Avaliação Heurística

"Even finding some problems is of course much better than finding no problems..."
Jakob Nielsen and Rolf Molich

1. A avaliação

Existem basicamente 4 maneiras para se avaliar uma interface com usuário: *formalmente*, através de uma técnica de análise; *automaticamente*, por um procedimento computadorizado; *empiricamente*, por experimentos com testes de usuários; e *heuristicamente*, simplesmente percorrendo a interface e julgando-a de acordo com sua própria opinião.

Os princípios básicos de usabilidade envolvem três categorias principais: facilidade com que novos usuários podem efetivamente começar a interagir e alcançar máxima performance; multiplicidade de maneiras com que o usuário e o sistema trocam informação; e o nível de suporte que o usuário tem para determinar seu sucesso e a avaliação de suas metas **[Romani & Baranauskas, 1998]**.

A intenção básica da avaliação é identificar elementos que possam causar dificuldades ao usuário, porque violam princípios cognitivos conhecidos ou ignoram os resultados empíricos já bem aceitos. São 3 as metas principais da avaliação: examinar a funcionalidade do sistema, o efeito da interface no usuário e identificar problemas específicos de design. O método de avaliação heurística foca na terceira meta, que relaciona tanto funcionalidade quanto usabilidade da interface. Trata-se de identificar os aspectos negativos do design: elementos que, quando usados em seu contexto intencional, causam resultados inesperados ou confusão entre os usuários **[Romani & Baranauskas, 1998]**.

A avaliação heurística envolve especialistas avaliando um design com base em um conjunto de critérios de usabilidade ou heurísticas. O design é examinado em busca de instâncias nas quais esses critérios são violados. Um conjunto de critérios proposto originalmente por **[Molich & Nielsen, 1990]** inclui:

- Diálogo simples e natural;
- Falar a língua do usuário;
- Minimizar a carga de memória do usuário;
- Consistência;
- FeedBack;
- Saídas marcadas claramente;
- Atalhos;
- Mensagens de erro precisas e construtivas;
- Prevenir erros.

1.1 O custo-benefício

Testes de usabilidade podem geralmente requerer custos muito altos. O método descrito aqui tenta diminuir os custos da avaliação apenas prevendo aspectos de uso ao invés de observá-los diretamente. Esse método não envolve usuários, mas sim, certa forma de inspeção. Como não envolve usuários, a avaliação heurística se torna um método rápido e barato. Além disso, não são necessários equipamentos especiais.

Molic e Nielsen **[Molich & Nielsen, 1990]** planejaram esse método conhecido como avaliação heurística, em resposta à necessidade de métodos de baixo custo que pudessem ser utilizados por pequenas empresas que não possuem facilidades, tempo, dinheiro ou pessoal treinado para a engenharia de usabilidade. Na avaliação heurística, são utilizados avaliadores, no lugar dos usuários, que examinam o sistema ou protótipo, guiados por um conjunto de heurísticas de alto nível **[Nielsen, 1992]**.

A avaliação heurística é uma técnica analítica e informal, que faz parte da chamada *"discount usability engineering"*, um método barato, rápido e fácil de usar. Além disso, a literatura tem mostrado, em estudos experimentais comparativos com outras técnicas de avaliação, que a avaliação heurística tem apresentado melhores resultados, encontrando problemas mais sérios com menos esforço despendido **[Jeffries et. al., 1991] [Nielsen, 1994] [Romani & Baranauskas, 1998]**.

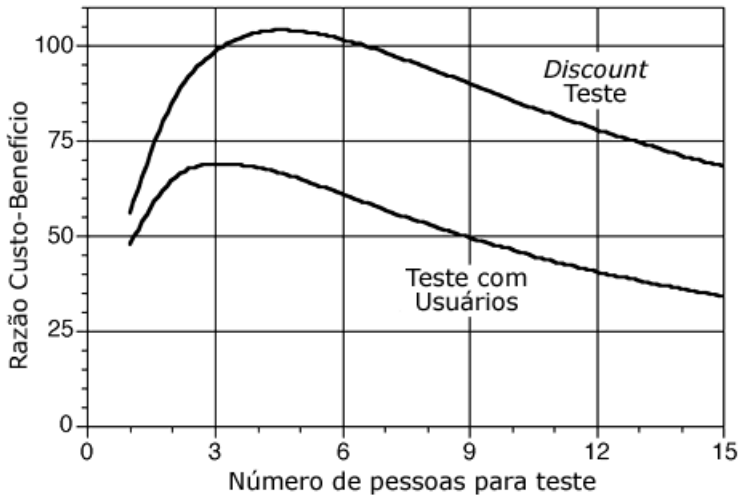


Figura 1 A curva mostra a razão custo-benefício, i.e., quantas vezes os benefícios são maiores do que os custos. Por exemplo, a razão igual a 50 provavelmente corresponderá para um custo de R$ 1.000, um benefício de R$ 50.000.

1.2 Os avaliadores

Testes empíricos realizados com diversos avaliadores em vários estudos da literatura envolvendo o método de avaliação heurística induzem à seguinte corrente de pensamento, na qual validaremos no nosso estudo de caso na **Seção 3** adiante. Estudos iniciais em **[Nielsen, 1989]**, **[Molich & Nielsen, 1990]** e **[Nielsen & Molich, 1990]** mostram que alguns avaliadores obtêm resultados melhores do que outros, mas não se engane achando que os avaliadores ditos "bons" (mais qualificados, com mais experiência e maior conhecimento do domínio específico) obtiveram melhores resultados que os "ruins" (iniciantes na engenharia de usabilidade). Esse é o grande mérito do método de avaliação heurística, rápido, prático, fácil de aprender e barato (por não despender de especialistas). Estudos mostraram que os bons avaliadores são mais eficientes, mas isso não indica que esses avaliadores conseguem descobrir todos os problemas dos avaliadores novatos. Em **[Nielsen & Molich, 1990]** e **[Baker et. al., 2002]** foram demonstrados que mesmo os avaliadores inexperientes conseguem descobrir problemas graves na interface, além é claro dos problemas mais fáceis que os avaliadores mais experientes acabam às vezes passando despercebidos. Estudos em diversas interfaces mostram que pessoas diferentes encontram diferentes problemas de usabilidade.

A Figura 2 extraída de **[Nielsen, 1994]** e **[Baker et. al., 2002]** relacionadas a problemas em interfaces diferentes, comprovam a hipótese levantada anteriormente:

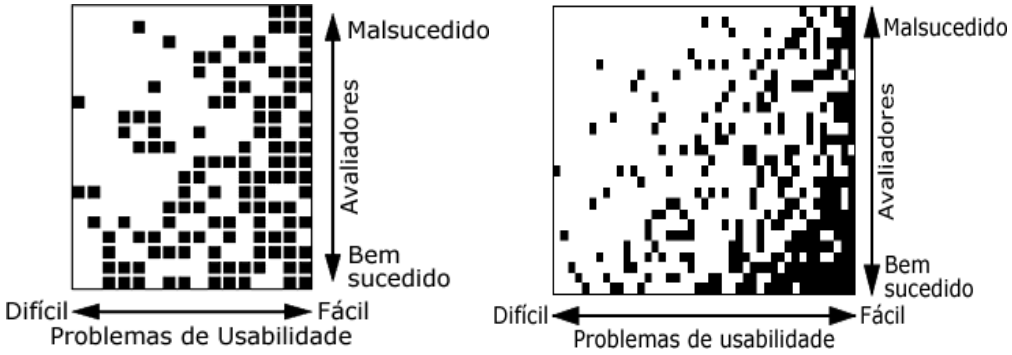


Figura 2 Cada linha representa um avaliador e cada coluna representa um problema de usabilidade. O quadrado preto representa se o problema foi encontrado pelo avaliador.

Esses gráficos mostram que mesmo avaliadores ruins conseguem encontrar os problemas mais graves, que são os mais importantes e o real objetivo de uma boa avaliação. Dessa forma, Nielsen conclui que a melhor forma de se realizar uma avaliação heurística seria com um agregado de avaliadores escolhidos aleatoriamente, não importando o grau de conhecimento de cada avaliador. Ao final da avaliação individual, as listas de problemas de usabilidade devem ser reunidas e discutidas pelo grupo de avaliadores ou por um perito, para que seja feita uma única lista com todos os problemas de usabilidade encontrados.

Bom, descobrimos que não necessariamente precisamos de peritos para avaliar uma interface, mas quantos avaliadores devem ser usados para que obtenhamos um bom resultado? Essa pergunta foi bastante explorada na literatura, e podemos encontrar diversas respostas que entram em consenso **[Nielsen & Molich, 1990] [Baker et. al., 2002]**. De acordo com a Figura 3, conclui-se que são necessários de 3 a 5 avaliadores para que sejam encontrados uma média de 60-80% dos problemas da interface avaliada. Pode parecer insuficiente, porém se você considerar que se trata de um método barato, intuitivo, de fácil motivação para fazê-lo, não requer um plano avançado e, além disso, possui rápida aplicação podemos concluir que é um resultado muito bom. É importante observar que, se desejar uma solução melhor basta seguir a curva do gráfico, ou seja, aumente o número de avaliadores, utilize também alguns avaliadores mais experientes e com maior conhecimento do domínio, mas tudo isso tem um preço, você está disposto a pagar e terá tempo para isso? Lembre-se que achar algum problema é melhor do que nenhum.

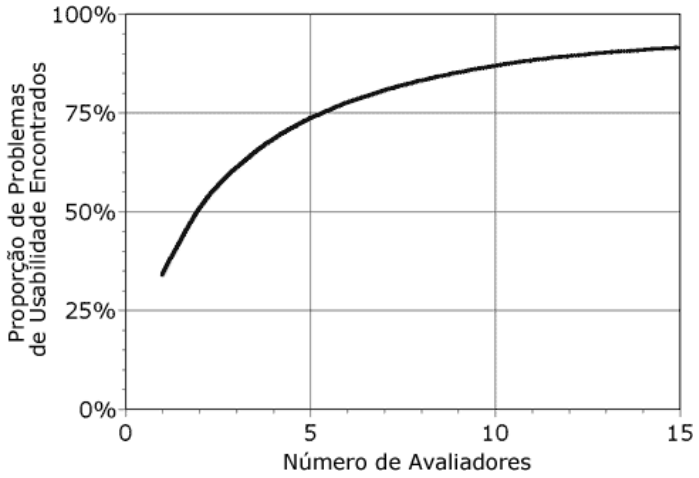


Figura 3 Proporção de problemas de usabilidade encontrados utilizando avaliação heurística pela média de 6 estudos de caso **[Nielsen, 1994]**.

Nielsen em **[Nielsen, 1994]** recomenda que sejam usados cinco avaliadores. Dessa forma, com a economia realizada, podem-se realizar outros métodos de avaliação para que outros problemas sejam detectados ou mesmo para afirmar a eficiência do método de avaliação heurística **[Nielsen & Molich, 1990]**, já que existe certa crítica em relação ao uso de métodos não estatísticos, comprovado por **[Lieberman, 2003]** que teve um artigo rejeitado por não ser "empírico o bastante".

2. Método de Avaliação Heurística

De acordo com **[Nielsen, 1993]**, cada avaliador normalmente faz duas ou mais iterações sobre a interface para:

- Inspecionar o fluxo da interface de uma tela para outra,
- Inspecionar cada tela por vez contra as heurísticas, examinando características como mensagens do sistema, diálogos e assim por diante dentro do escopo das heurísticas.

Uma sessão típica irá durar entre uma e duas horas, mas se a interface for grande pode ser necessário um tempo maior. Nas sessões de avaliação, cada avaliador conduz sua avaliação individualmente e independentemente dos outros avaliadores, ele não deve discutir com os outros até que todas as avaliações estejam prontas. Quando todos os avaliadores terminarem, eles devem juntar seus problemas em apenas uma única lista de problemas de usabilidade **[Preece et. al., 1994]**.

O método de avaliação heurística deve ser visto como parte do processo de design interativo de uma interface. Ele envolve um pequeno conjunto de avaliadores examinando a interface e julgando suas características em face de reconhecidos princípios de usabilidade, as heurísticas **[Rocha & Baranauskas, 2000]**.

De modo geral, é difícil de ser feita por um único avaliador, porque uma única pessoa nunca é capaz de encontrar todos os problemas de usabilidade de uma interface. A literatura tem mostrado que diferentes pessoas encontram diferentes problemas, e, portanto se melhora significativamente os resultados da avaliação heurística utilizando múltiplos avaliadores. A recomendação é que se use de três a cinco avaliadores (**Seção 1.2**) **[Nielsen, 1992]**.

2.1 Heurísticas revisadas

A avaliação heurística é feita em um primeiro momento individualmente. Durante a sessão de avaliação cada avaliador percorre a interface diversas vezes (pelo menos duas) inspecionando os diferentes componentes do diálogo e ao detectar problemas os relata associando-os claramente com as heurísticas de usabilidade que foram violadas. As heurísticas são regras gerais que objetivam descrever propriedades comuns de interfaces usáveis e foram revisadas em **[Nielsen, 1994]**:

1. **Estética e design minimalista:** diálogos não devem conter informação irrelevante ou raramente necessária. Qualquer unidade de informação extra no diálogo irá competir com unidades relevantes de informação e diminuir sua visibilidade relativa.
2. **Coerência do sistema com o mundo real:** o sistema precisa falar a linguagem do usuário, com palavras, frases, expressões e conceitos similares ao usuário, ao invés de termos orientados ao sistema. Seguir convenções do mundo real, fazendo com que a informação apareça numa ordem natural e lógica.
3. **Reconhecimento ao invés de relembrança:** tornar objetos, ações e opções visíveis. O usuário não deve ter que lembrar informação de uma para outra parte do diálogo. Instruções para uso do sistema devem estar visíveis e facilmente recuperáveis quando necessário.
4. **Consistência e padrões:** usuários não precisam adivinhar que diferentes palavras, situações ou ações significam a mesma coisa. Seguir convenções de plataforma computacional.
5. **Visibilidade do status do sistema:** o sistema precisa manter os usuários informados sobre o que está acontecendo, fornecendo um *feedback* adequado dentro de um tempo razoável.
6. **Controle do usuário e liberdade de opções:** usuários frequentemente escolhem por engano funções do sistema e precisam ter claras saídas de emergência para sair do estado indesejado sem ter que percorrer um extenso diálogo. Prover funções *undo* e *redo*.
7. **Flexibilidade e eficiência de uso:** usuários novatos se tornam peritos com o uso. Prover aceleradores de forma a aumentar a velocidade da interação. Permitir a usuários experientes atalhos em ações freqüentes.
8. **Ajudar usuários a reconhecer, diagnosticar e corrigir erros:** mensagens de erro devem ser expressas em linguagem clara (sem códigos) indicando precisamente o problema e construtivamente sugerindo uma solução.
9. **Prevenção de erros:** melhor que uma boa mensagem de erro é um design cuidadoso o qual previne o erro antes dele acontecer.
10. **Ajuda e documentação:** embora seja melhor um sistema que possa ser usado sem documentação, é necessário prover uma *Ajuda* (*Help*) e uma documentação. Essas informações devem ser fáceis de encontrar, focalizadas na tarefa do usuário, listando os passos para se realizar a tarefa e não serem muito extensas.

2.2 Etapas da avaliação

Com base nestes princípios, os avaliadores passam a percorrer a interface e descrevem em formulários os problemas nela encontrados. Nestes formulários devem constar os problemas encontrados (descrição), seus tipos (princípios infringidos), como foram descobertos (ações executadas que levaram à identificação dos problemas), grau de severidade e possível solução (opcional, pois não é objetivo da avaliação).

Etapas da avaliação heurística:

1. Definição dos requisitos da avaliação: objeto, avaliadores, objetivos, escopo, aspecto, recursos necessários, etc.
2. Introdução: apresentação de informação aos avaliadores, incluindo objetivos, princípios (heurísticas) e material de apoio (formulários, exemplos, manuais, etc.).
3. Avaliação da Interface: avaliadores testam a interface em busca de problemas de usabilidade. Os problemas encontrados devem ser registrados.
4. Discussão: avaliadores e desenvolvedores (opcional) reúnem-se para discutir os problemas detectados e atribuir um grau de severidade aos mesmos.
5. Apresentação dos resultados: divulgação dos problemas e determinação dos mais graves, que devem ser corrigidos.

Embora represente um aumento no custo, realizar uma avaliação heurística sem especialistas em usabilidade é impossível. Ao menos um indivíduo (orientador) é necessário para apresentar os princípios aos não-especialistas e realizar uma discussão para determinar a gravidade dos problemas. Tais discussões ocorrem após as sessões de interação e correspondem à exposição de problemas encontrados e atribuição consensual de graus de severidade aos mesmos.

Independentemente do perfil e experiência dos avaliadores escolhidos, este método exige que a interface esteja funcionalmente disponível, ou seja, implementada, ao menos parcialmente ou como protótipo para que os avaliadores possam utilizá-la. Isto limita a aplicabilidade do método em fases iniciais de um ciclo de desenvolvimento de interface, retardando a descoberta de problemas **[Chan & Rocha, 1996]**.

2.3 Exemplos

A seguir, são apresentados alguns problemas detectados em avaliações heurísticas extraídos de **[Rocha & Baranauskas, 2000]**:

Exemplo 1

O antivírus Norton 2000 para NT Server é um software projetado para proteger servidores de rede Windows NT de arquivos infectados por vírus. O software é executado no servidor NT e sempre que se tenta gravar/abrir algum arquivo infectado no servidor, o programa apresenta uma mensagem na tela do servidor avisando que o arquivo está infectado, mas os usuários em estações cliente NT não recebem esse aviso. Tem-se, portanto, a heurística **visibilidade do status do sistema** violada. Consequentemente, quando o usuário trabalhando em uma estação cliente tenta gravar um arquivo infectado no servidor o antivírus impede a gravação e não emite nenhuma mensagem para a estação cliente e o usuário pode então perder seu trabalho sem saber que isso está ocorrendo. O resultado é a perda do arquivo, o que teria sido facilmente prevenido se alguma mensagem de alerta fosse exibida à estação cliente. Claramente a heurística **ajudar usuários a reconhecer,**

diagnosticar e corrigir erros também é violada. Pode-se dizer que a heurística **prevenção de erros** também não é respeitada.

Exemplo 2

No sistema Windows quando se quer instalar um novo componente de hardware é iniciado um processo de busca e o indicador de detecção pode ficar parado por muito tempo, como indicado na janela de diálogo na Figura 4. O usuário fica perdido na maioria das vezes por não saber o que significa esse muito tempo e não sabe se deve ou não reiniciar o computador, mesmo porque usuários de sistemas semelhantes sabem quão pouco confiável é a relação da barra de detecção com o real andamento da operação (**visibilidade do status do sistema**; **ajudar usuários a reconhecer, diagnosticar e corrigir erros**).

Figura 4 Detecção de hardware do sistema Windows 98.

Exemplo 3

O software Winzip em uma mesma versão gratuita (não registrada) tem uma tela de abertura onde os botões aparecem em ordem aleatória a cada execução (Figura 5). Arbitrariamente os botões I Agree e Quit aparecem em ordem trocada levando o usuário a errar sem saber por que (**consistência e padrões**; **prevenção de erros**).

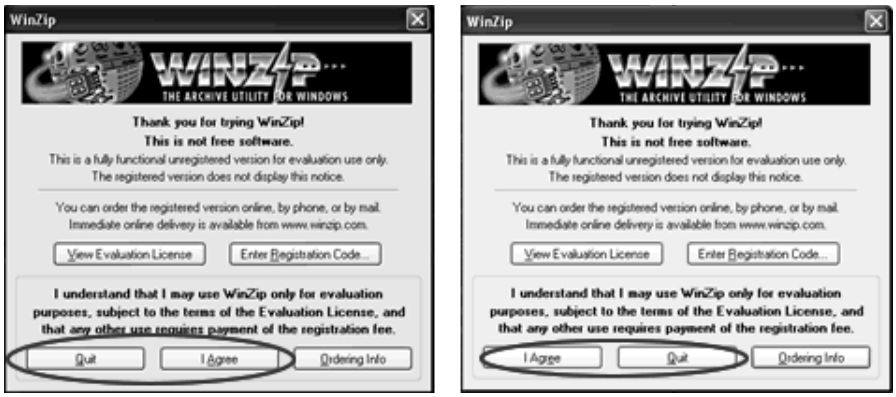


Figura 5 Tela de abertura da cópia para avaliação do software Winzip.

Exemplo 4

No Windows Explorer, ao tentarmos excluir um arquivo que está em uso, uma caixa de diálogo é aberta (Figura 6). Nessa caixa aparece a mensagem de que não foi possível acessar o arquivo, e recomenda ao usuário que verifique se o disco está cheio ou protegido, e finalmente se o arquivo não está sendo usado. Não há usuário

que não se confunda: o que tem a ver disco cheio com excluir um arquivo (**ajudar usuários a reconhecer, diagnosticar e corrigir erros**)!

Figura 6 Mensagem de erro do Windows Explorer.

Exemplo 5

No Adobe's ImageReady a mensagem de erro da Figura 7 é apresentada para o usuário. A mensagem diz que a aplicação não pôde ser iniciada porque um ponteiro estava *nil*. Qualquer usuário que não tenha experiência com programação não consegue entender o real motivo do erro e nem corrigi-lo (**coerência do sistema com o mundo real**; **ajudar usuários a reconhecer, diagnosticar e corrigir erros**).

Figura 7 Mensagem de erro do Adobe's ImageReady.

Exemplo 6

No diálogo da Figura 8, a ferramenta Oracle's SQL*Net Easy Configuration apresenta um problema de usabilidade, pois existe uma redundância de funções (*Cancel* e *Exit SQL*Net...*). O usuário acaba ficando confuso, quando na verdade todas conduzem ao mesmo resultado (**estética e design minimalista**).

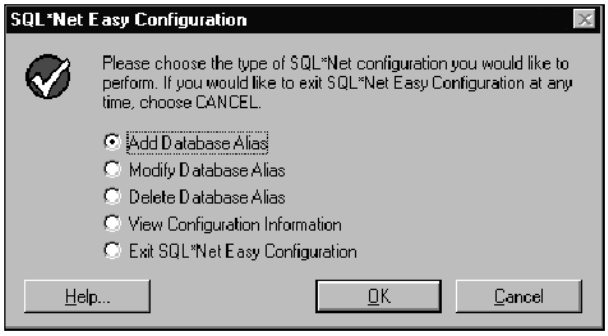


Figura 8 Oracle's SQL*Net Easy Configuration utility.

Com esses exemplos, podemos, de forma mais clara, perceber que o diagnóstico de um problema associado com as heurísticas que foram consideradas traz possibilidades concretas de redesign, embora esse não seja o principal objetivo desse método.

2.4 Graus de severidade

O uso efetivo de uma lista de problemas de usabilidade irá requerer que esses problemas sejam priorizados com relação à gravidade de cada problema. Prioridades são necessárias para não se despender esforços desproporcionais corrigindo problemas que não irão alterar em muito a interação do usuário com a interface.

Graus de severidade são geralmente derivados do impacto gerado pelo problema tanto no usuário quanto no mercado. Por serem muitas vezes critérios dependentes da aplicação, a definição de graus de severidade não é muito bem estabelecida na literatura, mas alguns autores dão exemplos de como atribuir graus de severidade à problemas de usabilidade **[Nielsen, 1994]**.

Precisa ser estimado o custo associado à implementação das sugestões de redesign. Certamente, problemas de usabilidade com alto grau de severidade devem ser corrigidos não interessando o quanto custe. Freqüentemente, muitos dos problemas não muito graves podem ser corrigidos com alterações mínimas de código. Esse compromisso não pode ser considerado como parte do método de inspeção de usabilidade, pois é preferível ter uma relação completa de todos os problemas sem o pré julgamento da viabilidade ou não da sua correção. Esta informação é importante no momento em que forem alocados recursos para corrigir os problemas mais sérios e se necessário deixar os menos graves para uma nova versão.

A gravidade de um problema é a combinação de três fatores:

- **A freqüência com que ele ocorre:** se é comum ou raro;
- **Impacto do problema quando ele ocorre:** se é fácil ou difícil para o usuário superá-lo;
- **A persistência do problema:** problema que ocorre uma única vez e que o usuário pode superar desde que saiba que ele existe, ou se os usuários serão repetidamente incomodados por ele.

Finalmente, é claro que é preciso considerar o impacto do problema no mercado, pois muitos problemas simples de serem superados têm um efeito importante na popularidade de um produto **[Rocha & Baranauskas, 2000]**.

É difícil conseguir uma boa estimativa de severidade por parte dos avaliadores durante uma sessão de avaliação heurística, pois os avaliadores estão concentrados em encontrar novos problemas de usabilidade. Também, cada avaliador irá apenas encontrar um pequeno número de problemas, então a classificação dos problemas do usuário estará incompleta em relação à lista de problemas detectados. Ao invés disso, graus de severidade podem ser coletados através de questionários entregues aos avaliadores depois das sessões de avaliação, listando todos os problemas de usabilidade descobertos, e pedindo a classificação de cada problema. Visto que cada avaliador só identificará um pequeno conjunto de problemas incluídos na lista, os problemas precisam estar bem descritos. As descrições podem ser resumos das descrições dos problemas encontrados pelos outros avaliadores. Essas descrições permitem que os avaliadores avaliem os diversos problemas mesmo que eles não os tenham encontrado. Tipicamente, os avaliadores gastam apenas 30 minutos para fornecer seus graus de severidade. É importante notar que cada avaliador deve fornecer seus próprios graus de severidade independentemente dos outros avaliadores.

Geralmente, os avaliadores não terão acesso ao sistema enquanto eles estão considerando a severidade dos vários problemas de usabilidade. É possível que os avaliadores consigam ganhar uma percepção adicional re-visitando partes da interface em execução do que apenas contando com suas memórias e com as descrições dos problemas. Ao mesmo tempo, não há dúvida de que os avaliadores serão mais lentos se recorrerem a mais uma interação com o sistema. Além disso, problemas de agenda irão certas vezes tornar difícil fornecer equipamentos para todos os avaliadores em uma hora conveniente se recursos especiais são necessários para executar um sistema protótipo ou se a distribuição do software está limitada devido a certas restrições (e.g. um projeto confidencial).

Nielsen ressalta que graus de severidade fornecidos de apenas um avaliador não são confiáveis. Quanto mais avaliadores fizerem o julgamento da severidade dos problemas de usabilidade, maior será a qualidade da média de classificação de severidade, e o uso da média de classificação feita por três avaliadores é satisfatória para muitos propósitos práticos.

2.5 Extensão das heurísticas originais

Além das heurísticas propostas originalmente por Molich e Nielsen em consideração a todos os elementos do diálogo, o avaliador, obviamente, pode considerar qualquer princípio de usabilidade adicional ou resultados relacionados que possam ser pertinentes a elementos de diálogo específico. Além disso, é possível desenvolver heurísticas específicas de categoria que são aplicadas a uma classe específica de produtos como complemento às heurísticas gerais. Uma forma de construir uma lista suplementar de heurísticas específicas de um domínio, é executar uma análise competitiva, testes com usuários em produtos existentes no domínio desejado e tentar abstrair princípios que explicam os problemas de usabilidade encontrados.

Diversos trabalhos foram realizados com o objetivo de estender as heurísticas gerais **[Romani & Baranauskas, 1998]**, **[Baker et. al., 2001]** e **[Baker et. al., 2002]**. Neles, foram discutidos princípios que visavam um escopo específico de um determinado domínio para uma dada interface. Por exemplo, as heurísticas utilizadas para avaliação de uma interface ditavam o escopo da avaliação, ou seja, em **[Baker et. al., 2001]** o alvo de avaliação era o quanto o seu ambiente de *groupware* motivava a colaboração entre os usuários. Para isso, são necessárias heurísticas específicas para trabalho em equipe, ou seja, heurísticas projetadas para identificar problemas de usabilidade específicos para trabalho em grupo, entre equipes separadas por uma grande distância para trabalharem sobre uma interface de trabalho visual compartilhada. Uma das heurísticas criadas era:

- **Facilitar o descobrimento de colaboradores e estabelecer contato:** a maioria dos encontros são informais, não programados, espontâneos ou iniciados por uma pessoa. No dia-a-dia, estes encontros são facilitados pela proximidade física, onde indivíduos próximos podem manter consciência sobre quem está por perto. Pessoas freqüentemente entram em contato com outra através de interações casuais (e.g. se encontrando casualmente em um corredor) e são capazes de iniciar e conduzir a conversa sem esforço. No pouco tempo durante a conversa, muito pode acontecer, como a troca de informações e coordenação de ações. Em comunidades eletrônicas, a falta de proximidade física significa que outros mecanismos são necessários para suportar a consciência e o encontro informal.

Note que a heurística acima não se encaixa nas heurísticas gerais definidas anteriormente, esta avalia um domínio específico. Veja que os avaliadores estão interessados em avaliar esta característica da interface, o que impede uma perda de tempo e dinheiro avaliando aspectos da interface que não interessam aos usuários específicos da aplicação. Isso significa que as heurísticas podem ser usadas para avaliação de um aspecto específico da interface, elas mesmas ditam o escopo da avaliação. Daí a importância de uma escolha de bons princípios de avaliação.

Em **[Romani & Baranauskas, 1998]** foram definidas heurísticas para outro contexto. A interface sendo avaliada era um Sistema de Acompanhamento e Avaliação de Rebanhos Leiteiros (*ProLeite*), que é usado na organização das informações de desempenho produtivo e reprodutivo dos animais de rebanhos leiteiros. O sistema possui grande quantidade de formulários para entrada de dados, possibilita consultas e emissão de relatórios. Para atender a categoria de usuários no domínio considerado,

que possuem pouca precisão para movimentos leves e precisos, pois são usuários acostumados a equipamentos pesados e rudes como a enxada, **[Romani & Baranauskas, 1998]** propuseram diversas heurísticas que complementam as heurísticas gerais. Uma das heurísticas propostas era:

- **Permitir opções de configuração:** para facilitar o uso do mouse para usuários com dificuldade de manipulação deste periférico, o sistema deve permitir alterar o tamanho dos botões para tamanhos maiores. Além disso, o sistema deve permitir a alteração das cores dos rótulos e cor dos campos de entrada, para facilitar a leitura e prever a possibilidade de execução do sistema em qualquer tipo de configuração de tela (por exemplo, 640X480, 800X600, fontes grandes e pequenas, etc) sem perda de qualidade dos diálogos.

Mas nem tudo é perfeito. A extensão das heurísticas gerais também pode causar problemas. **[Baker et. al., 2002]** mostrou que a avaliação com as heurísticas específicas para um domínio obteve um desempenho inferior às avaliações realizadas com as heurísticas gerais. Baker atribuiu esse resultado ao fato de que os avaliadores podem ter achado as heurísticas específicas mais difíceis de aprender e aplicar do que as heurísticas de Nielsen. Mesmo assim, as heurísticas específicas obtiveram um resultado satisfatório considerando o custo-benefício da avaliação.

3. Estudo de caso: OriOn

OriOn[1] é um grupo de discussão utilizado na Universidade Federal de Ouro Preto para ensino e pesquisa da disciplina Sistemas Interativos. Um grupo de discussão é uma localização *online* onde as pessoas interagem com outras através da colocação e leitura de mensagens sobre tópicos de interesse pessoal e para a comunidade.

O OriOn foi escolhido para realizar a avaliação heurística justamente para explorar o poder da extensão das heurísticas, pois para fazer esse estudo seria interessante definir heurísticas específicas para o domínio em questão: *"comunidades online"*.

Definida a interface que será avaliada, seguiremos o processo de avaliação heurística através das etapas de avaliação (**Seção 3.1**). Definidas as etapas do processo, devemos determinar o número de avaliadores que serão utilizados (**Seção 3.2**). Após a realização da avaliação e coleta dos dados, faremos a análise dos resultados (**Seção 3.4**) comparando com trabalhos relacionados da área **[Nielsen & Molich, 1990] [Baker et. al., 2002]**.

3.1 Etapas da avaliação

As etapas para a avaliação heurística sobre a interface do OriOn são as mesmas definidas na **Seção 2.2**, porém existe uma etapa adicional inicial para definir as heurísticas específicas para comunidades *online* (discutidas na **Seção 3.3**). Os resultados de cada etapa do processo de avaliação heurística serão discutidos na **Seção 3.4**. As etapas são:

1. Definição das heurísticas: específicas para comunidades *online*, ditando o escopo da avaliação;
2. Definição dos requisitos da avaliação: objeto de avaliação (OriOn), avaliadores (**Seção 3.2**);

[1] www.orion.iceb.ufop.br

3. Introdução: apresentação de informação do método aos avaliadores, incluindo as heurísticas para comunidades *online* (**Seção 3.3**);
4. Avaliação da interface: os avaliadores testam a interface em busca de problemas que infrinjam as heurísticas apresentadas. Os problemas devem ser registrados;
5. Discussão: um orientador reunirá todos os problemas encontrados pelos avaliadores e eliminará os problemas redundantes. Atribuição de graus de severidade não será necessária já que o objetivo do nosso estudo não é corrigir a interface avaliada;
6. Apresentação dos resultados: divulgação dos resultados obtidos comparados com trabalhos relacionados **[Nielsen & Molich, 1990] [Baker et. al., 2002]**.

3.2 Avaliadores

O objetivo desse estudo de caso é fazer um comparativo com outros trabalhos que utilizaram a avaliação heurística. Para isso, acompanhando a curva da Figura 3, foram necessários apenas quatro avaliadores para realizar toda a avaliação de forma satisfatória. Os avaliadores foram selecionados entre estudantes do curso de Ciência da Computação que já fizeram alguma disciplina da área de IHC. Assim, todos os avaliadores tiveram alguma experiência em relação aos conceitos de sistemas iterativos.

3.3 Heurísticas

Para que a avaliação da interface do OriOn seja feita sobre os aspectos relevantes de uma comunidade *online*, foram definidas algumas heurísticas que levam em consideração princípios como sociabilidade e usabilidade para ambientes *online*. Para isso, devemos lembrar que:

- Comunidades são dinâmicas; mudam e evoluem constantemente, influenciadas pela personalidade dos participantes, atividades do grupo, e por algumas influências externas. Por exemplo, o que pode ser importante no início de uma comunidade pode não ser mais tarde.
- Tecnologia não é o fator mais importante em comunidades *online*. Membros são.

A partir de guias definidos em **[Preece, 2000]** e de um estudo sobre as expectativas de usuários potenciais a respeito de grupos de discussão virtuais **[da Silva et. al., 2003]**, podemos definir algumas heurísticas:

1. **Definir um propósito claramente:** uma comunidade deve possuir um nome claro e significativo e conter uma descrição clara e concisa sobre o seu propósito.
2. **Prover acesso:** para que os usuários tenham acesso à comunidade é necessário que seja descrito claramente os requerimentos técnicos e outras informações adicionais essenciais para o acesso dos usuários. Não utilizar *URLs* longos e com caracteres incomuns. Utilizar cores padrões para os *links* (e.g. azul, para *links* não seguidos, e vermelho, para *links* já visitados), a mudança dos padrões pode causar confusão. Prevenir períodos longos de *download*, limitando o uso desnecessário de gráficos e animações que podem se tornar cansativos para os usuários em futuros acessos.
3. **Facilitar a comunicação:** ajudar os participantes a fazerem suas intenções claras provendo *emoticons* e um *menu* de gestos auxiliam o processo de comunicação. Prover ferramentas de edição (e.g. fontes, símbolos, verificador ortográfico, etc.) para a postagem de falas. Disponibilizar diferentes formas de visualização e estruturação da discussão para que cada usuário se sinta confortável com a de

sua preferência. Deve-se distinguir as mensagens que já foram lidas das que ainda não foram. Permitir que seja realizado um certo tipo de filtragem de informação e fornecer resumos sobre as discussões.

4. **Motivar a comunidade:** as discussões em uma comunidade *online* devem ter um auxílio para que sejam efetivas para todo o grupo de discussão. Para isso, deve-se encorajar a confiança, empatia e cooperação dos participantes. Focar no propósito da comunidade aumenta a proximidade entre os membros. Estabelecer políticas, para encorajar o processo de comunicação, ajuda a conter agressões e outros tipos de comportamentos inapropriados.
5. **Definir uma política de registro:** deve-se definir uma política de registro de novos usuários, bem como uma política que permita ou não a entrada de visitantes na comunidade. Os procedimentos de registro e *login* devem ser realizados com o menor número de passos possíveis e descritos claramente. Incluir uma ajuda para essas atividades auxilia o processo, bem como uma forma de usuários recuperarem e modificarem suas senhas de maneira rápida, fácil e segura.
6. **Garantir segurança:** uma comunidade deve proteger informação confidencial. Para isso, toda informação, das discussões realizadas, deve permanecer na comunidade e todos os membros devem concordar com uma política de privacidade.
7. **Navegação:** deve-se prevenir a ocorrência de páginas órfãs, que não estão ligadas ao site da comunidade, levando os usuários a becos sem saída. Para que a navegação ocorra de forma satisfatória, deve-se fornecer um suporte, ou seja, um mapa de site claramente definido. Esse mapa deve aparecer em qualquer lugar que o usuário esteja no site, para que auxilie os usuários a criarem um modelo mental de como partes diferentes do site se inter-relacionam.
8. **Consistência das informações:** manter as informações atualizadas e completas. Manter a consistência também é muito importante (terminologias, fontes, nome de *menus,* navegação, etc.).
9. **Participação na gestão:** lembre-se de que quem mantém a comunidade são os seus membros, por isso, deve existir algum mecanismo que delegue alguma forma de gestão para os membros da comunidade. Decisões de acesso, por visitantes, às discussões do grupo; decisões a respeito de membros que não contribuem nas discussões; decisões sobre a aparência do grupo, etc.

3.4 Resultados

Depois de realizada a avaliação e coletados os problemas encontrados pelos avaliadores, os problemas redundantes foram eliminados e o restante dos problemas foram contabilizados em um total de 17 infrações das heurísticas para comunidades *online*:

Problema 1. Não existe como opção exibir o nome do grupo e muito menos uma descrição clara sobre o seu propósito. O usuário que está entrando no grupo, não tem como saber qual o seu propósito (Figura 9). Antes de logar ou de cadastrar, o usuário não tem nenhuma descrição sobre do que se trata o grupo, e nem em qual está cadastrado, mesmo depois de estar logado, ele não tem nenhuma descrição do grupo. Uma solução seria para todos os grupos, colocar uma breve descrição do grupo, com os seus propósitos bem claros para o que o usuário possa saber se realmente é o que ele esperava. (**Definir um propósito claramente**).

Figura 9 Tela inicial do OriOn.

Problema 2. O ambiente possui um nome claro (OriOn - Orientação Online), mas não possui uma descrição clara sobre o ambiente (Figura 9). Somente quem o conhece ou quem de inicio explora o OriOn, observando a disposição das informações, sabe que se trata de um ambiente de discussão on-line. Poderia haver um link "Sobre" para uma tela com um parágrafo ou um pequeno texto descrevendo o OriOn. (**Definir um propósito claramente**).

Problema 3. O OriOn não possui uma descrição dos requisitos mínimos de sistema para uma execução adequada do ambiente (Figura 9). Deveria ser disponibilizado na tela de login um link para uma descrição dos requisitos mínimos do sistema. Caso este for pequeno (uma frase), talvez seja interessante disponibilizar na própria tela de login (e.g. no rodapé da tela). (**Prover Acesso**).

Problema 4. Há no Orion muitas imagens pesadas que são carregadas toda vez que se abre uma nova tela (Figura 10). Desta forma uma conexão lenta pode ter dificuldades para carregar as telas do OriOn e a interação com o ambiente se torna cansativa e desestimulante. Definir frames (HTML) para que certas partes do ambiente, com imagens pesadas, não sejam recarregadas novamente com a interação (e.g. a barra de título e a barra de opções lateral direita) pode ajudar. (**Prover Acesso**).

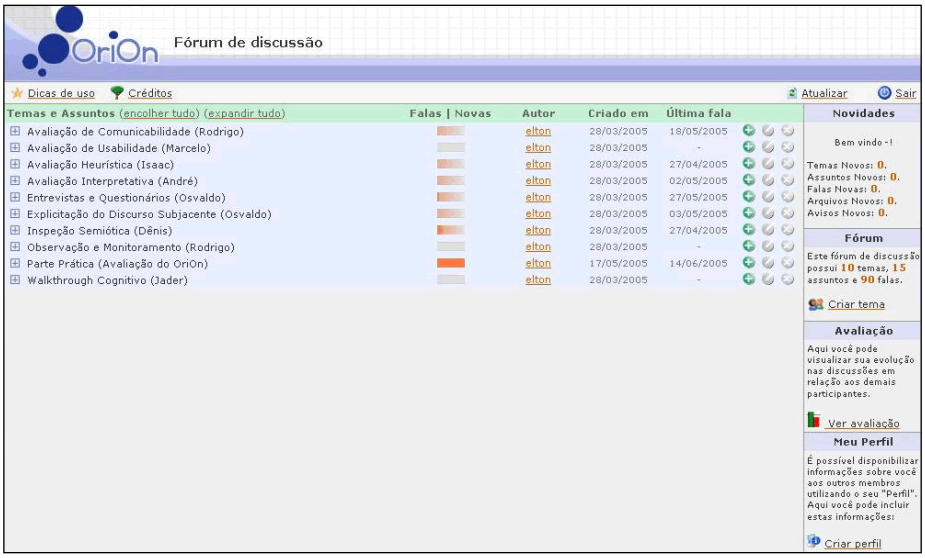


Figura 10 Tela de Temas e Assuntos.

Problema 5. Na página inicial e em outras páginas do Orion, alguns links têm a cor vermelha mesmo que nunca tenham sido visitados. Ex.: na página inicial, o link "cadastre-se aqui" é vermelho, mesmo que o usuário nunca tenha tentado se cadastrar (Figura 9). Colocar os links com cores padrões. Ex.: Azul para links nunca clicados, vermelho para links já visitados. (**Prover Acesso**).

Problema 6. Não existe um editor para a postagem de falas (Figura 11). Criar um editor completo com verificador ortográfico, barra de ferramentas, emoticos e menu de gestos para facilitar a postagem de falas e tornando a discussão mais agradável. (**Facilitar a comunicação**).

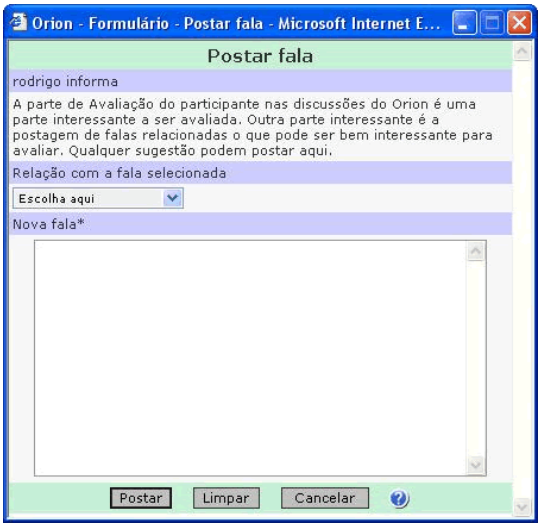


Figura 11 Tela para postar fala.

Problema 7. O OriOn possui uma única maneira de visualizar as discussões (Figura 12). Esse layout de apresentação das discussões pode agradar alguns membros, mas outros podem se sentir incomodados ou não gostar do mesmo layout. Para aumentar a proximidade entre um membro e o ambiente, poderia haver outros layouts disponíveis para personalização do OriOn. O membro deveria escolher o que mais lhe agradaria na tela de edição de perfil. Um tipo interessante de visualização é a

visualização em árvore que permite uma visão geral do andamento da discussão e facilita a identificação dos pontos onde a discussão está mais polêmica. (**Facilitar a comunicação**).

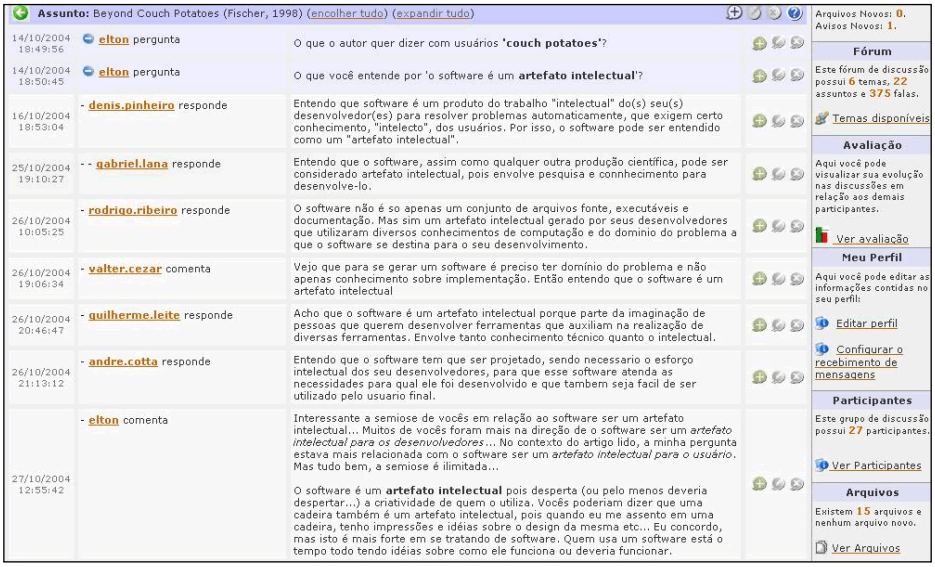


Figura 12 Visualização da discussão.

Problema 8. Há um mecanismo que avisa a um membro que existem novas falas (i.e. não lidas) e através da interação na tela de "Temas e Assuntos" o mesmo descobre em qual discussão que elas ocorreram, mas dentro da discussão não existe uma forma de distinguir falas novas de falas já lidas (Figura 13). Para resolver este problema, as falas novas poderiam ser apresentadas com um fundo de cor diferente, ou até mesmo, o texto em cor diferente. Isto para que esta fala se destaque em relação às outras. (**Facilitar a comunicação**).

Figura 13 Visualização da discussão, informação das novidades e sistema de busca.

Problema 9. Existe uma ferramenta para a filtragem da discussão, contudo, a ferramenta não dá muitas opções para o usuário (Figura 13). Criar uma ferramenta completa, para filtragem, utilizando técnicas mais especializadas, para esta função,

para facilitar a procura do usuário por certos termos em determinado período de tempo e por usuários específicos. (**Facilitar a comunicação**).

Problema 10. Não existe no OriOn um mecanismo de geração de resumo das discussões. Para que um novo membro obtenha uma leitura geral das discussões, este deve percorrer as discussões fazendo uma leitura das falas. Uma forma rápida de obter esta leitura geral seria muito interessante. Para uma solução parcial, a criação de uma ferramenta que forneça uma interface para a impressão amigável da discussão. (**Facilitar a comunicação**).

Problema 11. Não existe uma forma de motivação para que os usuários participem mais e melhor do Orion. A comunidade não tem uma pessoa que fica responsável por ficar motivando, ficar sempre observando e continuando o assunto tentando fazer com que os membros também participem. Promover premiações ou bonificações para os usuários que mais e melhor participarem, serve de estímulo para que o usuário sempre participe veementemente de forma séria e respeitosa. Também pode-se escolher um motivador, ou sempre ir revezando entre os membros. (**Motivar a comunidade**).

Problema 12. O usuário não consegue mudar sua senha, ou caso tenha esquecido não tem como recuperar, a não ser entrar em contato com o administrador, esperando até que ele possa passar uma nova senha. Fornecer uma opção segura para mudança e modificação de senha pelo usuário. (**Definir uma política de registro**).

Problema 13. Não existe no OriOn um mecanismo de registro on-line. Novos membros para obterem acesso devem procurar os gestores, administradores do ambiente para conseguir o acesso como membro. Definir um mecanismo de registro disponibilizado na tela de login do OriOn. Desta forma, novos usuários poderiam ser adicionados de forma mais dinâmica, mas é claro, somente com a liberação dos gestores e/ou administradores. (**Definir uma política de registro**).

Problema 14. Não existe uma política de privacidade. Os usuários não sabem quais são as políticas e regras quanto à privacidade no grupo. Quando o usuário cadastrar, ele terá que aceitar a política de privacidade, para que esteja sabendo como funciona o grupo e da punição aos usuários que desrespeitá-la. (**Garantir segurança**).

Problema 15. Não existe um mapa do site para facilitar a navegação dos usuários. Criar um mapa do site mostrando a estrutura do site para facilitar a navegação dos usuários. (**Navegação**).

Problema 16. Não existe nenhum mecanismo que delegue aos usuários uma participação na gestão do grupo. Os usuários simplesmente postam informações. Possibilitar aos usuários uma participação na gestão, como políticas de acesso pelos visitantes e decisões sobre membros que não contribuem nas discussões. Com a participação na gestão o usuário fica mais motivado a participar do grupo. Os usuários poderiam modificar a aparência do grupo em conjunto e algumas funcionalidades, para o grupo ter a "cara" de seus usuários. (**Participação na gestão**).

Problema 17. No OriOn, não existe um mecanismo explícito para definir quem está moderando ou gerindo as discussões. O simples fato de que um membro criou uma discussão, não impõe que esta irá moderar esta discussão. Não existe uma definição de papéis no OriOn (e.g., moderador, membro, visitante). Na atual situação, todos que tem acesso no OriOn são todos membros de igual poderes. A definição de tipos de usuários diferentes poderia solucionar esta questão. Poderia haver: moderadores, membros e visitantes. Os moderadores poderiam ser eleitos através de uma votação entre os membros do OriOn, de forma rotativa. Estes teriam a responsabilidade de

manter as discussões ativas, levantando questões, intimando os membros a participarem etc. Os membros poderiam: eleger os moderadores, tornar-se moderadores por um certo tempo e participar das discussões. Os visitantes somente poderiam visualizar as discussões, sem nenhum direito dentro do ambiente (e.g., postar falas, criar discussões etc). (**Participação na gestão**).

Analisando o número de problemas encontrados por cada avaliador, podemos construir o seguinte gráfico:

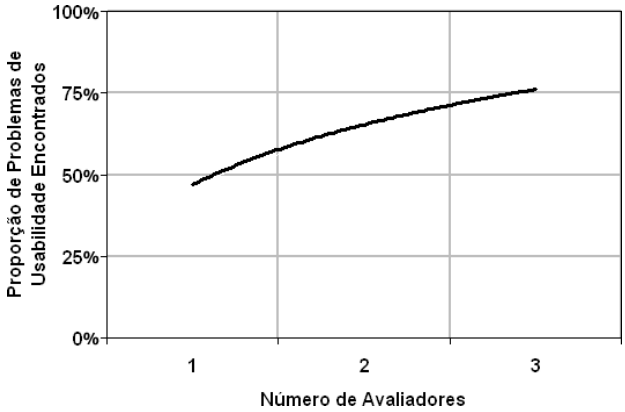


Figura 14 Proporção de problemas de usabilidade encontrados utilizando avaliação heurística no OriOn.

A Figura 14 corresponde com a curva da Figura 3, demonstrando que a avaliação heurística realmente necessita de poucos avaliadores (pelo menos três) para achar uma média entre 60-80% dos problemas da interface avaliada.

Além disso, analisando de perto quais problemas cada avaliador encontrou, podemos construir outro gráfico que também demonstra certa correspondência com os resultados da literatura, validando certos aspectos da avaliação heurística:

Figura 15 Cada linha representa um avaliador e cada coluna representa um problema de usabilidade. O quadrado preto representa se o problema foi encontrado pelo avaliador.

A Figura 15 comprova a nossa hipótese levantada na **Seção 1.2** de que pessoas diferentes encontram diferentes problemas de usabilidade. Os avaliadores ditos "*bem sucedidos*" acabam às vezes deixando os problemas "*fáceis*" para trás, enquanto que os avaliadores ditos "*malsucedidos*" também conseguem encontrar os problemas "*difíceis*" mesmo tendo encontrado menos problemas que os avaliadores em geral.

4. Conclusões

Os resultados de uma avaliação heurística vão ser muito melhores se não for utilizado apenas um avaliador, e se os avaliadores utilizados não mantiverem nenhum contato entre si. O número de problemas de usabilidade encontrados por um agregado de avaliadores cresce rapidamente no intervalo entre um e cinco avaliadores, mas esse crescimento diminui próximo a dez avaliadores. A partir do estudo de caso feito e dos

experimentos de outros autores, podemos concluir que a avaliação heurística pode ser realizada de forma satisfatória utilizando-se de três a cinco avaliadores.

As maiores vantagens da avaliação heurística são:

- Método de baixo custo;
- Método rápido;
- Método fácil de aplicar e de entender;
- Bom para descobrir que existem desfalques na interface;
- Bom para delimitar o escopo da avaliação.

As desvantagens da avaliação heurística são:

- Dependência das heurísticas para se realizar a avaliação;
- Ainda não foram definidas heurísticas para todos os tipos de interface;
- Heurísticas ruins ou difíceis levam a uma avaliação de baixa qualidade;
- Os avaliadores, geralmente, se limitam aos exemplos de problemas apresentados nas heurísticas, diminuindo o poder de alcance das mesmas;
- Não existe a preocupação com a interpretação do avaliador, só existe o erro se a interface não respeitar a heurística.

Esse estudo realizado sobre o método de avaliação heurística comprovou a sua eficiência, mediante ao baixo custo e curto prazo de tempo para se realizar uma avaliação com um número pequeno de avaliadores e encontrando um número satisfatório de problemas na interface.

O método é simples e fácil de aprender, porém a avaliação fica altamente dependente da utilização de boas heurísticas. Sendo que essas mesmas heurísticas podem prejudicar a qualidade da avaliação realizada pelos avaliadores, pois muitos avaliadores ficam presos apenas aos exemplos apresentados para uma determinada heurística, não explorando o seu verdadeiro potencial. Isso só comprova que a avaliação fica restrita as heurísticas e suas "sugestões", o que pode tornar a avaliação um tanto quanto limitada, pois não é levada em consideração a interpretação do avaliador em relação aos elementos da interface, levando erros ainda desconhecidos pela heurística a nunca serem descobertos.

Exercícios

1. Um sistema de resposta por voz, é um sistema computacional acessado através de um telefone. O sistema, chamado aqui de *Sistema Bancário*, é uma interface operada pelo telefone para acesso à conta do usuário. A tarefa do usuário no diálogo de exemplo foi transferir R$ 1.000 da poupança para a conta corrente do próprio usuário. As únicas opções de entrada do usuário são 12 botões iguais aos de um telefone comum (os dígitos 0-9 e os caracteres especiais * e #). A única resposta do sistema é através de fala e algumas vezes efeitos sonoros. A tecla # significa o fim da entrada do usuário. Enquanto o usuário não pressionou a tecla #, é possível pressionar duas vezes a tecla * para corrigir a entrada. Um exemplo de diálogo entre um usuário (U) e o *Sistema Bancário* (S) é mostrado a seguir:

(1) S: Entre um para informação sobre conta, três para transferências entre suas próprias contas, ...

(2) U: 3# *{o usuário interrompe o sistema}*
(3) S: Entre a conta de origem da transferência
(4) U: 1234567890# *{número da poupança}*
(5) S: Entre a conta destino da transferência
(6) U: # *{uma abreviação para conta corrente}*
(7) S: Entre com o valor em centavos
(8) U: 100000#
(9) S: Do número de conta doze, trinta e quatro, cinqüenta e seis, setenta e oito, noventa, para o número de conta, conta primária, uma transferência de um mil reais está para ser feita. Pressione um para confirmar, zero para cancelar.
(10) U: 1#
(11) S: Você não tem acesso para usar esta função.

Os números dos enunciados não fazem parte do diálogo, apenas são usados para referenciar enunciados individuais. Tente encontrar problemas de usabilidade utilizando o método de avaliação heurística através das nove heurísticas originais. **Dica:** é conhecido um total de 16 problemas **[Nielsen, 1992]**.

2. Sua tarefa é aconselhar uma companhia sobre a qualidade do diálogo humano-computador de um de seus sistemas. A administração da companhia quer ter certeza que usuários inexperientes conseguirão obter resultados rapidamente quando usarem o sistema. Com isso em mente, você deve apontar o máximo de problemas de usabilidade no diálogo utilizando o método de avaliação heurística através das nove heurísticas originais.

A funcionalidade básica do sistema é pré-determinada. O propósito deste exercício é criticar o diálogo do sistema e não sua funcionalidade. Novas funções possivelmente elevarão a usabilidade do sistema – mas sugestões para novas ou mudanças nas funções não fazem parte desse exercício.

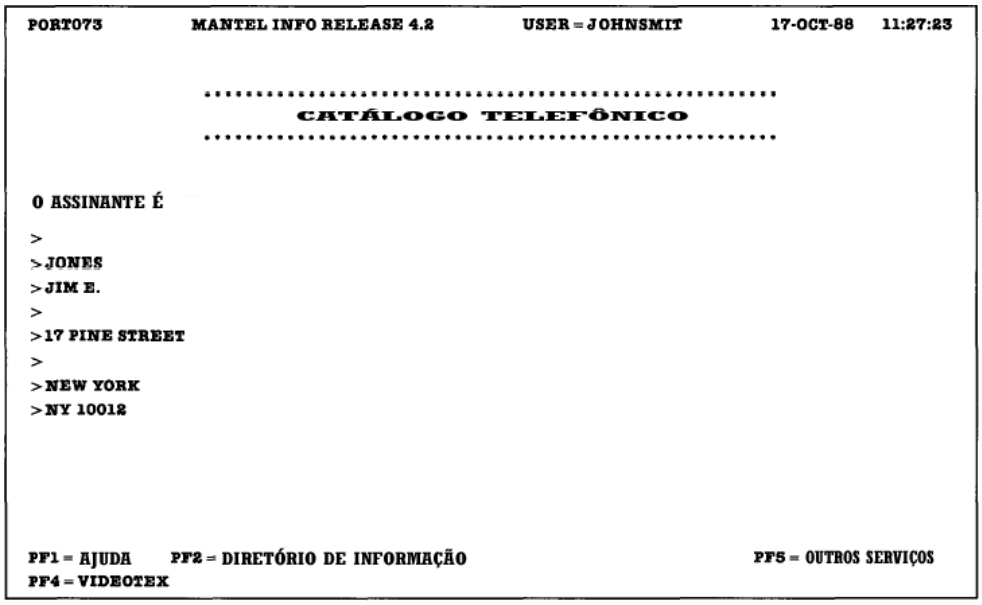
Sua solução deve consistir de uma lista com todos os problemas de usabilidade que você pôde encontrar no diálogo. Você pode incluir sugestões de como aperfeiçoar o diálogo para evitar os problemas e até pode especificar um diálogo aprimorado. Seu objetivo principal é articular os problemas de usabilidade identificados, do que meramente indicar para eles mudanças em um design alternativo.

O Sistema de Catálogo Telefônico

Este sistema é parte de um serviço da "Manhattan Telephone" (MANTEL)[2] para usuários domésticos. Usuários típicos têm pouco conhecimento sobre processamento de dados. Eles podem discar para o sistema, que proverá o nome e endereço de um assinante nos Estados Unidos, dado o número de telefone do assinante.

Para simplificar o exercício, nós fizemos as seguintes suposições. Para cada número de telefone, temos somente um assinante. Todos os números de telefone consistem de exatamente dez dígitos (3 dígitos para código de área e os outros 7 dígitos para o número). O computador do usuário possui um monitor alfanumérico, monocromático com 24 linhas de 80 caracteres cada e um teclado com teclas extras encontradas na maioria dos teclados, incluindo 10 teclas de funções marcadas como PF1-PF10. Uma tela é mostrada na figura abaixo:

[2] O nome "MANTEL" e o sistema foram inventados somente para o propósito deste exercício. Qualquer relação com companhias ou serviços de informações existentes é mera coincidência.



Especificação

O usuário entra neste sistema selecionando "Catálogo de Telefone" a partir do menu principal do MANTEL. O sistema então lança o seguinte prompt:

ENTRE O Nº DE TELEFONE DESEJADO E ENTER

Se o usuário entrar com qualquer coisa que não seja exatamente dez dígitos, o sistema responde:

NÚMERO ILEGAL. TENTE NOVAMENTE!

Se o usuário entra com um número de telefone que não está em uso, o sistema responde:

NÚMERO DE TELEFONE DESCONHECIDO

Se o código de área do telefone é 212 (código de área de Manhattan), o sistema irá normalmente mostrar a tela correspondente a figura anterior em cinco segundos. Para outros códigos de área, o sistema precisa recuperar a informação necessária de bancos de dados externos; isto pode durar cerca de 30 segundos.

Dica: são conhecidos 29 problemas. Para ajudá-lo a iniciar a avaliação, aqui está um dos problemas de usabilidade, bem como uma sugestão de como melhorar o diálogo: "O design da tela usa apenas letras maiúsculas, apesar de que nós sabemos que estudos sobre fatores humanos alertam que a mistura de letras maiúsculas e minúsculas é muito mais legível. Está OK usar letras maiúsculas para um número limitado de palavras que você queira enfatizar" **[Molich & Nielsen, 1990]**.

Referências Bibliográficas

[Baker et. al., 2001] Baker, K., Greenberg, S. e Gutwin, C. Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration. *Proceedings of the 8th IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI'01)*. (Maio 11-13, Toronto, Canada), 2001.

[Baker et. al., 2002] Baker, K., Greenberg, S. e Gutwin, C. Empirical Development of a Heuristic Evaluation Methodology for Shared Workspace Groupware. *Proceedings CSCW'02.* (Novembro 16-20, New Orleans, Louisiana, USA), 2002.

[Chan & Rocha, 1996] Chan, S. e Rocha, H. V. Estudo comparativo de métodos para avaliação de interfaces homem-computador. *Relatório Técnico* IC-96-05, 1996.

[da Silva et. al., 2003] da Silva, E. J., Nicolaci-da-Costa, A. M., de Souza, C. S., Prates, R. O. Expectativas de usuários potenciais a respeito de grupos virtuais de discussão.

[Jeffries et. al., 1991] Jeffries, R., Miller, J. R., Wharton, C., Uyeda, K. M. "User Interface Evaluation in the Real World: A Comparison of Four Techniques". Em *ACM CHI Conference Proceedings*, pp. 119-124, 1991.

[Lieberman, 2003] Lieberman, H. The Tyranny of Evaluation. MIT Media Lab. Disponível em: http://web.media.mit.edu/~lieber/Misc/Tyranny-Evaluation.html. Acessado em: 26 de Abril de 2005.

[Molich & Nielsen, 1990] Molich, R. e Nielsen, J. Improving a human-computer dialogue. *Communications of the ACM* 33, 3 Março, 338-348, 1990.

[Nielsen & Molich, 1990] Nielsen, J. e Molich, R. Heuristic Evaluation of User Interfaces. *Proceedings ACM CHI'90* (Seattle, WA, 1-5 Abril): 249-256, 1990.

[Nielsen, 1989] Nielsen, J. Usability engineering at a discount. Em G. Salvendy *et.al.* (eds.), *Designing and Using Human-Computer Interfaces and Knowledge Based Systems*. Amsterdan: Elsevier Science Publishers, 1989.

[Nielsen, 1992] Nielsen, J. Finding usability problems through heuristic evaluation. *Proceedings ACM CHI'92 Conference* (Monterey, CA, 3-7 Maio): 373-380, 1992.

[Nielsen, 1993] Nielsen, J. *Usability Engineering*. Academic Press, Cambridge, MA, 1993.

[Nielsen, 1994] Nielsen, J. Heuristic Evaluation. Em J. Nielsen (ed.) *Usability Inspection Methods*. John Wiley, New York, 1994.

[Preece et. al., 1994] Preece, J., Sharp, H., Benyon, D., Holland, S., Carey, T. *Human-Computer Interaction*, Addison Wesley, 1994.

[Preece, 2000] Preece, J. *Online Communities: Designing usability, supporting sociability.* John Wiley & Sons, Ltd.

[Rocha & Baranauskas, 2000] Rocha, H. V. e Baranauskas, M. C. C. *Design e Avaliação de Interfaces Humano-Computador*. São Paulo, 2000.

[Romani & Baranauskas, 1998] Romani, L. S. e Baranauskas, M. C. C. Avaliação heurística de um sistema altamente dependente do domínio. *Relatório Técnico* IC-98-26, Julho, 1998.