinâmicos das aplicações Web e suas interfaces; e

c) ser independente de ferramentas implementadas que podem ser a ele associadas.

Para construir sites Web complexos necessitamos, além de entender e representar os conceitos do domínio do problema (classes e objetos, propriedades, associações, regras de negócio, serviços, etc), entender claramente e representar explicitamente as características mais relevantes do(s) usuário(s) e suas tarefas. Além disto deve-se usar este conhecimento para projetar cuidadosamente os aspectos de interação e os respectivos modelos de navegação, diálogo e apresentação envolvidos de forma a conseguirmos desenvolver aplicações usáveis que suportem adequadamente estas tarefas.

Não há um consenso sobre como deve-se realizar a sistematização do processo de desenvolvimento de aplicações Web e quais modelos seriam necessários e/ou suficientes para este processo. Embora a prática informal seja ainda aceitável para projetos de pequeno porte (projetos “*do your own web page*”), isto torna-se crítico para grandes e/ou complexos sites.

Alguns problemas comuns da ausência de enfoque ou da utilização de enfoques informais incluem:

- ambigüidade dos requisitos e das especificações;
- ausência de ferramentas para suportar sistematicamente, incrementalmente e coerentemente o processo de desenvolvimento a partir dos requisitos e especificações; e
- dificuldade de gerenciar a manutenção de um site quando sua complexidade e/ou tamanho aumentam.

O objetivo da utilização de modelos durante o processo de desenvolvimento de aplicações Web e suas interfaces é duplo:

- registrar/documentar o processo de desenvolvimento da aplicação, representando os diferentes estágios e as diferentes informações pertinentes e relevantes para o projeto de interfaces Web com usabilidade. Exemplos típicos são os modelos para as diferentes dimensões de concepção mencionadas mas também informações sobre os usuários (suas características, habilidades, deficiências, freqüência de uso,etc) e sobre as tarefas que eles realizarão com o suporte da aplicação (sua decomposição em sub-tarefas, seu ordenamento, suas restrições, suas pré- e pós-condições de realização, freqüência de realização, criticidade,etc);
- permitir a comunicação entre os membros da equipe (geralmente multidisciplinar) envolvida com o desenvolvimento da aplicação Web.

Modelar uma aplicação Web significa, como para qualquer tipo de aplicação, descrever suas características mais relevantes, sem consideração de detalhes de implementação.

3.4.3.1. Recomendações para projeto de interfaces web

A grande maioria dos métodos e modelos para desenvolvimento de interfaces Web consideram a interface como um componente de uma aplicação Web e portanto são um

subconjunto de métodos e modelos para desenvolvimento de aplicações Web. Estes por sua vez são variantes de métodos e modelos concebidos inicialmente para aplicações hipermídia.

Embora não haja consenso metodológico, alguns autores propõem um conjunto de recomendações que, se consideradas, podem resultar num projeto de sites Web de qualidade. Apresentamos a seguir um resumo destas recomendações, que estão organizadas nos seguintes tópicos:

- Reduzindo o tempo de resposta;
- Preparando conteúdo para Web;
- Organizando o site Web;
- Definindo controles de navegação;
- Projetando páginas Web.

Reduzindo o tempo de resposta

A grande maioria dos usuários Web conectam-se via modem e portanto têm velocidades relativamente baixas de conexão. O tempo de resposta mais significativo para usuários Web é o tempo de carga (download) de páginas. Este tempo é determinado basicamente pelo tamanho da página e pela velocidade de conexão. Como os projetistas podem controlar somente o primeiro, sugere-se:

- projete páginas pensando nas pessoas que conectam-se via modem;
- evite grandes gráficos e geralmente use poucos gráficos nas páginas;
- se usar gráficos, reduza o número de cores, use taxa de compressão elevada e use sempre textos associados aos gráficos (atributos ALT para imagens);

Preparando conteúdo para Web

O conteúdo de um site Web deve ser concebido de modo específico para este meio, através de uma escrita do texto para ser lido on-line e de uma escolha adequada do vocabulário do site.

Escrevendo texto para ser lido on-line

Usuários Web não lêem palavra por palavra, mas tipicamente procurando informação destacada e alternando entre blocos de texto, ou seja, "passando a vista" ou "lendo em diagonal". Logo, o texto na Web deve:

- ser fácil de "passar a vista", incluindo itens para facilitar esta tarefa como tabelas de conteúdo, índices, listas numeradas, realçando palavras-chave, usando títulos e subtítulos significativos;
- ser conciso: linguagem direta, e sem detalhes irrelevantes;
- ser objetivo: evitar jargão "promocional" de marketing e vendas, das quais o usuário geralmente quer distância;

Escolhendo o vocabulário do site

Esta escolha envolve termos do domínio do problema e de preferência adotados comprovadamente pelos usuários: use técnicas de entrevistas, observações, *card sorting* ou *participatory design* para descobrir e confirmar isto. Mais, a noção de que um termo

é "óbvio", "natural", "evidente", e "intuitivo" é um mito: idealmente deve-se oferecer mais de um termo de acesso a usuários que desejam achar informações ou ações no site. Um glossário de sinônimos ou termos relacionados pode ser um auxílio importante.

Organizando o site Web

Organizar um site implica em definir a topologia deste site e escolher o número de níveis em que seu conteúdo será distribuído.

Em relação à topologia, muitos autores defendem que a melhor é a hierárquica (mais fácil de estruturar informação), enquanto outros defendem que a melhor é a forma de rede (mais fácil de navegar). O importante, de fato, é arranjar a informação de uma forma significativa para os usuários e adotá-la em todo o site. Para sites institucionais ou sites com muita informação, sugere-se fortemente uma estrutura hierárquica que inclua links transversais (como uma rede) para ligar informações relacionadas.

Como uma consequência direta da escolha da topologia, o mais importante não é o número de níveis em si mas se a forma de organização do site é significativa para os usuários. Ao invés de sites com muita amplitude (poucos níveis mas com muitos links a cada nível da hierarquia) ou com muita profundidade (muitos níveis com poucos links a cada nível), sugere-se agrupar os links relacionados para diminuir o número de links e facilitar a navegação contextual.

Definindo controles de navegação

A definição de uma boa navegação inclui algumas preocupações básicas:

- tornar links predizíveis: usar o vocabulário adequado para mostrar opções de links e adicionar descritores aos links que os navegadores mais recentes permitem ao usuário visualizar antes de selecioná-lo;
- escolher cores de links visando a manutenção de um código cromático para os links não visitados (azul) e para os links já visitados (vermelho ou roxo);
- aproveitar a estrutura hipertextual e embutir links no texto;
- para os links não embutidos no texto, arranje-os em listas (verticais ou horizontais) como menus de navegação ou como links que marcam o histórico do caminho de acesso percorrido até a página corrente;
- se usar links associados a imagens, sempre inclua rótulos de texto junto às imagens e faça toda a imagem "clicável" e não só uma parte dela.

Projetando páginas Web

Uma página Web é bem projetada se a) revela claramente a estrutura da informação que ela contém; e b) oferece pontos de acesso facilmente reconhecíveis para esta informação. Isto pode ser obtido através de técnicas como agrupamento de informação relacionada, uso de espaços em branco para separar estes grupos e uso de títulos e subtítulos para estruturar a página.

Além disto, é importante considerar também:

- densidade da página: em páginas Web, usa-se menos espaço em branco do que se costuma usar em páginas impressas;

- a quantidade de informação exibida em cada página, que deve ser minimizada: use abreviações bem conhecidas e escreva tão concisamente quanto possível;
- cores de fundo e cores de texto: para facilitar "passar a vista" tipicamente use caracteres escuros com fundos claros; mais detalhes ver (Borges et al., 2000);
- legibilidade do texto: tamanho e tipo de fonte, uso de maiúsculas e minúsculas, estilo do fonte (itálico, negrito, etc) , contraste, e outros parâmetros similares.

Muitos links contendo recomendações e outras informações úteis para concepção de interfaces Web existem. Alguns dos mais interessantes são:

- Yale C/AIM WWW Style Manual ,
<http://info.med.yale.edu/caim/manual/contents.html>
- Creating Killer Web Sites , <http://www.killersites.com/index.html>
- Usabilidade, <http://www.usabilidade.com>
- Site Use-o de Jacob Nielsen , <http://www.useit.com>
- Usable Web , <http://www.usableweb.com>
- Web site usability: design guidelines .
<http://infoweb.magi.com/~asd/guidelines/guide.htm>
- MicroSoft – Improving Web Site Usability and Appeal,
- <http://msdn.microsoft.com/library/en-us/dnsiteplan/html/improvingsiteusa.asp>
- Research-based Web Design and Usability Guidelines ,
<http://usability.gov/guidelines/>

3.4.3.2. Abordagens sistemáticas para concepção de interfaces Web

O estágio atual no campo de métodos e processos de desenvolvimento para Web evoluiu a partir do desenvolvimento da área de aplicações hipermídia standalone (off line).

Nas abordagens a serem discutidas a seguir, há uma intenção bem clara de separar os conceitos em camadas de projeto: projeto conceitual (dimensão estrutural), projeto navegacional (dimensão de navegação), aspectos de interface com usuário (dimensão de apresentação). Além disso, as modelagens utilizadas tentam, na maioria, se distanciar do ambiente de implementação e das tecnologias subjacentes.

Para simplificar, as informações apresentadas nas tabelas 3.2 e 3.3 resumem as principais abordagens existentes para o desenvolvimento sistemático de interfaces Web: Hypertext Design Model (HDM), Relationship Management Methodology (RMM), Object-Oriented Hypermedia Design Method (OOHDM), JESSICA , Web Application Extension (WAE) eWeb Modeling Language (WebML).

Para simplificar a análise de cada método, apresentaremos apenas como cada abordagem considera algumas características tais como passos do processo de desenvolvimento, modelos e notações usados e adoção de ferramentas de suporte.

Abordagens	Passos do Processo	Modelos e Notações	Ferramentas de suporte	Referências Bibliográficas
HDM / HDM-lite	1. Authoring-in-the-large 2. Authoring-in-the-small	Entidade-Relacionamento (E-R)	Autoweb System (HDM-lite)	(Garzotto et al, 1993) (Gaedke et al, 2000) (Frernali, 1998)
RMM	1. Projeto de entidades 2. Projeto de relacionamentos (slice) 3. Projeto navegacional 4. Projeto de protocolo de conversão 5. Projeto de tela de interface de usuário 6. Projeto de comportamento em tempo de execução 7. Construção	E-R + RMDM (Relationship Management Data Model)	RMCase	(Isakowitz et al, 1995) (Diaz et al, 1995)
OOHDM	1. Projeto conceitual 2. Projeto navegacional 3. Projeto de interface de usuário 4. Implementação	UML/OMT (1) Própria (2) Abstract Data Views (3)	OOHDM-Web	(Schwabe et al, 1992) (Rossi et al, 1996) (Rossi et al, 1999) (Segor et al, 2000)
JESSICA	1. Modelagem: a.Diagrama de classes b.Use-cases (cenários) 2. Geração/Parsing 3. Compilação 4. Visualização	UML/XML	JESSICA System	(Barta et al, 1998)
WAE	RUP (Rational Unified Process) 1. Requisitos 2. Análise 3. Projeto 4. Implementação 5. Testes 6. Avaliação	UML c/ estereótipos próprios	Rational Rose	(Conalle, 2000)
WebML	1. Modelo estrutural 2. Modelo de composição 3. Modelo de apresentação 4. Modelo de personalização	Própria/ XML	WebML Tool Suite	(Ceri et al, 2000)

Tabela 3.2: Resumo de abordagens para concepção de aplicações Web

Na tabela 3.2, a coluna **Passos do processo** relaciona os passos realizados no processo de desenvolvimento segundo cada abordagem. Apesar de cada método usar nomenclaturas diferentes para cada passo descrito, podemos perceber a importância ou ausência dada a certas fases, notadamente a estrutura navegacional e projeto de interface. A coluna **Modelos e Notações** indica os modelos e notações usados nos passos do processo, referenciando, quando pertinente, os passos em que cada modelo/notação é usado. A coluna **Ferramentas de suporte** apresenta qual ferramenta de suporte à geração de modelos e/ou códigos de implementação existe associada à abordagem. Maiores detalhes podem ser obtidos consultando as referências fornecidas na coluna **Referências Bibliográficas**.

A tabela 3.3 identifica quais passos do ciclo de vida de desenvolvimento de aplicações Web são cobertos por cada abordagem.

	Engenharia de Requisitos	Espec. do Site	Design do Site			Implem.	Avaliação
			Conceitual	Naveg.	Apresent.		
HDM / HDM- lite							
RMM							
OOHDM							
JESSICA							
WAE							
WebML							

Tabela 3.3: Cobertura do ciclo de vida

3.4.3.3. Ferramentas

Uma simples busca na internet nos mostra que há hoje em dia uma grande quantidade de ferramentas com o objetivo de auxiliar a construção de novas aplicações Web ou até a migração de sistemas legados para a plataforma Web. Com este objetivo comum, todas permitem facilidades e um ganho considerável de esforço em relação ao desenvolvimento manual de sites. No entanto, um cuidadoso olhar sobre suas características revela que a maioria destas ferramentas concentra-se ainda sobre a fase de implementação do site, não considerando ou considerando apenas parcialmente as outras fases do ciclo de desenvolvimento Web (ver seção 3.4.2).

Os principais tipos de ferramentas para concepção de interfaces Web são:

- editores visuais HTML : editores WYSIWYG que permitem a construção de páginas HTML sofisticadas sem programação (p.ex. MicroSoft FrontPage);
- verificadores de links: que lêem um documento HTML, extraem os vínculos (links) de hipertexto e testam a validade dos URLs; estas ferramentas são tipicamente associadas à avaliação de sites e por isto estão descritas na seção 3.6;
- gerenciadores de sites: que permitem exibir graficamente o conteúdo de um site e suportam funções como remoção, renomeação, upload de páginas além de detecção e reparo de links "quebrados" (p.ex. VisualWeb, SiteAnalyst, **WebTracer** <http://www.nullpointer.co.uk/-/tracer.htm>, e **Mapuccino** <http://www.alphaworks.ibm.com/tech/mapuccino>) ;
- ferramentas de autoria de hipermídia estendidas para Web (*Web-enabled hypermedia authoring tool*): que foram inicialmente concebidas para desenvolvimento de aplicações hipermídia off-line (não conectadas à Web) e

que foram estendidas para suportar geração de aplicações Web (p.ex. Asymetrix Toolbook e o Macromedia Director);

- gateways Web-Bancos de Dados: que produzem páginas HTML dinamicamente a partir do conteúdo de um banco de dados, geralmente através de templates de páginas HTML embutindo consultas SQL, processadas dinamicamente por um script CGI; exemplos desta categoria incluem JavaSoft Java Server Pages (JSP), MicroSoft Active Server Pages (ASP) e Allaire Cold Fusion Web Database Construction Toolkit;
- editores de formulários Web, geradores de relatórios e assistentes de publicação de bancos de dados na Web (*Web database publishing wizards*): que auxiliam a migração de aplicações baseadas em formulários (forms) da arquitetura cliente-servidor centrada em bancos de dados para a Web e que suportam tarefas específicas associadas a esta migração; o assistente de publicação, por exemplo, permite a personalização de páginas geradas pela aplicação de modelos pré-definidos de apresentação aos objetos exportados de bancos de dados;
- geradores de aplicações Web orientados a modelos: que fornecem cobertura completa a todas as atividades de desenvolvimento, produzindo automaticamente sites compostos de páginas de formato fixo criadas a partir de modelos que descrevem a) o esquema do banco de dados, b) a definição das aplicações e suas unidades que vão manipular o banco de dados e c) as preferências dos usuários na forma de parâmetros (cores, cabeçalhos, imagens de fundo, rodapés, texto de ajuda, etc) que serão usados para direcionar a geração da apresentação das páginas. Exemplos incluem a ferramenta AutoWeb (Frernali, 2000) e Web Development Suite da Oracle, que contém inclusive a Designer 2000.

3.5. Métodos de avaliação de usabilidade

Nos últimos anos um grande número de métodos tradicionais de avaliação de usabilidade têm sido utilizados em projetos Web com pequenas adaptações, e outros têm sido desenvolvidos especificamente para este ambiente. Nesta seção são apresentados os principais métodos de avaliação de usabilidade aplicáveis a interfaces Web. Evidentemente esta lista não é exaustiva e outros métodos podem ser utilizados para a mesma finalidade.

3.5.1. Diversidade de métodos de avaliação

Os métodos de avaliação de usabilidade disponíveis podem ser classificados, em um primeiro momento, como métodos de inspeção de usabilidade e testes empíricos com a participação de usuários. Métodos de inspeção caracterizam-se por empregarem especialistas em interface que a utilizam em busca de possíveis problemas de usabilidade. Como exemplo cita-se a avaliação heurística. Os métodos com a participação de usuários caracterizam-se pelo uso de questionários ou observação direta ou indireta de usuários durante a utilização da interface, como fonte de informações que possam levar à identificação de problemas. Como exemplos destes métodos podem ser

citados ensaios de interação (ou teste com usuário), questionários e análise de arquivos de *log*, entre outros. Outros métodos que envolvem usuários como, por exemplo, *focus group* e classificação de cartões (*card sorting*), também são utilizados para descobrir como os usuários organizam as informações do domínio de problema, quais suas expectativas e necessidades com relação à interface.

3.5.2. Avaliação heurística

A avaliação heurística é um método tradicional de avaliação de usabilidade. Este método foi desenvolvido por Nielsen e Molich (Nielsen, 1993) e consiste da inspeção sistemática da interface do usuário com relação à sua usabilidade. O método foi utilizado pela primeira vez em um interface Web em 1994, num estudo para o web site da Sun Microsystems (Nielsen e Sano, 1995). Seu procedimento básico é o seguinte: um avaliador interage com a interface e julga a sua adequação comparando-a com princípios de usabilidade reconhecidos, as heurísticas. Nielsen sugere um conjunto com apenas 10 recomendações heurísticas para guiar a avaliação, enumeradas a seguir.

Diálogos Simples e Naturais: as interfaces de usuários devem ser o mais simples possível. Interfaces devem combinar as tarefas do usuário de forma a simplificar o mapeamento entre os conceitos computacionais e os do usuário. Deve-se apresentar exatamente a informação que o usuário precisa – nem mais nem menos - na hora e lugar exatos onde é necessária. Informação que será usada em conjunto deve ser exibida em conjunto, ao menos na mesma tela. Tanto os objetos de informação quanto as operações devem ser acessados em uma seqüência compatível com o modo pelo qual os usuários irão realizar suas tarefas efetiva e produtivamente. Muitas vezes tais seqüências são forçadas pela interface, mas normalmente é melhor permitir que o usuário controle o diálogo o máximo possível, de tal forma que a seqüência possa se ajustar às preferências do usuário.

Falar a Linguagem do Usuário: a terminologia da interface deve ser baseada na linguagem do usuário, e não orientada ao sistema. Para tanto, deve-se verificar quais termos são utilizados com maior freqüência pelos usuários. As informações também devem ser organizadas conforme o modelo mental que o usuário possui do domínio.

Minimizar a Sobrecarga de Memória do Usuário: o software deve exibir elementos de diálogo para o usuário e permitir que o mesmo faça suas escolhas, sem a necessidade de lembrar deste ou daquele comando específico. Para facilitar a utilização da interface, deve ser apresentado ao usuário um pequeno número de recomendações que se aplicam por toda a interface. Se o número de recomendações é grande o usuário terá de aprender/lembrar todas as recomendações, o que pode não ser tão simples. Por outro lado, se o software não tiver regra alguma, então o usuário deverá lembrar de cada elemento de diálogo. O uso de comandos genéricos é uma maneira de se ter um pequeno conjunto de recomendações. Comandos genéricos fazem com que coisas similares ocorram em diferentes circunstâncias, sendo suficiente ao usuário aprender poucos comandos para trabalhar com vários tipos de dados.

Consistência: consistência é um dos princípios básicos de usabilidade. Se os usuários souberem que um mesmo comando ou uma mesma ação terá sempre o mesmo efeito, eles ficarão mais confiantes no uso do software, e serão encorajados a fazerem novas descobertas. A mesma operação deverá ser apresentada na mesma localização em todas as telas e deverá ser formatada da mesma maneira para facilitar o reconhecimento.

Feedback: O sistema deverá informar continuamente ao usuário sobre o que ele está fazendo. O tempo de resposta influí no tipo de feedback que deve ser dado ao usuário. Um décimo de segundo (0,1s) é o limite para o usuário pensar que o sistema está reagindo instantaneamente, o que significa que nenhum feedback especial é necessário; um segundo (1,0s) é o limite para que o fluxo de pensamento do usuário não seja interrompido, mesmo que o usuário perceba uma certa demora; e dez segundos (10s) é o limite para manter a atenção do usuário focalizada no diálogo. Muitas vezes, feedbacks especiais são necessários para mostrar o andamento de uma tarefa ou contextualizar uma navegação mais demorada do usuário.

Saídas Claramente Marcadas: De modo a fazer com que o usuário sinta que pode controlar o software, deverá ser fácil sair das situações mais variadas possíveis. Por exemplo, todas as caixas de diálogo devem possuir um botão **Cancelar** para abortar uma tarefa. Muitas vezes, as saídas podem ser fornecidas por meio de uma facilidade de desfazer (undo) a última operação e retornar ao estado anterior. Os usuários rapidamente aprendem a confiar neste mecanismo e portanto ele deve estar disponível como um comando genérico por todo o software. Neste caso, o usuário poderá confiar no aprendizado por exploração, pois saberá desfazer eventuais erros.

Atalhos: embora deva ser possível operar a interface conhecendo-se apenas algumas regras gerais, deveria também ser possível para o usuário experiente executar mais rapidamente operações freqüentemente utilizadas, através de atalhos. Aceleradores típicos incluem abreviações, teclas de função, clique duplo do mouse, ou botões especiais para funções freqüentes. Também podem ser apresentados através da exibição dos últimos comandos executados, ou da função de volta (backtrack) em sistemas de hipertexto. Atalhos são também necessários quando por uma política de uma empresa ou organização a informação que se encontra em uma maior profundidade da árvore navegacional tenha que ser recuperada diretamente pela interface principal. Por exemplo, na página da Receita Federal, na época de declarar o IRPF, deve haver um *link* na primeira página para declaração on-line ou para *download* dos programas de declaração ao invés de obrigar o usuário a navegação pelo site até achar a página correta, o que muitas vezes não é fácil.

Boas mensagens de erro: as mensagens de erro devem seguir algumas regras: linguagem clara e sem códigos. Devem ser precisas. Devem ajudar o usuário a resolver o problema. Não devem intimidar ou culpar o usuário.

Prevenir Erros: melhor do que possuir boas mensagens, é evitar situações de erro. Conhecendo-se as situações que mais provocam erro, sempre é possível modificar a interface e tornar muito improvável que este erro ocorra.

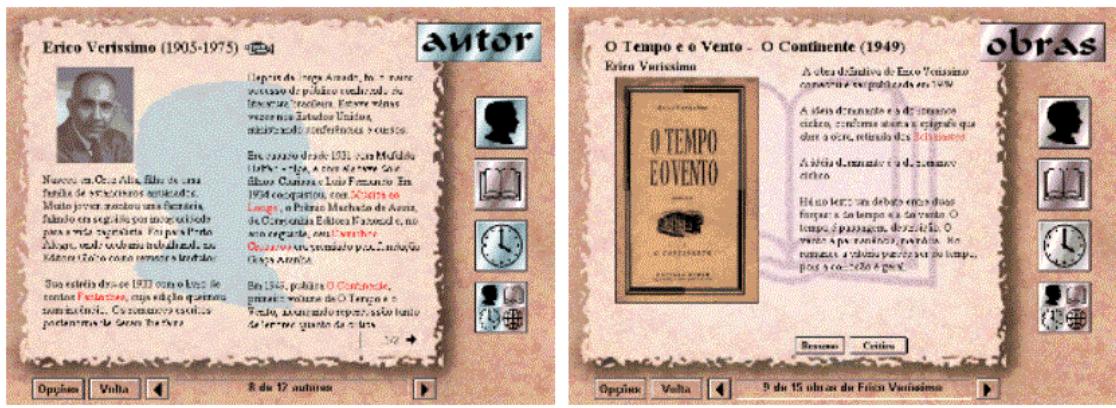
Ajuda e Documentação: o melhor é que um software que seja tão fácil de usar que não necessite de ajuda ou documentação. No entanto, se preciso, esta ajuda deve estar facilmente acessível on-line. Além disto, sabidamente usuários raramente lêem a documentação.

Como certamente um só avaliador não irá encontrar todos os problemas de uma interface, idealmente são utilizados vários. Nielsen sugere que a melhor relação custo/benefício é alcançada quando se utilizam entre 3 e 5 avaliadores. Cada avaliador deve realizar a sua inspeção individualmente e somente depois de todas avaliações terem sido concluídas, os avaliadores podem se comunicar. Este cuidado é importante para garantir avaliações independentes e sem influências. Os resultados tanto podem ser registrados por cada avaliador como por um observador presente durante as sessões, onde os avaliadores verbalizam seus comentários. Ao final de todas as sessões o observador deverá reunir todas as avaliações feitas em um único documento. Além disto, o observador poderá auxiliar o avaliador em caso de problemas com o protótipo, se este for o caso.

Tipicamente, uma sessão de avaliação heurística dura entre uma e duas horas. O resultado da avaliação é uma lista de problemas de usabilidade, indicando qual ou quais princípios foram violados e a gravidade do problema. Na tabela 3.4 são apresentados alguns dos problemas encontrados durante a avaliação do CD-ROM Literatura Gaúcha (veja figura 3.8), assim como comentários que ilustram como os avaliadores utilizaram as heurísticas para identificar problemas.

Descrição do Problema	Severidade [1 – 3]	Heurística violada e Comentários
Na página “Érico Veríssimo” há a opção de visualizar um vídeo sobre o autor, que é disponível pelo ícone de uma câmera filmadora (figura autor). Ao iniciar o vídeo não há como interromvê-lo.	3	<i>Fornecer saídas claramente marcadas e prevenir erros:</i> os avaliadores consideraram o problema muito grave (5) porque ele obriga o usuário a ver todo o vídeo mesmo que a sua ativação tenha sido ocasionada por um ‘click’ acidental. Esta é uma situação de erro que poderia ser minimizada com a opção para interromper o vídeo. Observa-se aqui um problema que foi classificado por mais de uma heurística, o que é aceitável na identificação de problemas complexos como este.
Na obra “O tempo e o Vento” o botão [crítica] (base da figura Obra) é disponível mas, uma vez selecionado, não mostra o texto correspondente à crítica do livro, como esperado.	2	<i>Fornecer Mensagens de Retorno Adequada:</i> Sabendo que mensagens adequadas são necessárias para a eficiente utilização da interface, o avaliador inspeciona a interface em busca de todas as situações onde boas mensagens de retorno não são fornecidas. Assim, o avaliador identificou este problema e atribuiu severidade moderada (3) supondo que os usuários ficariam desorientado sobre a função de crítica.
Não existem opções de atalho.	1	<i>Fornecer atalhos:</i> o avaliador usou esta heurística para procurar situações que necessitassem de atalhos. Contudo, foi considerado que atalhos não são necessários nesta aplicação e por isto uma severidade baixa (1) foi atribuída. Esta heurística lembra o avaliador de considerar as situações onde atalhos são necessários.

Tabela 3.4: Exemplos de problemas de usabilidade



Autor - Érico Veríssimo.

Obra – O Tempo e o Vento.

Figura 3.8: CD-ROM Literatura Gaúcha

O método de avaliação heurística pode ser empregado sem modificações em projetos Web. Contudo, recentemente Nielsen e outros autores (Bailey, 1999) tem sugerido outros conjuntos de recomendações que podem ser utilizados em substituição das heurísticas clássicas de Nielsen. A avaliação heurística é um método simples, cuja eficiência reside na capacidade dos avaliadores de reconhecer problemas de usabilidade. Em princípio, qualquer pessoa pode ser treinada para a aplicação deste método, embora melhores resultados sejam obtidos com avaliadores experientes. Outra vantagem deste método é que ele pode ser aplicado em qualquer etapa do desenvolvimento, mesmo sobre protótipos em papel.

3.5.3. Ensaios de interação

Neste tipo de avaliação, usuários participam realizando algumas tarefas com a interface enquanto são observados por avaliadores em um laboratório de usabilidade (Rubin, 1994). Os laboratórios de usabilidade são salas equipadas com câmeras para filmagem do teste e espelhos falsos que permitem aos avaliadores observar os usuários sem serem visto. Contudo, tais testes também podem ser realizados sem a necessidade de laboratórios sofisticados, utilizando-se apenas uma câmera filmadora convencional ou mesmo um gravador somente para áudio. Se duas câmeras são disponíveis, uma deve focalizar o rosto do usuário enquanto a segunda registra a tela do computador. Com uma única câmera é preferível focalizar a tela do computador para registrar o que o usuário está fazendo. A revisão da gravação é ao mesmo tempo uma forma de registro e um meio para a equipe discutir os problemas de usabilidade. Porém, estes equipamentos podem ser dispensados se o avaliador dispõem de outro meio de registro dos problemas (por exemplo, ferramentas como LotusCam ou até mesmo papel e caneta). A Figura 3.9 apresenta um ensaio de interação no laboratório de usabilidade da UFSC, o LabiUtil⁸.

Sempre que possível deve-se selecionar para o teste usuários reais da interface. Em alguns casos, se os usuários escolhidos não forem representativos, todo o teste pode falhar na identificação de problemas de usabilidade. Contudo, pode ser difícil localizar

⁸ Imagem reproduzida com a permissão do LabiUtil (Universidade Federal de Santa Catarina – UFSC, Florianópolis; <http://www.labutil.inf.ufsc.br/>).

usuários reais de interfaces Web. Ainda que usuários reais possam ser localizados, a distribuição geográfica pode inviabilizar a sua participação.

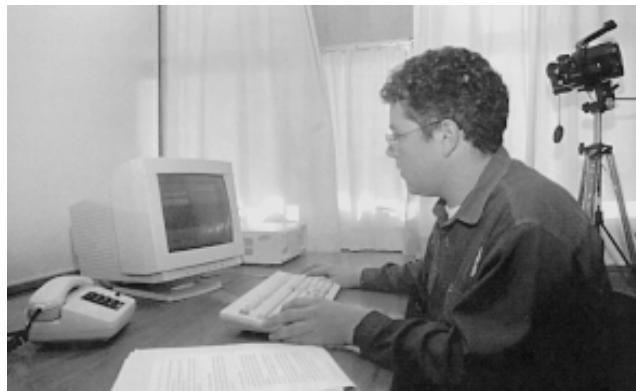


Figura 3.9: Ensaio de Interação em um laboratório de usabilidade

Um grande número de usuários é desejável mas, por questões de custo e tempo, tem se adotado um número reduzido em cada ciclo como forma de viabilizar a avaliação de interfaces Web. (Nielsen, 1993) sugere que com 5 usuários pode-se identificar aproximadamente 70% dos problemas mais críticos da interface. Isto caracteriza uma situação adequada para a maioria dos projetos, embora 5 seja um número relativo para garantir a eficiência do método (Woolrych e Cockton, 2001).

Durante o teste os usuários poderão ser solicitados a realizar tarefas que são pré-definidas (cenários de teste) pelo avaliador, responderão algumas perguntas ou simplesmente utilizarão livremente a interface. Se utilizado, o conjunto de tarefas deve ser previamente definido e passado ao usuário no início do teste. Como por exemplo: “*Procure no site da UFRGS o endereço postal do Instituto de Informática.*”, “*Qual o nome do reitor da UFRGS?*”.

Durante a sessão, o usuário deve ser instruído para dizer tudo o que está pensando e fazendo. Como esta não é uma atitude muito natural para a maioria das pessoas, o avaliador precisa estimular o usuário com perguntas como, por exemplo, “*O que você está pensando?*” ou “*Tem alguma coisa na interface que você não gosta?*”. Esta forma de diálogo conhecida como “*Thinking Aloud Protocol*”, ou pensamento em voz alta, induz o usuário a verbalizar seus pensamentos de modo a captar suas opiniões. Algum treinamento é necessário para conduzir tais testes satisfatoriamente, que fica geralmente a cargo de especialistas em ergonomia. Basicamente, problemas de usabilidade são identificados quando se observa usuários com dificuldades para completar suas tarefas.

Existem algumas questões éticas em torno de teste com usuários. Primeiramente, quem está sendo testado é a interface e não o usuário. Isto deve ser norma de conduta do avaliador e também ser informado claramente ao usuário. Ainda que o usuário saiba disto, a sessão de teste geralmente é estressante para ele e é dever do avaliador criar um clima agradável para deixá-lo à vontade. A idéia de participar de um teste pode amedrontar algumas pessoas e, por isto, usuários não devem ser obrigados a participar. Do mesmo modo, se um usuário desejar interromper o teste por qualquer motivo o avaliador não deve insistir no contrário.

Ensaios de interação costumam ter custo maior que o método de avaliação heurística e, por este motivo, recomenda-se que seja empregado a partir de protótipos

funcionais. Este método não substitui a avaliação heurística e melhores resultados são obtidos quando ambas as técnicas são empregadas em conjunto.

3.5.4. Inspeção de recomendações ergonômicas (*guidelines* e *checklist*)

Boa parte do conhecimento sobre usabilidade tem sido sistematizado sob a forma de *guidelines* ou conjuntos de recomendações ergonômicas. A construção desses *guidelines* é resultado de pesquisas nas áreas de ciência cognitiva, psicologia e ergonomia. Em alguns casos, trata-se de conhecimento prático que foi acumulado durante o desenvolvimento de vários projetos ou ainda, recomendações de “bom senso”. Exemplos de recomendações foram apresentados na seção 3.4.3.1. De um modo geral não existe um padrão de descrição de recomendações ergonômicas, que podem ter um escopo amplo (que exige interpretação por parte do avaliador) ou específico (aplicáveis diretamente ao projeto). Como exemplos de tais recomendações cita-se duas:

- *Páginas iniciais, que suportam navegação, que devam ser lidas rapidamente ou, que contenham gráficos grandes, devem ser curtas. Usar páginas longas para simplificar a manutenção do site e tornar as páginas mais fáceis de imprimir.* Fonte: (Lynch e Hortson, 1999)
- *Selecione cuidadosamente os títulos de páginas. Use nomes que estejam relacionados com o conteúdo ou função da página.* Fonte: (Nielsen, 1999)

Recomendações ergonômicas podem ser usadas com o duplo propósito de auxiliar o processo de concepção ou guiar a avaliação. Durante o processo de concepção, designers devem consultar tais recomendações que auxiliam evitar problemas de usabilidade. Por outro lado, pode-se utilizar o conjunto de recomendações ergonômicas como suporte para a inspeção da interface; neste caso, a interface é inspecionada minuciosamente por um avaliador que verifica se todas as recomendações ergonômicas são respeitadas.

A avaliação de usabilidade usando *guidelines* consiste basicamente em ter um ou mais avaliadores que investigam a interface e, quando um problema é identificado este é associado à uma ou mais recomendações que foram violadas. O processo é simples mas exige uma grande experiência do avaliador, pois o conjunto de recomendações ultrapassa facilmente algumas dezenas a considerar. Algumas recomendações não são facilmente aplicadas e exigem a interpretação por parte do avaliador. Por estes motivos, o processo de avaliação usando recomendações ergonômicas pode ser bastante exaustivo se o avaliador não tiver conhecimentos prévios sobre as recomendações.

Uma solução alternativa para este problema é utilizar um *checklist*, que nada mais é do que um conjunto mínimo de recomendações diretamente aplicáveis ao projeto, que em geral, que não necessitem um grande esforço de interpretação. Geralmente *checklists* focalizam alguns aspectos considerados importantes da interface e que, potencialmente, podem hospedar os problemas mais graves de usabilidade. A tabela 3.5 apresenta um *checklist* para avaliação de sites Web de vendas online.

Checklists facilitam a análise de recomendações ergonômicas durante a avaliação de usabilidade. Este tipo de inspeção pode ser particularmente interessante quando se deseja realizar avaliações rápidas de usabilidade, investigar a consistência da

interface e verificar mudanças ocasionadas pela manutenção do site. Trata-se de um tipo de inspeção de custo relativamente baixo que pode ser adaptado às diversas situações de avaliação, bastando para tanto selecionar as recomendações ergonômicas adequadas.

Navegação do site		
	Sim	Não
	Obs.	
A navegação atual é sempre clara (“Onde eu estou?”)?		
Todas as páginas tem um links para a página inicial?		
A estrutura do site é simples?		
Nomes técnicos e jargões são evitados?		
Nenhum recurso ou plug-ins desnecessário é utilizado?		
Páginas são menores do que 50 Kb ?(para minimizar o tempo d download)		
Encontrando produtos		
	Sim	Não
	Obs.	
Clientes podem procurar produtas de diferentes maneiras?		
Existe uma ferramenta de busca fácil de utilizar para procurar produtos?		
Produtos são descritos adequadamente?		
Produtos são classificados claramente?		
Comprando		
	Sim	Não
	Obs.	
O processo de compra é claro e simples?		
São aceitos cartões de crédito?		
Pagamentos com cartão de créditos usam mecanismos de segurança (SSL)?		
A descrição da política de vendas é clara e fácil de encontrar no site?		
O procedimento para devolução de produtos é claro e fácil de encontrar no site?		
Clientes podem pagar por telefone?		
A empresa fornece informações claras sobre a entrega de produtos?		
A empresa tem suporte para clientes fora da área metropolitana?		
Clientes podem abandonar facilmente um processo de compra?		
Clientes podem cancelar facilmente uma ordem de compra?		
Serviço ao cliente		
	Sim	Não
	Obs.	
O suporte pode ser feito por mail?		
O suporte pode ser feito por telefone e o número é fácil de encontrar no site?		
Todos os links gráficos são também disponíveis como links de texto (para clientes com deficiência visual)		
Todas as imagens tem uma ALT tag (para clientes com deficiência visual)		
Prevenção e recuperação de erros		
	Sim	Não
	Obs.	
Erros não ocorrem facilmente.		
Mensagens de erro são claras e úteis.		
Aspecto visual		
	Sim	Não
	Obs.	
O layout é claro?		
Animações desnecessárias são evitadas?		
O aspecto visual é agradável?		
As páginas são legíveis?		

Tabela 3.5: Checklist para avaliação de usabilidade para um sistema de vendas online
 (fonte: Information & Design <http://www.infodesign.com.au>)

Contudo, tal avaliação tende a ser restrita aos itens que fazem parte da avaliação. De uma maneira geral, *guidelines* completas contam com centena de recomendações (ver Vanderdonckt, 1994; ISO, 1999), o que pode tornar o processo de inspeção tedioso. A inspecção de *guidelines* deve, portanto ser utilizado com critério e não substitui o uso

de outros métodos de avaliações com usuários. A inspeção com o uso de *guidelines* é adequada, durante a construção de protótipos funcionais, uma vez que a maioria das recomendações é endereçada a objetos de interação.

3.5.5. Questionários

Questionários são ferramentas muito úteis na avaliação da interação entre o usuário e a interface. São utilizados para coletar informações subjetivas sobre dados sobre o perfil dos usuários, a qualidade da interface e quais problemas são encontrados no seu uso. Estas informações são tão importantes quanto a performance no uso do sistema, e não podem ser obtidas de outra forma senão perguntando aos usuários. O uso de questionários dá ao avaliador a vantagem de aplicar vários testes ao mesmo tempo em locais diferentes. Questionários podem ser úteis de diferentes maneiras dentro do desenvolvimento de interfaces Web como, por exemplo, para:

- Identificar o perfil dos usuários. O objetivo deste tipo de questionário é basicamente coletar informações sobre os usuários. Tais informações podem ser de origem funcional, pessoal, sobre preferências ou mesmo sobre a utilização de computadores e sistemas. Obviamente outras questões podem ser adicionadas para fins específicos de avaliação.
- Determinar o grau de satisfação dos usuários com relação a interface. Questionários específicos para descobrir a satisfação de usuários vêm sendo pesquisados desde a década de 80, e uma versão específica para sites Web tem sido desenvolvida sob o nome de WAMMI (disponível por <http://www.wammi.com/>).
- Estruturar informações sobre problemas de usabilidade identificados por usuários, na forma de questionários para relato de incidentes críticos (ver seção 3.5.6).

Pode-se ainda utilizar questionários para estruturar avaliações com usuário (ver seção 3.5.3), onde o usuário deve utilizar a interface de modo a poder responder as questões. Neste caso, as perguntas devem ser definidas antes do teste e adaptadas a cada contexto de avaliação.

Uma das grandes vantagens de questionários é a possibilidade de aplicá-los à um grande número de usuários ao mesmo tempo, utilizando o próprio ambiente Web através de formulários eletrônicos. Contudo, deve-se salientar que os resultados exigem um grande esforço de interpretação para identificar problemas de usabilidade. Os questionários para avaliar a satisfação dos usuários são interessantes do ponto de vista de marketing, mas na maioria dos casos não explicam os resultados obtidos como por exemplo “por que razão os usuários gostam do site” ou “o que deve ser mudado para melhorar a interface”. Assim, questionários de satisfação devem sempre ser acompanhados de algum outro método de avaliação que possa explicar as respostas subjetivas dos usuários.

3.5.6. Relatos de incidentes críticos por usuários

Relato de incidentes críticos (IC) consiste simplesmente na análise de descrições de ações e eventos que refletem problemas de usabilidade, ausência de funções que deveriam estar na interface, ou outros problemas que interferem na interação do usuário com a interface. Geralmente, tais descrições são realizadas pelos próprios usuários que relatam ICs enquanto trabalham em suas tarefas diárias. São descrições espontâneas que não ocorrem dentro de um contexto de avaliação e, portanto, são normalmente informais e pouco estruturadas.

Webmasters e a equipe de manutenção do site são os alvos mais visados das críticas e relatos de problemas encontrados no site. Observa-se, no entanto, que muitas equipes de desenvolvimento negligenciam este tipo de informação. Existem várias razões que poderiam justificar o descaso dos *webmasters*. Uma delas é o fato de que tais profissionais nem sempre são as mesmas pessoas responsáveis pelo design ou avaliação do site, e portanto, não estão aptas a interpretar tais dados como indícios de que a usabilidade do site pode estar sendo comprometida.

Alguns estudos experimentais realizados por (Castilho, 1998) sugerem que alguns fatores contextuais podem auxiliar na interpretação dos relatos. Estes fatores são:

- início e fim do IC;
- tarefa e objetivo do usuário antes do IC;
- expectativa do usuário antes do IC;
- freqüência do IC;
- como evitar um IC;
- sugestões do usuário para resolver o IC.

Um suporte mínimo pode ser oferecido aos usuários de modo a lhes facilitar a tarefa de descrever ICs. Este suporte pode ser feito na forma de um treinamento rápido que visa ensinar usuários a reconhecer um IC e a descrevê-lo. Este tipo de treinamento é recomendado especialmente no caso de Intranet ou quando usuários representam um público cativo, como no caso de cursos de treinamento a distância, por exemplo. Outra solução é oferecer questionários que permitam aos usuários descrever o cenário de ocorrência do problema.

Os principais problemas com esta técnica são a qualidade da descrição e confiabilidade nos relatos. Além disto, o sucesso depende da boa vontade de usuários em relatar os problemas que encontram na interface. Esta técnica é útil, tendo um custo menor que testes com usuários, embora a descrição e confiabilidade dos achados seja bem inferior. Este método de análise deve ser aplicado durante todo o período de utilização da interface por usuários.

3.5.7. Análise de logs

Neste método são analisadas as interações do usuário através de arquivos de *logs* gerados durante a utilização do sistema. Análise de arquivos de *logs* é uma técnica que pouco interfere na realização das tarefas, preservando assim porções do contexto do trabalho real. A tarefa de coleta de dados é relativamente barata, não exige a

participação do avaliador durante o teste e guarda todas as ações do usuário, o que é de difícil aquisição em outras técnicas.

Em princípio este parece ser o método de avaliação ideal para o ambiente Web, uma vez que os servidores Web, que fornece as páginas ao cliente na Web, mantém um arquivo de *log* das requisições dos clientes. Basicamente são registrados pelo servidor:

- o IP da máquina utilizada pelo cliente;
- data e horário da solicitação;
- nome do arquivo solicitado;
- identificação do *browser* cliente;
- identificação do sistema operacional do cliente;
- página onde o usuário selecionou o *link*; e,
- erros de solicitação de arquivo.

Os dois maiores problemas, porém, são identificar que tipos de dados são úteis e a forma de analisá-los. No trabalho realizado por (Siochi e Ehrich, 1991), na análise de repetição de padrões em arquivos de *log*, foram encontradas ações que sugeriram problemas de usabilidade; porém, alguns somente foram identificados ou confirmados quando os usuários explicaram o que estavam tentando fazer com aquelas ações. No caso da Web, os registros em arquivos de log mantidos pelos servidores não são suficientes para analisar toda a interação do usuários. Isto ocorre quando usuários usam *browsers off line*, com *cache* local ou servidores *proxy*, que mascaram as requisições evitando o acesso direto ao servidor. As ferramentas mais frequentes para análise de *logs* apenas realizam estatísticas de acesso das páginas mais visitadas e, em alguns casos, a origem e os usuários mais frequentes. Entretanto estas informações não dizem muito sobre a usabilidade.

Contudo, uma simples análise dos erros armazenados nos arquivos de *log*, pode ser útil na identificação de problemas relacionados a páginas não encontradas devido a links inválidos (Sullivan, 1997). Outra forma de utilizar eficientemente arquivos de *log* é analisar as palavras-chaves informadas pelos usuários às ferramentas de busca, pois elas indicam que tipo de informação os usuários estão procurando e podem sugerir como facilitar o acesso a elas.

3.6. Ferramentas para avaliação automática

Nesta seção é introduzida a idéia de avaliação automática e são descritas algumas ferramentas existentes para auxiliar esta tarefa.

3.6.1. Inspeção automática

No intuito de minimizar o esforço de avaliação de interfaces, várias ferramentas de software para inspeção automática de interfaces Web têm sido desenvolvidas nos últimos anos. As primeiras ferramentas criadas verificavam apenas a consistência do código HTML mas atualmente algumas são também capazes de verificar recomendações ergonômicas, identificar problemas de usabilidade e realizar correções

automaticamente. Tais ferramentas se tornaram bastante populares pois não exigem que a pessoa que as utiliza seja um especialista para identificar problemas de usabilidade.

O processo de avaliação com o auxílio de tais ferramentas é semelhante ao método de inspeção de recomendações ergonômicas. Um programa lê as páginas do site e tenta identificar problemas de usabilidade que podem ser extraídos a partir do código HTML. São alguns exemplos de problemas que podem ser identificados por tais ferramentas:

- Imagens que não possuem um texto alternativo (tag ALT) para usuários que não carregam imagens ou são deficientes visuais;
- A página contém mais de um link com o mesmo nome que ligam páginas diferentes;
- A cor de fundo da página e a cor do texto tem pouco contraste tornando difícil a leitura;
- A página do site não contém nenhum link apontando para ela;
- O link aponta para uma página que não existe;
- A imagem informada no documento não existe no endereço especificado ou não pode ser apresentada.

Algumas ferramentas limitam-se a identificação de problemas, outras conseguem corrigir alguns dos erros encontrados. Contudo deve-se salientar que são poucos os tipos de problemas que podem ser identificados automaticamente sem a interpretação de um avaliador. Na maioria dos casos tais ferramentas identificam situações que sugerem a existência de problemas e chamam a atenção da pessoa que as utiliza, o qual deve interpretar o alerta e determinar se é realmente um problema de usabilidade. Portanto, o uso de tais ferramentas **não** resolve todos os problemas de usabilidade nem elimina a necessidade de conhecer princípios básicos de concepção de interfaces ergonômicas, contudo elas facilitam o processo de inspeção que pode ser tedioso quando realizado manualmente sobre grandes sites.

3.6.2. Funcionalidades de ferramentas de avaliação automática

Podemos distinguir 3 categorias de inspeção: a) verificação do código HTML/CSS; b) verificação de recomendações de acessibilidade; c) verificação de recomendações ergonômicas.

Ferramentas que verificam a conformidade do código HTML/CSS não se preocupam com outros aspectos ligados a usabilidade. Neste caso, a verificação é restrita a comparação do código da página com o padrão definido para a linguagem. Tais ferramentas são interpretadores de código, semelhantes aqueles que vem junto com os compiladores de algumas linguagens de programação. Tais interpretadores são capazes de identificar os marcadores <tags> utilizados e de determinar se eles fazem parte da recomendação W3C (a organização que define os padrões para a web). Marcadores que não seguem o padrão podem funcionar em alguns *browsers*, mas devem ser evitados pois podem comprometer a usabilidade da interface se os usuários dispõem de um *browser* diferente.

Para exemplificar o uso catastrófico de marcadores não padronizados a figura 3.10 apresenta o resultado de uma página que utiliza tais marcadores em dois *browsers*

diferentes (Internet Explorer e Netscape). Como pode-se observar, a página é comprehensível quando exibida pelo Internet Explorer mas é ilegível no Netscape. Salienta-se que todos os fabricantes de *browser* estão cientes dos padrões W3C e implementam todas funcionalidades previstas. Seguir as recomendações W3C é uma maneira que o desenvolvedor tem de se assegurar que o conteúdo das páginas Web será exibido corretamente por todos os *browsers*.



Figura 3.10: Problema de apresentação devido ao uso de marcadores não-padrонizados (<http://www.terra.com.br/istoe/>)

A segunda categoria de funcionalidade oferecida diz respeito a verificação de recomendações de acessibilidade, de acordo com a definição da Iniciativa para Acessibilidade da Web (WAI <http://www.w3.org/WAI/>). Como mencionado anteriormente na seção 3.2.2, acessibilidade (*accessibility*) implica em tornar utilizável a interface por qualquer pessoa, independente de sua deficiência física, sensorial, cognitiva, condição de trabalho ou barreiras tecnológicas. Contudo, observa-se que tais recomendações não beneficiam apenas os usuários com necessidades especiais mas usuários em gerais, assim muitas ferramentas utilizam o conjunto de recomendações desenvolvido dentro da WAI como base para a inspeção.

A terceira categoria de funcionalidade inclui a inspeção de recomendações ergonômicas. Na maioria dos casos, tais recomendações não são padronizadas e são definidas pelos próprios autores das ferramentas. Tais recomendações podem ou não ser baseadas em estudos científicos que justificam a sua aplicação. Tais ferramentas podem incluir outras verificações que não são definidas formalmente como acessibilidade ou usabilidade mas podem ser úteis ao desenvolvedor tais como a verificação ortográfica do texto, contadores de palavras e compatibilidade com indexadores de páginas que facilitam a recuperação por ferramentas de busca.

Nas duas últimas categorias de ferramentas ocorre o fenômeno de interpretação da recomendação. Isto ocorre porque as recomendações de usabilidade e acessibilidade não foram necessariamente escritas para serem avaliadas automaticamente por ferramentas. Para automatizar o processo, os desenvolvedores devem traduzir a recomendação de modo a eliminar a ambigüidade contida no texto. Assim, a recomendação é interpretada pelo desenvolvedor da ferramenta que a traduz na implementação, o que explica em parte porque os resultados variam de um ferramenta para outra. Por exemplo, considere o simples exemplo de contraste de cores. A recomendação diz que a cor de fundo deve contrastar com a cor do texto mas não

informa nada sobre os valores aceitáveis de contraste. Ao implementar esta recomendação, a ferramenta armazena uma tabela de valores aceitáveis, e tal tabela não é necessariamente a mesma para todas as ferramentas.

Constata-se que a capacidade de identificação das ferramentas de avaliação automática é fortemente ligada à forma de especificação da recomendação utilizada. Para explicar melhor isto apresentamos uma segunda categoria de recomendações e seus níveis possíveis de automatização:

- Inspeção automática da recomendação: todos os elementos para a automatização são descritos na recomendação, sem ambigüidade. Ex. “Um link não deve apontar para uma página que não existe”.
- Inspeção automática com a interpretação da recomendação. Ex. “Usar cores contrastantes para o texto e fundo da página”, esta recomendação pode ser automatizada se uma tabela de valores aceitáveis para cores for fornecida;
- Inspeção parcial da recomendação, falta informação para identificar o problema mas alertas podem ser dados. Ex. “O nome de um link é completamente diferente do título da página destino”, neste caso a ferramenta não pode determinar se trata-se de um link que liga uma página equivocadamente ou não, um alerta pode ser dado para o avaliador.
- A recomendação é abstrata e não pode ser verificada automaticamente. Ex. “Crie um contexto de utilização para a interface”.

3.6.3. Ferramentas para avaliação automática

De um modo geral, tais ferramentas oferecem recursos bastante limitados de avaliação e não são capazes de identificar mais do que 35% dos problemas de usabilidade, no melhor de todos os casos. Contudo, a sua utilização é simples e pode realmente auxiliar na identificação de problemas importantes tais como a ocorrência de links inválidos, uso de cores com contraste adequado e existência de texto alternativo para imagens. Algumas ferramentas implementam mais do que uma destas funcionalidades. Outras ferramentas não apenas oferecem recursos para a verificação mas são capazes de corrigir automaticamente alguns dos problemas identificados.

Salienta-se que em geral as avaliações são realizadas pelas ferramentas baseadas no código HTML e são puramente sintáticas. Apenas algumas ferramentas são capazes de avaliar recomendações ergonômicas. Ferramentas podem ser usadas para auxiliar o processo de avaliação de usabilidade, mas até o momento nenhuma das ferramentas desenvolvidas é capaz de substituir o uso de métodos não-automáticos de avaliação. A seguir são apresentadas algumas ferramentas (e informações de acesso) para avaliação.

Bobby - <http://www.cast.org/bobby>

Desenvolvido pela CAST, ele suporta inspeção automática e manual de recomendações de usabilidades incluindo alertas. Analisa a compatibilidade entre vários browsers. Pode ser usado como ferramenta online ou instalado no servidor. A versão para download é escrita em Java. As recomendações verificadas incluem o pacote WAI entre outras.

Doctor HTML - <http://www2.imagiware.com/RxHTML/>

Desenvolvido pela Thomas Tongue e Imagiware, ele realiza verificações básicas de acessibilidade tais como verificação da tag "alt" em imagens e links válidos. Ele também verifica erros de sintaxe e correção ortografia do texto. Pode usar uma versão online ou comprar ou comprar uma autorização para instala-lo no servidor.

Dr. Watson - <http://watson.addy.com/>

Dr. Watson é um serviço gratuito oferecido pela Addy & Associates. Esta ferramenta verifica o código HTML 3.2, assim como extensões Netscape e Microsoft de HTML 4.version 4.x. Watson também pode verificar a validade de links, tempo de download de páginas, compatibilidade com ferramentas de busca, popularidade do link, número de palavras no texto e correção ortográfica. Nenhum verificação automática da acessibilidade é verificada. Apenas a versão online é disponível.

Lift - <http://www.usablenet.com/>

Lift Onsite e Lift Site são duas ferramentas desenvolvidas por UsableNet, Inc.

- LIFT Onsite permite web designers testar e corrigir problemas de acessibilidade e de usabilidade em páginas web incluindo problemas de navegação, velocidade de carga da pagina, qualidade das imagens utilizadas, etc. Ele integra editores web tais como Dreamweaver, GoLive, FrontPage, BB Edit. Esta ferramenta executa localmente sobre MacOS.
- LIFT Online é uma parte do LIFT Onsite e funciona num esquema de pre-inscrição no site.

Netmechanic - <http://www.netmechanic.com/>

Netmechanic contém um pacote completo de ferramentas de verificação que inclui inspeção do código, otimização de imagens, velocidade de conexão e monitoração de acesso do servidor.

WebSAT - <http://zing.ncsl.nist.gov/WebTools/WebSAT/overview.html>

WebSat faz parte de um pacote de 4 ferramentas para desenvolvimento de interfaces Web com usabilidade. WebSite é a ferramenta que realiza a inspeção de recomendações de acessibilidade de paginas web, navegação, legibilidade e tempo de carga do site. Assim como Bobby, esta ferramenta é capaz de dar alerta ao avaliador para porções do site que requerem a interpretação da regra. Pode ser utilizada online ou instalado localmente em plataformas Unix ou Windows 95/NT.

PageScreamer - <http://www.crunchy.com/tools/PageScreamer.html>

Desenvolvido pela Crunchy Technologies. Verifica e corrige conteúdo web fornecendo equivalência para elemento de texto. Verifica tabelas HTML, frames para navegação. Executa sobre Windows 9x, Windows NT, Windows 2000, Linux, Sun Solaris.

Page Valet - <http://valet.webthing.com/page/>

Page Valet combina verificação formal e validação de acessibilidade baseadas nas recomendações WAI.

W3C CSS validator - <http://jigsaw.w3.org/css-validator/>

Esta ferramenta verifica documentos web que utilizam CSS. Pode ser utilizada online ou instalada localmente.

W3C HTML validation service - <http://validator.w3.org/>

É um serviço de verificação do código HTML de acordo com as normas W3C HTML. O serviço é disponível online.

A-Prompt - <http://aprompt.snow.utoronto.ca/>

Desenvolvido pela universidade de Toronto. É gratuito e pode ser utilizado para inspecionar a interface e corrigir problemas identificados.

Colorfield Insight - <http://www.colorfield.com/insight.html>

Permite aos designer predizer a legibilidade de uma imagem para deficientes visuais. Desenvolvido pela Colorfield Digital Media.

Para uma lista completa de ferramentas disponíveis verifique o endereço: <http://www.w3.org/WAI/ER/existingtools.html>.

3.7. Discussão e considerações finais

Até aqui foram descritos vários conceitos e métodos de avaliação de usabilidade contudo pouco foi dito sobre como e quando utilizá-los. Métodos de avaliação de usabilidade são auxílios que devem ser considerados dentro do processo de desenvolvimento de aplicações e não como uma etapa isolada. Deve-se lembrar sempre que o objetivo principal de uma avaliação é melhorar a interface e não apenas estimar o quanto uma interface é boa ou ruim. Pode-se dizer que uma boa avaliação de usabilidade não é aquela que apenas identifica os problemas de usabilidade mas que auxilia a equipe de desenvolvimento a solucioná-los e a melhorar a interação do usuário com a aplicação.

Avaliação de usabilidade exige preparação. Alguns podem argumentar que apenas uma inspeção baseada no seu próprio “feeling” é suficiente para determinar a usabilidade de um site Web. Porém, não duvide que os resultados de tal avaliação serão

tão duvidosos quanto os critérios que foram utilizados. Na maioria dos casos os critérios expressam apenas uma opinião pessoal do avaliador e não a realidade dos usuários.

Vários aspectos contam para o sucesso de uma avaliação, entre os principais incluem-se a escolha do método adequado a cada situação de avaliação, a aplicação do método nas etapas adequadas de desenvolvimento, a documentação da avaliação e a formação contínua dos avaliadores. Esta seção visa justamente apresentar algumas recomendações e reflexões que podem auxiliar a considerar tais aspectos.

O processo de desenvolvimento de interfaces Web é um ciclo de etapas sucessivas de prototipação e avaliação. Interfaces Web estão sempre sendo atualizadas e cada alteração pode interferir na usabilidade. Na maioria dos casos os projetos Web dispõem de recursos limitados para realizar avaliação de usabilidade tornando proibitivo avaliação a cada atualização do site. Assim, torna-se vital escolher o melhor momento para realizá-las.

Observa-se que nas etapas em que se tem protótipos avançados e durante o uso de interfaces por usuários pode-se optar entre diferentes métodos de avaliações. Deve-se prestar atenção à etapa de análise de requisitos na qual uma série de métodos de coleta de informação devem ser empregados. Várias técnicas, como por exemplo reunião com usuários (*focus group*)(Nielsen, 1997) e classificação de cartões (Gaffney, 2000) podem evitar alguns de problemas de usabilidade. Tais técnicas não foram descritas aqui pois estão além do escopo deste capítulo mas não devem ser menosprezados.

Além das etapas sugeridas acima para avaliação, alguns eventos podem exigir avaliações adicionais. Exemplos de tais eventos incluem:

- Mudanças na estrutura ou organização do espaço de informação do site;
- Inclusão ou mudança de tecnologia da interface;
- Mudança do design gráfico;
- Inclusão de novos módulos ou novas tarefas suportadas pela aplicação.

Além destes eventos, outros sinais de alerta, no mínimo, exigem a atenção dos desenvolvedores:

- Redução visível no número de acesso nos arquivos de logs;
- Registros freqüentes de erros nos arquivos de log;
- Comentários dos usuários incluindo reclamações ou solicitações de ajudas para encontrar informações no site.

É imprescindível documentar bem todo o processo de avaliação. A documentação do projeto pode variar de projeto à projeto de acordo com tempo e recursos disponíveis para esta tarefa e o escopo da avaliação. Aqui são apresentados um conjunto mínimo que pode ser seguido (ver anexos A, B e C).

O primeiro documento comprehende a descrição dos objetivos e requisitos para a avaliação (anexo A). Nele definem-se as responsabilidade técnicas e administrativas para a boa condução dos testes. Quanto à responsabilidade técnica incluem-se itens que descrevem os objetivos gerais da avaliação como por exemplo, verificar se pelo menos 50% dos usuários do site são capazes de utilizar uma interface para fazer compras pela internet, ou determinar qual o percentual dos usuários satisfeitos com a mudança do design gráfico. Este tipo de definição auxiliará na escolha do método mais adequado

para avaliação. Então, deve-se especificar plataforma a ser utilizada e recursos necessários para execução da interface. As responsabilidades administrativas incluem a definição dos papéis e responsabilidade das pessoas participantes (o que é especialmente importante em grande projetos de avaliação), orçamento disponível, prazos, etc.

Durante a avaliação, cada seção de teste deve ser acompanhada de uma descrição do método utilizado, usuário, avaliador responsável e problemas identificados em cada sessão de teste (ver anexo B). Este documento serve de base para a geração de um relatório geral que resume os problemas encontrados durante todo o processo de avaliação (ver anexo C). Neste relatório geral deve-se eliminar as redundâncias de problemas identificados por mais de um usuário ou por mais de um teste; o importante é descrever o problema e o impacto geral para a aplicação. Sempre que possível deve-se incluir também sugestões para a solução dos problemas.

A documentação da avaliação não deve ser restrita aos documentos apresentados aqui, pois cada método de avaliação irá exigir documentos específicos, especialmente quando se realiza teste com usuários. (Rubin, 1994) apresenta uma série de documentos e procedimentos para condução de testes com usuários que devem ser verificados pelos avaliadores que escolherem este método.

No sentido de guiar a avaliação de usabilidade sugere-se, ainda, as seguintes recomendações:

Nº 1 – A melhor avaliação é aquela que começa com um bom projeto. Importante lembrar que a avaliação é apenas um etapa dentro do processo de concepção da interface e não uma atividade independente. Deve ser considerado que mesmo a “melhor” avaliação não substitui os cuidados para evitar problemas de usabilidade. O custo da correção de problemas de usabilidade é sempre muito mais elevado que o custo de evitá-los.

Nº 2 – A usabilidade do site pode ser alterada em função de manutenções, atualizações de conteúdo do site, mudanças de tecnologia, etc. Portanto, avaliações periódicas devem ser realizadas. Com que freqüência? – Sempre que possível, embora de maneira geral a freqüência seja limitada, principalmente, ao prazo e orçamento disponíveis.

Nº 3 – Documentação é a palavra-chave para a boa condução da avaliação. Crie uma base de problemas de usabilidade identificados, soluções utilizadas e quais são os problemas mais freqüentes.

Nº 4 – Investimento na capacitação das pessoas mesmo aquelas que não estão diretamente relacionadas à avaliação. É importante que toda a equipe de desenvolvimento esteja consciente da importância da usabilidade do site. Desta forma será mais fácil explicar à equipe de desenvolvimento a severidade dos problemas e discutir alternativas possíveis para sua solução.

Nº 5 – Utilização de mais de um método de avaliação, pois um único método não é capaz de identificar todos os problemas possíveis.

Nº 6 – Considerar que a Web é um ambiente dinâmico, assim como seus usuários. Planeje a avaliação cuidadosamente considerando diferentes plataformas (software e hardware), velocidades de conexão, etc.

3.8. Bibliografia

- Attendees and Position Papers at CHI97 Workshop “Usability testing World Wide Web Sites”, (1997). Available at: <http://www.acm.org/sigchi/webhci/chi97testing/particip.htm>
- BAILEY, B. (1999) Heuristic Evaluations. Insights from Human Factors International. Available at: <http://www.humanfactors.com/library/may99.asp>
- BARTA, R.; Schranz, M.: “JESSICA: an object-oriented hypermedia publishing processor”. In Special Issue on the 7th. Int. WWW Conference, Brisbane, Australia. 1998. pp. 239-249.
- BERGHEL, H. (1996) The client's side of the World-Wide Web. Communication of the ACM 39, 1. P. 30 – 40.
- BERGHEL, H. (1996) The client's side of the World-Wide Web. Communication of the ACM 39, 1. P. 30 – 40.
- BERNERS-LEE, T. (1989) Information Management: A proposal. <http://www.w3.org/History/1989/proposal.html>
- BERNERS-LEE, T.; et al. (1994) The World Wide Web. Communication of the ACM, New York, v.37, n.8, p.76-82.
- BEVAN, N. (1995) Usability is quality of use. In: Anzai & Ogawa (eds) Proc. 6th International Conference on Human Computer Interaction, July. Elsevier. <http://www.usability.serco.com/papers/usabis95.pdf>
- BEVAN, N. (1998) Usability Issues in web site design. In: Proceedings of UPA'98, Washigton DC, 22-26. Also available at: <http://www.usability.serco.com/papers/usweb98.pdf>
- BIAS, R. (2000) Usability Triage for Web Sites. 6th Conference on Human Factors and the Web, Austin, TX, USA.
- BORGES, R.C.M. ; Winckler, M.A.A.; Basso, K. Considerações sobre o Uso de Cores em Interfaces WWW. In: Proc. of IHC2000.
- BRAJNIK, G. (2000) Automatic web usability evaluation: what needs to be done? 6th Conference on Human Factors and the Web, Austin, TX, USA, June 19, 2000.
- BRAJNIK, G. (2000) Automatic web usability evaluation: what needs to be done? 6th Conference on Human Factors and the Web, Austin, TX, USA, June 19, 2000.
- CASTILHO, J. C.; HARTSON, H. R.; HIX, D. (1998) The User-Reported Critical Incident Method at a Glance. Disponível por WWW em http://hci.ise.vt.edu/~josec/remote_eval/docs/TR_user_reported_CI_method.pdf
- CERI S., P.; Fraternali, P.; Bongio, A.: “Web Modeling Language (WebML): a modeling language for designing Web sites”. Proceedings of the 9th

- International Conference on the WWW (WWW9), Amsterdam, May 2000.
- CERI, S.; FRATERNALI, P.; BONGIO, A. Language (WebML): a modeling language for designing Web sites. 9th World Wide Web Conference, Amsterdam, May 15-19, 2000
- CONALLEM, J.: "Building Web Applications with UML". Addison Wesley, 2000.
- COWAN, D. D.; LUCENA, J. P.: "Abstract Data Views". Na Interface Specification Concept to Enhance Design for Reuse. IEEE Transactions on Software Engineering, 21(3), março 1995.
- CYBIS, W. et al. (1998) Uma Abordagem Ergonômica para o Desenvolvimento de Sistemas Interativos. In: WORKSHOP EM FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, IHC, 1998, Maringá – Paraná. Atas... [S.l.:s.n.].
- DEL GALDO, E.; NIELSEN, J. (1996) International User Interfaces. John Wiley & Sons. New York. 276 p.
- DIAZ, A., Isakiwitz, T., Maiorana, V. e Gilabert, G. RMC: A Tool to Design WWW Applications. In Proc. Fourth Int. WWW Conf. Boston, 1995. pp. 11-14.
- ETGEN, M.; CANTOR, J. (1999) What does getting WET (Web Event-logging Tool) Mean for Web Usability? Proceedings of 5th Conference on Human Factors & the Web, June 3, 1999, Maryland USA. Also available at: <http://zing.ncsl.nist.gov/hfweb/proceedings/etgen-cantor/index.html>
- FOWLER, S. L.; Novack, A. J.; Stillings, M. J. The Evolution of a Manufacturing Web Site. 9th WWW Conference, Amsterdam, May 15-19, 2000.
- FRATERNALI, P. Tools and approaches for developing data-intensive Web applications: a survey. ACM Comput. Surv. 31, 3 (Sep. 1999), Pages 227 – 263.
- FRATERNALI, P.; Paolini, P. Model-Driven Development of Web Applications: the Autoweb System. ACM Transactions on Office Information Systems vol. 18 (4), 2000.
- FRATERNALI, P.; Paolini, P.: "A Conceptual Model and a Tool Environment for Developing More Scalable, Dynamic, and Customizable Web Applications". In EDBT'98. 1998: 421-435.
- GAEDKE, M.; Graef, G.: "Development and Evolution of Web-Applications using the WebComposition Process Model". Int. Workshop on Web Engineering at the 9th Word-Wide Web Conference (WWW9). Amsterdam. May, 2000.
- GAFFNEY, G. (2000) Card Sorting. <http://www.infodesign.com.au/usability/cardsorting.html>

- GARZOTTO, F.; Paolini, P.; Bolchini, D.; Valenti, S.: "Modelling by patterns of Web Applications", Proc. Int'I Workshop on the World Wide Web and Conceptual Modeling. Springer Verlag, Berlin, 1999.
- GARZOTTO, F.; Paolini, P.; Schwabe, D.: "HDM – A Model-Based Approach to Hypertext Application Design". 1993. TOIS.
- GELLERSEN, H-W; Gaedke, M. "Object-oriented Web Application Development". IEEE Internet Computing, Vol.3, No.1, February 1999.
- GÜELL, N.; SCHWABE, D.; VILAIN, P. Modeling Interactions and Navigation in Web Applications. Proceedings of WWW and Conceptual Modeling 2000 Workshop; Salt Lake City, USA. 2000.
- HARTSON, H. R.; et al (1996). Remote Evaluation: The Network as an Extension of the Usability Laboratory. Proceedings of CHI'96, April 13 - 18, 1996, Vancouver Canada, 228-235.
http://miso.cs.vt.edu/~usab/remote/docs/chi96_remoteusab.html
- HOM, J. The Usability Methods Toolbox:
<http://www.best.com/~jthom/usability/usable.htm>
- ISAKOWITZ, T.; Stohr, E.; Balasubramaniam, P.: "RMM: A methodology for structured hypermedia design". Comm of the ACM, outubro 1995.
- ISO Draft International Standard (DIS) 9241-11 (1999), Ergonomic Requirements for office work with visual display terminals, Part 11: Guidance on Usability, International Standards Organization, Geneva.
- LYNCH, P. J; HORTON, S. (1999) Web Style Guide : Basic Design Principles for Creating Web Sites. Yale Univ Press. ISBN: 0300076754. 164 p.
- MATIAS, M. (1995) Checklist: Uma ferramenta de suporte à avaliação ergonômica de interfaces. Florianópolis: PPGEUFSC. Dissertação de Mestrado. Disponível por WWW em: <http://www.eps.ufsc.br/disserta/matias/>
- MAYHEW, D. (1992). *Principles and Guidelines in Software User Interface Design*, Prentice Hall, Englewood Cliffs.
- MURUGESAN, S.; Deshpande,Y.; Hansen, S.; Ginige, A.: "Web Engineering: A New Discipline for Development of Web-based Systems". Proc. First Int'I Conf. Software Engineering (ICSE), Workshop on Web Engeering, Univ. of Western Sydney, Austrália, 1999. Disponível em <http://aeims.uws.edu.au/WebEhome/ICSE99-WebE-Proc/San.doc>.
- NANARD, J., Nanard, M.: "Hypertext Design Environments and the Hypertext Design Process", CACM 38(8). Augosto, 1995. Pág. 49-56.
- NIELSEN, J. (1993) Usability Engineering. Boston - USA: Academic Press, 362 p.
- NIELSEN, J. (1997) The Use and Misuse of Focus Groups.
<http://www.useit.com/papers/focusgroups.html>
- NIELSEN, J. (1999) The Top Ten New Mistakes of Web Design.
<http://www.useit.com/alertbox/990530.html>

- NIELSEN, J.; SANO, D. (1995) Sun Web: User Interface Design for Sun Microsystem's Internal Web. Computer Networks and ISDN Systems (Selected papers from the 2th WWW Conference1994). Amsterdam, v.28, n.1&2. 1995, p.179-188. Also available at: <http://www.useit.com/papers/sunweb/>
- ROSSI, G.; Schwabe, D.: "OOHDM – Object Oriented Hypermedia Design Method". 1996. Departamento de Informática - Tese PUC-Rio. Disponível em <http://www-lifia.info.unlp.edu.ar/~fer/oohdm/>
- ROSSI, G.; Schwabe, D.; Lyardet, F.: "Improving Web Information Systems with Navigational Patterns". Proc. WWW8, Elsevier, 1999.
- RUBIN, J. Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests. John Wiley & Sons. New York. 1994. 330 p.
- SANO, D. (1996) Large-Scale Web Sites. John Wiley & Sons, New York, 288 p.
- SCAPIN, D. et al. Transferring Knowledge of User Interfaces Guidelines to the Web. In. Proc. of Tools for Working with Guidelines,. London: Springer; 2001; pp. 293-304.
- SCHWABE, D. et al. "Hypertext Development Using a Model-based Approach". Software-Practice and Experience, New York, v.22, n.11, p.937-962, Nov. 1992.
- SCHWABE, D. et al. Engineering Web Applications for Reuse, IEEE Multimedia, Spring 2001, pp. 2-12.
- SEGOR, C.; Gaedke, M.: "Crossing the Gap - From Design to Implementation in Web-Application Development". In Proceedings of Information Resources Management Association International Conference, Anchorage, Alaska, USA. 2000.
- SIOCHI, A. C.; EHRICH, R. W. (1991) Computer Analysis of Users Interfaces Based on Repetition in Transcripts of users Sessions. ACM Transaction on Information System, New York, v.9, n.4, p. 309-335.
- SULLIVAN, T. (1997) Reading Reader Reaction: A proposal for Inferential Analysis on Web Server Log Files. Disponível por WWW em: <http://www.uswest.com/web-conference/proceedings/rrr.html>
- Useit.com: Jakob Nielsen's Website: <http://www.useit.com/alertbox/>
- VANDERDONCKT, J (1994). *Guide ergonomique des interfaces homme-machine*, Presses Universitaires de Namur, Namur.
- Web Usability Questionnaire <http://www.wammi.com/>
- WINCKLER, M. et al. (1999). Remote Usability Testing: a Case Study. (Short paper) In: Proceedings OZCHI99. 28-30 November 1999, Wagga-Wagga, Australia.
- WINCKLER, M.; Farenc, C.; Palanque, P.; Bastide, R. Designing Navigation for Web Interfaces. In Proceedings:... IHM-HCI2001, Lille, França, 10-14 Sep. 2001 (Short paper).

WINCKLER, M.A.; FARENC, C.; PALANQUE, P.; PIMENTA,M.S.; *Avaliação da Navegação de Interfaces Web a partirde Modelos* In: Proc. of IV Workshop sobre Fatores Humanos em Sistemas Computacionais (IHC 2001), Florianópolis, Outubro de 2001.

WINCKLER, M.A.; PIMENTA,M.S; PALANQUE, P.; FARENC, C.; Usability Evaluation Methods: What is still missing for the WWW? In: Proc. of 9th International Conference on Human-Computer Interaction, HCII2001, New Orleans USA , August 5-10, 2001..

WOOLRYCH, A.; COCKTON, G. (2001) Why and when five users aren't enough. In proceedings... IHM-HCI 2001 (volume II, short paper). Lille, França.

World Wide Web Consortium: <http://www.w3.org/>

ANEXO A – Formulário de definição da avaliação de usabilidade

Objetivo: definir o escopo da avaliação. Inclui a descrição de métodos a serem utilizados, pessoas responsáveis pela avaliação, objetivos a serem alcançados, produto a ser avaliado, etc. Este formulário deve ser preenchido logo no início do processo de avaliação pelo responsável do projeto.

1. Identificação Geral da avaliação

1.1 Código da avaliação : _____

1.2 Produto/ Web site a ser avaliado: _____

1.3 Objetivos da avaliação: _____

1.4 PÚBLICO-alvo considerado: _____

1.5 Coordenador geral do processo de avaliação: _____

1.6 Cliente / responsável pela avaliação: _____

2. Procedimentos

2.1 Método(s) de avaliação a serem utilizado(s): _____

2.2 Procedimentos gerais para realização: _____

3. Recursos Necessários

3.1 Prazos para realização: _____

3.2 Número de avaliadores participantes: _____

3.3 Número de usuários participantes: _____

3.4 Equipamentos:

Computador

- PC
- Macintosh
- Estação Sun
- Outro (especificar): _____

Sistema Operacional

- Windows NT
- Windows Millenium
- Linux
- OS/2
- OS Mac
- Outro (especificar): _____

Software, browser, etc.

- Nestcape (especificar versão): _____
- Internet Explorer (especificar versão): _____
- Plug-in Java
- Plug-in Real Áudio
- Plug-in Cosmo Player
- Plug-in Adobe Acrobat Reader (PDF)
- Plug-in Shockwave
- Plug-in QuickTime
- Plug-in LiveAudio
- Plug-in Live3D
- Outro (especificar): _____

Conexão internet

- Modem: () 14 kb/s () 28 kb/s () 56 kb/s
- 128 kb/s
- T1
- T3/ Cable

Laboratório de usabilidade

- Sala com espelhos falsos
- Câmera e vídeo
- Software para coleta de interações
- Outros requisitos (especificar): _____
- Ambiente normal de trabalho do usuários

Outros recursos, favor especificar:

4. Orçamento

(Descrição detalhada do orçamento para realizar da avaliação...)

ANEXO B - Formulário para descrição de problemas

Objetivo: auxiliar na descrição de problemas de usabilidade identificados durante a realização de avaliações de usabilidade. Cada seção de teste deve ser identificada por este formulário. O conjunto de problemas identificados aqui serve como base para a geral do relatório geral de usabilidade (ver anexo A.).

1. Identificação da avaliação

1.1 Código da avaliação: _____

1.2 Código da seção de avaliação: _____

1.3 Método de avaliação utilizado: _____

1.4 Data de realização: ___/___/___

1.5 Tempo total de avaliação: ___ horas ___ minutos. Início: ___h___m Fim: ___h___m

1.6 Avaliador⁹: _____

1.7 Usuário¹⁰: _____

2. Tabela de problemas identificados

Cod. ¹¹	Tarefa ¹²	Tempo ¹³	Descrição do problema ¹⁴	URL + obj. ¹⁵	Cenário de uso ¹⁶	Impacto ¹⁷
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
...						

⁹ Incluir o nome do avaliador que realiza as observações ou análise dos dados. No caso de métodos de inspeção como, Avaliação Heurística, incluir apenas o nome do avaliador.

¹⁰ Observar a utilização de códigos para preservar a identidade dos usuários.

¹¹ Cod. é um número sequencial que identifica o problema.

¹² Tarefa é identificador ou a descrição da tarefa afetada pelo problema.

¹³ Tempo é o tempo (min:seg) utilizado para realização da tarefa ou o tempo investido pelo usuário antes de desistir da tarefa.

¹⁴ Descrição do problema é a descrição textual do problema.

¹⁵ URL + obj. é a URL da página onde ocorreu o problema mais objetos considerados (ex. texto, link, imagem, etc.).

¹⁶ Cenário de uso é a sequencia de ações realizadas antes, durante e após a ocorrência do problema.

¹⁷ Impacto é o impacto atribuído ao problema (grave/ importante/ menor).

ANEXO C – Formulário resumo de problemas identificados

Objetivo: o objetivo deste formulário é criar uma tabela única com todos os problemas encontrados durante a avaliação de usabilidade do site, incluindo o resultado das seções de avaliação. Neste formulários os problemas identificados são descritos independentemente do(s) método(s) de avaliação utilizados.

1. Identificação da avaliação

1.1 Código da avaliação: _____

1.2 Responsável pelo compilação dos resultados :_____

1.3 Data: ___/___/___

Cod.	Descrição do problema	Sessões de ocorrência ¹⁸	Impacto	Freqüência	Severidade	Soluções possíveis
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
29						
30						
31						
...						

¹⁸ Código de todas as seções onde o problema foi identificado.

