

Processo de Design em Interfaces

Ricardo Vilela Machado
Ciência da Computação
Universidade Federal Fluminense
Email: ricardovilela@id.uff.br

Thiago Henriques Cunha
Ciência da Computação
Universidade Federal Fluminense
Email: thiagocmd@hotmail.com

Resumo—Nos últimos anos cada vez mais as interfaces de usuário vem ganhando mais importância, seu papel é importante, seja atraindo o usuário, ajudando-o a cumprir tarefas mais eficientemente ou simplesmente, não confundindo-o e irritando-o. Esse texto, aborda algumas técnicas importantes do processo de design de interfaces de usuário, utilizando prototipação.

I. INTRODUÇÃO

As interfaces de aplicações tem recebido mais atenção nos últimos anos, quando a sua grande importância para aplicações tem sido notada. A interface de uma aplicação computacional envolve todos os aspectos de um sistema com o qual mantemos contato [Moran, 1981]. As funcionalidades de uma aplicação são fornecidas aos usuários através das interfaces. Um ótimo design de interfaces favorece em uma satisfação do cliente em usar o sistema, de eficiência no uso, na segurança ao se utilizar o sistema, no custo de se treinar um usuário e consequentemente o retorno de investimento.

O design de interfaces tem sido desenvolvido por processos iterativos na construção e avaliação, com usos de protótipos com métodos específicos para validação e verificação de requisitos. Muitas das vezes esses métodos são conflitantes para determinadas aplicações e para isso é importante e fundamental guiar o desenvolvimento de design de interfaces com uma fundamentação teórica[Hartson, 1998]. Essa fundamentação guiará o desenvolvedor durante a elaboração da solução para os problemas que serão resolvidos pela aplicação.

A área de Interação Humano-Computador (IHC) tem por objetivo principal fornecer aos pesquisadores e desenvolvedores de sistemas explicações e previsões para fenômenos de interação usuário-sistema e resultados práticos para o design da interface de usuário [ACM SIGCHI, 1992]. Com a utilização de formas padrão e com uso de teorias nas fases de desenvolvimento é possível adivinhar se o sistema resolverá as necessidades do usuário em relação a usabilidade, aplicabilidade e comunicabilidade. Para isto, estudos de IHC tem o objetivo de desenvolver modelos teóricos de desempenho e cognição humanos, assim como técnicas efetivas para avaliação da usabilidade [Lindgaard, 1994].

Existem propostas que pregam que aplicações devem procurar alcançar a aplicabilidade [Fischer, 1998] e comunicabilidade [de Souza, 1999], proporcionado ao usuário artefatos que sejam fáceis de se utilizar, aplicar e comunicar. Interface Humano Computador é uma área com várias disciplinas dentro de uma mesma área do conhecimento, que envolve disciplinas como [Preece et al., 1994]: Ciência da Computação; Psicologia Cognitiva; Psicologia Social e Organizacional; Ergonomia ou

Fatores Humanos; Lingüística; Inteligência Artificial; Filosofia, Sociologia e Antropologia; Engenharia e Design. Em relação a interface humano computador, temos que levar em consideração 4 elementos básicos, que são o sistema, os usuários, os desenvolvedores e o ambiente que será usado (domínio de alicação)[Dix et al., 1993]. Sabendo que estes elementos estão relacionados em 2 processos importantes, que são a interação usuário-sistema e o desenvolvimento do sistema.

II. PROCESSO DE DESENVOLVIMENTO DE SISTEMAS

O desenvolvimento de sistemas computacionais não parte da escrita de código em alguma linguagem de programação. O processo de desenvolvimento inicia-se pela análise do problema e necessidades do usuário ou cliente interessado na aplicação.

Após esta análise, é preciso explicitar os objetivos do sistema de forma que ele atenda as necessidades do usuário. Isso é feito através do levantamento de requisitos do sistema, seguido da análise e modelagem mais precisa do que o sistema vai apresentar, do projeto de implementação, codificação dos testes e sua implantação.

Para organização de todo o desenvolvimento, deve ser utilizado um modelo de processo de desenvolvimento, que contém todos os passos necessários para o desenvolvimento de software. Este modelo, deve ser escolhido de acordo com a necessidade do desenvolvedor e cliente, podendo, por exemplo, visar a agilidade do desenvolvimento ou focar na documentação do software. Alguns dos modelos existentes são:

O modelo cascata, onde todo o software é desenvolvido em uma sequência de passos. Modelo incremental, onde o software é dividido em partes e cada uma dessas partes são desenvolvidas separadamente.

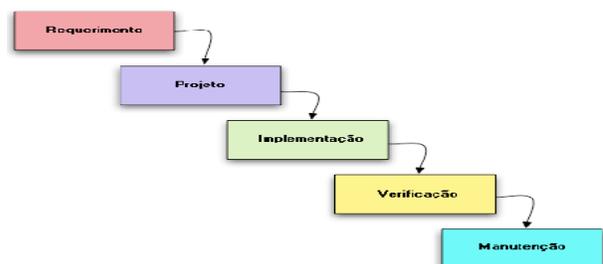


Figura 1. Modelo Cascata

E modelo espiral, onde o processo é representado por uma espiral, e cada loop dessa espiral representa uma fase do processo do software e que explicita rigorosamente os riscos de desenvolvimento (prazos e custos).

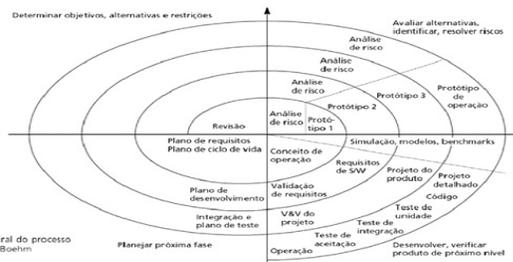


Figura 4.5 Modelo em espiral do processo de software de Boehm (IEEE, 1988)

Figura 2. Modelo Espiral

III. DESIGN

Design é uma área que é próprio a duas ou mais disciplinas, onde abrange atividades de concepção e projeto para algum produto novo, podendo ser um carro, um painel, um novo modelo para aparelho de celular, um computador, interface de usuário de uma loja virtual ou uma interface para sistema desktop. [3]

Podemos definir design como uma atividade que possui uma grande gama de aplicações e com isso necessita que o desenvolvedor, o projetista, possua uma atenção em relação aos aspectos funcionais e estéticos do produto, assim como exige bastante criatividade e habilidades na criação de modelos e nos acertos iterativos e re-design. [3]

“Você sabe que você conseguiu a perfeição em design, não quando você não tem nada mais a acrescentar, mas quando você não tem mais nada para tirar.”Analisando o pensamento de Antoine de Saint-Exupéry, vemos que é bastante difícil entender quando alcançamos a perfeição ou extatidão no design. Foi mostrado por Antoine que pode ser conseguida através da simplicidade, com algo mais simples que facilite o usuário manusear. [3]

A satisfação de um usuário e o sucesso da aplicação dependem da simplicidade. Um exemplo é o Google, que se tornou-se o mais usado site para buscas pelos usuários. Tem-se uma tela quase toda com a cor branca e com uma única janela para ser digitado a palavra da pesquisa, onde não é necessário fazer treinamento para se utilizar, pois foi projetado de forma bastante simples. [3]

Temos atualmente que os processos encontrados nas empresas necessitam que exista um suporte automatizado em relação as tarefas das pessoas de uma forma mais estruturada e esperada do trabalho e também existe a necessidade de interação. Para isso, precisa de equipamentos e de ferramentas para tornar as atividades mais simples de serem feitas pelos usuários e com isso diminuindo o tempo de trabalho. [3]

O design é uma atividade importante em relação as pessoas e determinante no sucesso do artefato. Possuir muitos detalhes e ser simples causa o sucesso de novos produtos, como exemplo: i-Pod da Apple. [3]

IV. DESIGN DE INTERFACES

Existem algumas abordagens de desenvolvimento de interface, a que será abordada nesse texto é baseada em prototipação, que tem o objetivo de demonstrar conceitos e opções para o desenvolvedor e cliente para que requisitos e ideias possam ser validados. As etapas do processo de prototipação são as definidas a seguir.

A. Análise de requisitos: análise de usuários e de tarefas

Os projetistas de interfaces tem como foco central o usuário no processo de design de interface. Sendo assim precisam definir quem são os usuários do software e suas tarefas, para que então, possam projetar interfaces com maior nível de aceitação e usabilidade por parte do usuário.

O objetivo da análise de usuários é identificar quem são os usuários e caracterizá-los, ou seja, especificar suas funções, nível de conhecimentos, desde o domínio à familiaridade com computadores, frequência de uso da aplicação, características pessoais e culturais. Abaixo estão as definições de alguns pontos que devem ser analisados:

Papel: Define qual função o usuário exerce, numa organização por exemplo, os cargos são definidos de forma estruturadas, tendo que uns tem uns podem ter acesso a alguma parte da aplicação que outro cargo inferior não tenha.

Familiaridade com computadores: Essa etapa tende a avaliar a experiência do usuário com outras aplicações, pode utilizar aplicações anteriores como base para desenvolver comandos e funções que já são de conhecimento do usuário, para uma melhor usabilidade do sistema.

Nível de conhecimento do domínio da aplicação: O usuário pode conhecer o domínio ou não, sendo assim, a interface pode ser projetada para dar dicas ao usuário sobre seu estado e ajudar o usuário na recuperação de erros.

Frequência de uso da aplicação: Define se o usuário é um usuário ocasional ou frequente. Um usuário ocasional, pode ter dificuldades para cumprir funções, o que aumenta a necessidade de ajuda e apoio. Já um usuário frequente, tem as necessidades de ajuda e apoio reduzidas com o passar do tempo.

Contexto sócio-cultural: Avalia fatores como linguagem, tradições culturais e comportamento. Um usuário que, por exemplo, fala português, pode ser brasileiro ou português, algumas palavras não são de uso comum ou tem significados diferentes nos dois países.

A análise de tarefas: Tem o objetivo de fornecer ao projetista, uma visão das tarefas que usuário precisa executar. As tarefas podem ser analisadas baseadas em cenários, que são narrativas textuais de procedimentos do usuário para cumprir determinadas tarefas.

Uma técnica de análise de tarefas baseadas em cenários é a técnica de questionamento sistemático, onde o objetivo dessa técnica é compreender tudo o que envolve o contexto que está sendo passado na narrativa. Nessa técnica são realizadas diversas questões sobre cada componente da narrativa, as respostas a esta questões servem de ponte entre uma ideia e outra, e revelam novas conexões cruciais para o entendimento

preciso da narrativa [1]. A técnica citada pode ser decomposta nos seguintes passos:

Geração de cenário: O cenário é gerado de acordo com um objetivo do usuário e os passos que ele deve tomar dentro do sistema, como por exemplo: “Eu quero sacar R100,00. Eu insiro o cartão do banco no caixa eletrônico, pressiono o botão de saque rápido, digito minha senha, retiro o dinheiro e o cartão” [1].

Elaboração da rede de proposições: Simplificação da narrativa com proposições, por exemplo: “o cliente insere o cartão”.

Análise: A partir das proposições, pode-se determinar o perfil das tarefas (ações e objetos) e dos usuários (agentes das ações) [1].

Questionamento sistemático: A partir de questionamentos de proposições anteriores, novas proposições podem ser formadas, num processo iterativo. Existem diversos tipos de questões, dentre as quais destaca-se:

Por que ...?: Relevam consequências e causas de eventos da narrativa.

Como ...?: Oferece maiores detalhes sobre determinados eventos.

O que é ...?: Revelam atributos de determinado objeto.

Então “isto” é/ocorre assim...?: Esse tipo de pergunta verifica se as proposições estão sendo entendidas de maneira correta. Essas perguntas tem resposta “sim” ou “não”.

B. Concepção

O objetivo dessa fase é com base nos dados obtidos na análise. Nessa fase são planejados os passos do projeto, é uma fase mais gerencial, que também envolve decisões de como o problema vai ser resolvido. Os desenvolvedores, nessa fase, devem determinar uma visão da solução do problema e seus limites, determinar tecnologias, alguns pontos chaves que devem ser levados em consideração e ter noção de custos, riscos e criar um cronograma para o desenvolvimento do projeto.

C. Especificação

A especificação da interface do usuário começa na definição dos requisitos feitos na análise do usuário e de suas tarefas, e a partir daí segue para a modelagem das tarefas do usuário e storyboarding. [1]

1) *Modelagem de tarefas:* Existem muitas técnicas de modelagem de tarefas, a seguir estão descritas algumas:

TAG (Task-Action Grammar): gramática gerativa, que especifica ações a partir das tarefas. Essa técnica identifica tarefas simples, as representa em categorias e através de regras de produção. [1]

UAN (User Action Notation): utilizado principalmente para interfaces de manipulação direta, onde representa aspectos do comportamento do sistema do ponto de vista do usuário, ou seja, que tarefas e ações o usuário realiza na interface. Este modelo associa as ações do usuário ao feedback do sistema, ao estado do sistema e ao modelo computacional da aplicação [1].

GOMS (Goals, Operators, Methods and Selection Rules): representa o comportamento dinâmico da interação com o computador. Com base num modelo do comportamento humano que possui três sistemas de interação: perceptual, motor e cognitivo. [1]

Modelo Keystroke-Level: faz parte da família GOMS de modelos, porém, utiliza um nível mais baixo, o nível de atividade motora. Visando o tempo do usuário concluir uma tarefa. São considerados pequenas ações, como: o clicar de um botão, realizar uma tarefa mental ou esperar a resposta do sistema. [1]

2) *Storyboarding:* É uma técnica parecida com a representação por cenários, porém, não usa narrativas e sim desenhos e ilustrações para representação de funções, ações e situações de uma interface de usuário. Essa técnica é importante para a validação de cenários e elaboração protótipos de interfaces de usuário. Também é importante para representar ideias e fazer demonstrações, uma vez que, representações gráficas são mais rapidamente captadas que representações textuais. Abaixo está um exemplo de storyboard que descreve uma janela de um sistema de biblioteca. [1]

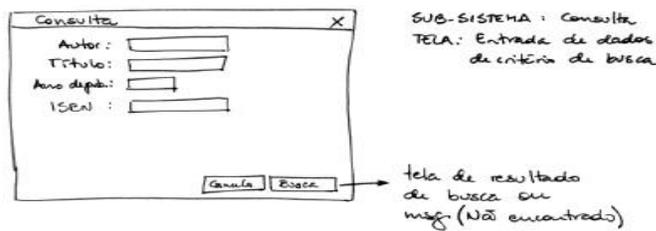


Figura 3. Representação de um Storyboarding

D. Prototipagem

Difícil é fazer uma avaliação do sistema que está sendo desenvolvido sem o ter implementado. Versões do sistema são criados pelos desenvolvedores através da prototipagem, sendo que essas versões não tem todas as funcionalidades criadas [4].

Há protótipos que são verticais, onde apenas alguns recursos do sistema são implementados e com isso é feito o teste em condições reais. Existe também os protótipos que são ditos horizontais, onde toda a interface é desenvolvida, porém sem qualquer funcionalidade possível para se testar em condições reais. Protótipo intermediário também podem ser criados, onde apenas uma parte das funcionalidades no sistema podem ser usadas pelos usuários [4].

A prototipagem se caracteriza pela rápida e econômica forma para um projeto ser definido e experimentado. A importância da prototipação é garantida por esses dois motivos. Porém sistemas são formados antes de qualquer esboço [4].

Embora o levantamento de requisitos que os sistema irá possuir contenha muitas informações que estarão presentes na interface, é na interação direta com o usuário que os detalhes importantes serão notados. Para que isso aconteça, faz-se necessário o uso de protótipos para fazer a simulação de interação, onde deve ser feita já no momento da concepção do sistema, para que assim seja evitado gastos maiores tanto de

tempo e dinheiro, que é mostrado pela frase de Jakob Nielsen, "As maiores melhorias na interface de um produto são obtidas através da recolha de dados de usabilidade nas fases iniciais de seu desenvolvimento."

É indagado por vários desenvolvedores sobre começar o desenvolvimento do projeto pela parte visual e não pela funcional. A interface é o principal meio que faz a comunicação do sistema e o usuário, oferecendo a melhor experiência possível ao usuário. Devendo possibilitar ao usuário a mensagem de forma clara e objetiva para satisfazer as suas necessidades sem gerar dúvidas [4].

É importante que seja estudada desde a parte inicial de concepção de um sistema. Vai ser a interface que irá definir e exprimir a interação do software com o usuário ou vice versa. Logo a programação deverá se basear no visual e não o visual se basear na programação [4].

Temos que maiores são os impactos na realização de correções e alterações no código quando este está bem avançado. Demora mais tempo alterando a programação para alterar determinada funcionalidade do que fazendo um protótipo várias vezes. Logo o tempo que é gasto na prototipagem é inversamente proporcional ao tempo que é gasto na programação, quanto mais protótipo é tratado com importância menos se gastará na implementação [4].

Os protótipos podem ser classificados de 3 formas, Protótipos de Baixa Fidelidade, Protótipos de Média Fidelidade e Protótipos de Alta Fidelidade.

1) *Protótipos de Baixa Fidelidade:* Esse tipo de prototipagem podem ser chamados de rascunhos também. São criados na fase inicial, já na concepção do sistema. Forma mais simples, quando são geralmente criados com uso de lápis, borracha e papel, sendo que é uma maneira mais rápida e superficial para apenas se ter uma ideia básica do projeto, não entrando em detalhes de layout, cores e disposições. Fundamental para definir o produto e levantar seus requisitos [4].

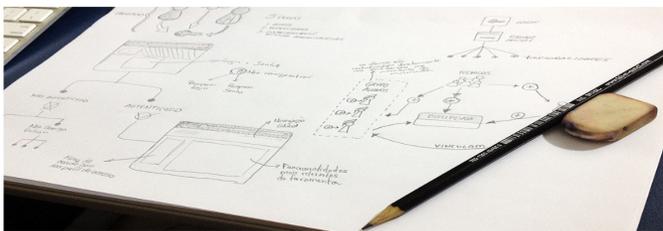


Figura 4. Representação de baixa fidelidade

2) *Protótipos de Média Fidelidade:* Esse tipo de prototipagem podem ser chamados de wireframes também, e são criados durante a fase da arquitetura dos dados. Pode-se fazer uso de lápis e papel ou uma ferramenta case que possa auxiliar na prototipação, sendo que esses documentos possuem a estrutura da interface e seu conteúdo, definindo a importância e relação que os elementos vão possuir, gerando por exemplo o layout de forma simples do projeto [4].

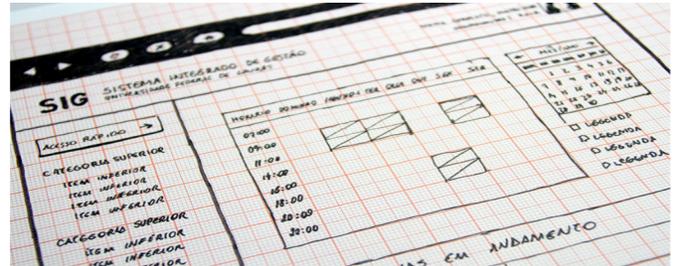


Figura 5. Representação de média fidelidade

3) *Protótipos de Alta Fidelidade:* Esse tipo de prototipagem constitui a representação que é mais exata do sistema que será criado. Sendo que para alguns casos é possível simular o fluxo total que as funcionalidades terão, possibilitando a interação do usuário como se fosse já produto final. Sendo que a aparência visual, as formas de navegação e interatividade já são criadas e usadas nos protótipos de alta fidelidade [4].

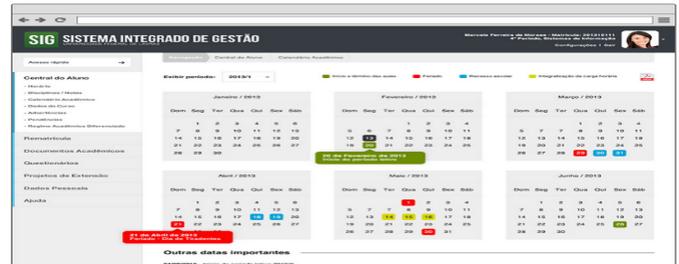


Figura 6. Representação de alta fidelidade

E. Avaliação

Uma importante tarefa do processo de design é a avaliação de interfaces, pois com essa tarefa é conseguido avaliar o sucesso ou fracasso com relação a solução que está sendo desenvolvida pela sua funcionalidade e interação. Ainda que o designer se baseie em uma abordagem teórica e conte com a ajuda de diretrizes e princípios de design, é necessário que ele avalie o resultado obtido [Hartson, 1998] [5].

Essa tarefa pode ser classificada como formativa ou somativa [Preece et al., 1994; Hartson, 1998]. Formativas são feitas no processo de design, podendo que um problema seja identificado e consertado antes do final da aplicação ser finalizada ou antes de sua implementação. Já as somativas fazem a avaliação do produto quando este está terminado. Há diferentes métodos para fazer a coleta e análise da informação, dependendo de cada contexto. Como por exemplo a disponibilidade de pessoas que são especialistas, ambientes e ferramentas para se aplicar e o próprio usuário. [5].

A grande maioria dos meios de avaliação são observando e monitorando o usuário, fazendo coleta de opinião de cada

usuário, fazendo experimentos. A interpretação das interações do usuário com o sistema pode ser avaliada ou ainda com predição do uso que pode ser feito. [5].

Observação e monitoração de usuários: Geralmente é feita na forma informal, ou no ambiente de trabalho que o usuário possui ou também em um laboratório, e as informações são: coletadas através de notação do observador ou algum tipo de gravação, como exemplo de vídeo. [5].

Coletar a opinião dos usuários: Fazer um levantamento da opinião dos usuários é tão importante quanto avaliar o seu desempenho, uma vez que se os usuários não gostarem da aplicação por qualquer razão, eles não a usarão. [5].

Aplicação de experimentos: Com o uso de experimentos geralmente adotam um modo semi-científica. Isto porque as variáveis a serem analisadas envolvem interações complexas com os seres humanos e a informação que é conseguida em alguns casos é questionada. Logo, este tipo de avaliação se refere a técnicas mais rigorosamente controladas do que em observação e monitoração de usuários. [5].

Métodos interpretativos: Possui como objetivo permitir que o desenvolvedor compreenda melhor como usuários usam o sistema em seu ambiente natural e como estes sistemas estão integrados com outras tarefas. Logo, a informação são coletadas de modo informal e mais tarde interpretadas pelos desenvolvedores. Em muitos casos os usuários participam dessa coleta, análise ou interpretação da informação. [5].

Métodos de predição: A predição busca prever os tipos de problemas que os usuários vão ter quando usar o sistema, mas não testando com os usuários realmente. Geralmente esse tipo envolve técnica de modelagem psicológicas dos usuários ou métodos de inspeção, onde os especialistas em interfaces estudam a aplicação em relação aos problemas que os usuários podem possuir. [5].

Testes de usabilidade: O objetivo que esse teste possui é fazer a medição quantitativamente do valor a ser alcançado pelo sistema em cada um dos fatores de usabilidade de interesse. A engenharia de usabilidade é uma das técnicas mais conhecidas para testes de usabilidade.[Nielsen, 1993]. Permite que seja aplicado procedimento sistemático para se avaliar a usabilidade da aplicação no seu desenvolvimento. Para isso, os desenvolvedores devem definir quais são os fatores de usabilidade que são prioritários e é definido um valor quantitativo para cada um desses aspectos. No desenvolvimento da aplicação, esses valores são analisados para verificar se os objetivos foram atingidos. [5].

Testes de comunicabilidade: O método de avaliação da comunicabilidade [de Souza et al., 1999] de um software é baseado na Engenharia Semiótica e tem como objetivo avaliar a sua interface com relação à sua propriedade de comunicabilidade. [5].

Para isto, este método propõe um conjunto de interjeições que o usuário potencialmente pode usar para se exprimir em uma situação onde acontece uma ruptura na sua comunicação com o sistema. Estas interjeições de fato não são direcionadas à aplicação, mas sim ao seu designer. [5].

Nesse método existem duas etapas, que são a coleta de dados e a análise dos dados. Na coleta, temos os passos:

- 1) Solicitar ao usuário a execução de uma tarefa pré-determinada na aplicação.
- 2) Gravar a interação do usuário com a aplicação, usando para isto um software que captura as ações do usuário, como por exemplo o Lotus ScreenCam ®. Anotações do aplicador do teste e gravação em vídeo podem ser feitos para enriquecer os dados.
- 3) Entrevista com o usuário (opcional) sobre a sua interação com a aplicação.

Feito a coleta das informações, é feito a análise das informações que foram levantadas.

- 1) Ver gravações da interação e atribuir a interjeição apropriada nos momentos de ruptura da interação.
- 2) Tabular a informação obtida, ou seja, as interjeições obtidas.
- 3) Interpretar a tabela de acordo com as interjeições e os problemas de usabilidade associados a elas, obtendo então um mapa dos pontos críticos da interação e um perfil da interação da aplicação.

As interjeições que são escolhidas formam um conjunto que seja capaz de mostrar o bloqueio da interação do usuário com a aplicação quando o sistema está sendo usado, com expressões naturais do cotidiano dos usuários utilizadas nessas situações. O conjunto de interjeições com seus significados e suas ações que possibilitam conferir uma interjeição a situação específica são:

Cadê? / E agora?: Usuário procura em menus e toolbars por uma função específica que ele deseja executar. No caso do “E agora?” o usuário não sabe o que fazer e tenta descobrir qual o seu próximo passo. Sintomas: Usuário inspeciona menus, sub-menus e tooltips (dicas) sem, no entanto, executar nenhuma ação. [5].

Que é isso?: O usuário tenta descobrir o que significa um objeto ou ação da interface. Sintomas: Usuário coloca cursor sobre algum símbolo da interface esperando um tooltip, ou procura o help daquele símbolo, ou ainda hesita entre duas opções que lhe pareçam equivalentes. [5].

Epa! / Onde estou?: O usuário executa uma ação que não era a desejada e imediatamente o percebe, desfazendo então a ação. No caso do “Onde estou?” o usuário, sem perceber, executa ações que apropriadas para outros contextos, mas não para o que ele se encontra. Sintomas: Usuário executa uma ação e em seguida a desfaz. [5].

Por que não funciona? / Ué, o que houve?: A ação executada não obtém o resultado esperado, no entanto, o usuário não entende porque este resultado não foi alcançado. Assim, ele insiste, acreditando que ele tenha cometido algum erro na execução da ação. No caso do “Ué, o que houve?” o usuário não tem feedback do sistema e não consegue entender o resultado da sua ação. Sintomas: Usuário executa uma ação e não percebe, entende ou aceita o resultado. Ele então repete os mesmos passos para conferir o resultado. [5].

Para mim está bom...: O usuário obtém um resultado que ele acredita ser o desejado, mas que não o é. Sintomas: Usuário dá a tarefa por terminada sem, no entanto, perceber que não alcançou o resultado desejado. [5].

Não dá.: O usuário não é capaz de alcançar o objetivo proposto, ou porque os recursos (tempo, paciência, informação desejada, etc.) não estavam disponíveis, ou porque ele não sabia como. Sintomas: Usuário abandona a tarefa sem ter conseguido atingir seu objetivo. [5].

Deixa pra lá... / Não, obrigado.: O usuário não entende as soluções de interação primárias oferecidas pelo designer, e resolve seu problema de alguma outra forma. No caso do “Não, obrigado.” ele entende a solução, mas prefere outras formas de interação. Sintomas: Usuário não encontra a forma de interação principal oferecida para se executar uma ação, ou então decide não usá-la. Note-se que este caso tipicamente segue uma interjeição de “Onde está?” [5].

V. CONCLUSÃO

O processo de design de interfaces com seus conceitos e práticas proporciona que aplicações criadas pelos desenvolvedores atendam as reais necessidades dos usuários finais. Como vimos, o processo é dividido em 5 etapas, onde em cada etapa é usado técnicas específicas para o levantamento, análise e validação dos requisitos. Vimos que o uso de protótipos proporcionam um feedback importante entre o desenvolvedor e o futuro usuário da aplicação, ressaltando pontos antes não observados pelos projetistas, as dificuldades de execução do sistema e diminuição do custo e tempo sem o retrabalho para modificação de incoerências. Sendo que os protótipos podem ser criados já nos estágios iniciais do desenvolvimento.

REFERÊNCIAS

- [1] S. de Sousa, Clarisse. C. Leite, Jair. O. Prates, Raquel. D. J. Barbosa, Simone; *Projeto de Interfaces de Usuário - Perspectivas Cognitivas e Semióticas*. Disponível em http://www.dimap.ufrn.br/jair/piu/JAI_Apostila.pdf Acesso em 26 de abr de 2014.
- [2] Eclipse Foundation (2014, 26 de abr de 2014) Fase de Concepção. Disponível em: http://epf.eclipse.org/wikis/openuppt/openup_basic/guidances/concepts/inception_phase_0hmKgBOMEduCNqgZdt_OaA.html.
- [3] Filho, A. M. da S. Design de Interfaces: foco no usuário e simplicidade. Disponível em <http://www.espacoacademico.com.br/076/76amsf.htm>. Acesso em 27 de abr de 2014.
- [4] Nascimento, T. A importância dos protótipos no desenvolvimento de sistemas. Disponível em <http://thiagonasc.com/tag/interface>. Acesso em 27 de abr de 2014.
- [5] de Souza C. S. Prates R. O. Leite J.C. Simone. D.J. Barbosa Projeto de Interfaces de Usuário. Perspectivas Cognitivas e Semióticas. Disponível em http://www.dimap.ufrn.br/jair/piu/JAI_Apostila.pdf. Acesso em 26 de abr de 2014.