

**Técnicas para requisitos funcionais de Interfaces Homem Computador com recursos em elicitación**

**Techniques for functional requirements of Human Computer Interfaces with resources in elicitation**

**Juliano Schimiguel**

Universidade Cruzeiro do Sul, Brasil

E-mail: schimiguel@gmail.com

**Leonardo de Fabris Lopes**

Centro Universitário Salesiano de São Paulo, Brasil

E-mail: profleolopes@gmail.com

**Cecilia Sosa Arias Peixoto**

Centro Universitário Salesiano de São Paulo, Brasil

E-mail: cecilia.sosaarias@gmail.com

**Carlos Adriano Martins**

Universidade Cruzeiro do Sul, Brasil

E-mail: ead.adriano@gmail.com

Recebido: 26/06/2017 – Aceito: 01/11/2017

**Resumo**

A Interface Homem Computador (IHC) é parte fundamental nos critérios de aceitação de um software, na determinação de seu tempo de vida e no interesse que o mesmo despertará em seus usuários. Apesar de sua importância, as fábricas de *software*, em geral, pouco se aprofundam em análises e investimentos na área de IHC, tendo seu foco profissional em regras de negócio e funcionalidades implementadas. O objetivo é utilizar o uso de papel e caneta com as técnicas já conhecidas e reduzir o tempo de elaboração das ideias e elicitación de requisitos de IHC. A metodologia adotada para o desenvolvimento deste trabalho foi a revisão bibliográfica e análise a partir de um estudo de caso. O resultado obtido demonstra que as técnicas existentes na literatura são adequadas para realizar o correto levantamento dos requisitos e que, quando combinadas da forma proposta, os requisitos podem ser avaliados antes da fase de codificação, economizando tempo e recursos.

**Palavras-chave:** Interface humano computador, Análise de requisitos, Análise de usabilidade, Designer, Protótipo.

## **Abstract**

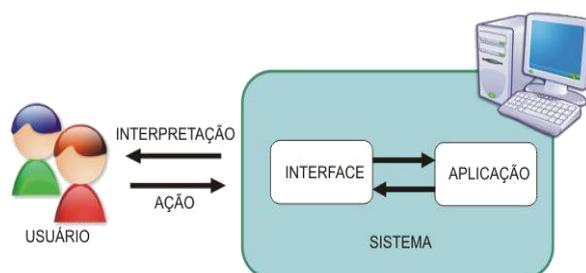
The Computer Human Interface (IHC) is a fundamental part of the acceptance criteria of a software, determining its life time and the interest that it will arouse in its users. In spite of their importance, the software factories generally do not go much further in analyzing and investing in the IHC area, having their professional focus on business rules and functionalities implemented. The goal is to use the use of paper and pen with the techniques already known and reduce the time of elaboration of ideas and elicitation of IHC requirements. The methodology adopted for the development of this work was the bibliographic review and analysis based on a case study. The obtained results demonstrate that the existing techniques in the literature are adequate to carry out the correct survey of the requirements and that, when combined in the proposed way, the requirements can be evaluated before the coding phase, saving time and resources.

**Keywords:** Human interface computer, Requirements analysis, Usability analysis, Designer, Prototype.

## **1. Introdução**

A Engenharia de *Software* (ES) tem como principal motivação à especificação, desenvolvimento e manutenção de sistemas. Através do uso de modelos abstratos e precisos, a ES permite a um engenheiro modelar um sistema de informação, acompanhar todos os passos do desenvolvimento de um *software* e produzi-lo de maneira econômica, com qualidade, de modo que atenda aos requisitos do cliente (PRESSMAN, 2005). Uma das demandas mais presentes hoje em dia consiste em desenvolver sistemas centrados nos usuários (GAMA et al, 2016), focando, principalmente, na usabilidade do sistema (CAETANO et al 2016; SANTOS et al, 2017). O conceito da “interface” é comumente abreviado na literatura como camada de apresentação do sistema. Segundo Laurel (1990), uma interface é uma superfície de contato que reflete as propriedades físicas das partes que interagem, as funções a serem executadas e o balanço entre poder e controle. A Figura 1 apresenta os componentes da interface, o *software* da interface, o *software* da aplicação, a interpretação do usuário e a ação.

**Figura 1 – Processo de interação Homem-Computador**



Fonte: elaborado pelos autores

No processo apresentado na Figura 1, é de extrema importância que não existam ruídos no diálogo entre usuário e sistema. Os ruídos são tudo o que o sistema pode entender de forma errada à requisição do usuário, ou as informações que o requerente não receba do sistema por algum motivo. A literatura sobre o tema IHC cita a importância do diálogo, recomendando que ocorra de maneira clara e coesa. Para que isto seja possível, diversos ciclos de vida, técnicas e métodos são empregados a fim de auxiliar aos projetistas de interfaces durante a especificação e o desenvolvimento.

Tem-se observado que os métodos convencionais da ES, utilizados na área de Interfaces, não têm levado a resultados satisfatórios devido a que não fornecem guias suficientes para os sistemas interativos. Este trabalho propõe tornar mais ágil o processo de levantamento de requisitos de IHC, com foco no estudo de usabilidade, abrangendo também os outros requisitos de *software*. Segundo Sommerville, (2011), um dos pressupostos ágeis é minimizar a documentação, pois se utiliza mais a comunicação informal do que reuniões formais com documentos escritos. Isto garante entregas mais rápidas e fase de desenvolvimento menos onerosa. O objetivo deste trabalho é utilizar o uso de papel e caneta com as técnicas já conhecidas com o escopo de reduzir o tempo de elaboração das ideias e elicitação de requisitos de IHC pela equipe de *designers* e sua comunicação para o cliente.

Este artigo apresenta as técnicas complementares de elicitação de requisitos fundamentais para evitar os "ruídos", ambiguidades e inconsistências na fase de definição do sistema (seção 2). A seção 3 apresenta um estudo de caso onde foi utilizada a técnica de Snyder (2003) e a seção 4 apresenta a conclusão do trabalho.

## **2. Técnicas complementares de elicitação de requisitos**

Nesta seção são apresentadas técnicas complementares para elicitação de requisitos de

um sistema interativo.

A técnica de Mágico de Oz (KELLEY, 1985), consiste em simular efeitos de computador, utilizando raciocínio humano. Isto é, uma pessoa faz o papel da máquina devolvendo a resposta para o usuário.

A técnica pode ser utilizada no lugar de algoritmos complexos ou na avaliação de diversos modos de interação como, reconhecimento de voz, e outras diversas tarefas em que seja dispendioso utilizar sistemas informatizados (BARBOSA e SILVA, 2010).

É importante para a correta utilização da técnica, um estudo sobre o escopo da encenação. Este estudo deverá discutir quais são os cenários suportados pelo sistema, quais funcionalidades serão encenadas, e quais as saídas esperadas para os mais diversos cenários. O ser humano é capaz de além de encenar responder de forma assertiva a uma entrada inesperada. Para que a representação seja o mais semelhante possível ao resultado computacional, que irá a substituir a cena em alguma fase do projeto, um roteiro deve ser fornecido ao ator que representa o computador (BARBOSA E SILVA 2010). Desta forma a técnica de Magico de Oz permite descobrir durante a sua aplicação se o usuário concorda com as cenas ou não, se existem problemas no diálogo e fundamentalmente elicitam os requisitos, verificar a usabilidade e viabilidade de forma econômica.

Outra técnica muito utilizada é de “*Mockup*” consiste em representar graficamente as telas do sistema (BARBOSA E SILVA, 2010), (NIELSEN, 1993). O objetivo dos *Mocks* é semelhante ao dos protótipos, isto é, ser um objeto de comunicação e gerar *feedback*. O escopo de criação de *Mocks* é menor em relação a protótipos, já que nenhuma funcionalidade é implementada, apenas o layout das telas do sistema é concebido. Desta forma o tempo despendido para a criação de *Mocks* tende a ser menor do que o aplicado na elaboração de protótipos.

Estas representações podem ser criadas em níveis incrementais. A primeira versão, classificada como de baixa fidelidade pode ser desenhada a mão livre em papel, a próxima versão feita de forma rápida no computador com o auxílio de uma ferramenta, poderá ser classificada como de alta-fidelidade dependendo da semelhança com um *software* real, e por último, a versão sendo desenvolvida em linguagem de programação, contendo apenas os componentes de GUI (*Graphical User Interface*) (SNYDER, 2003), (BARBOSA E SILVA, 2010), (NIELSEN, 1993). Entre as ferramentas de suporte à criação de *Mocks* podemos citar Balsamic (2014), uma ferramenta paga para desenvolvimento ágil de interfaces com estilo rascunho. Outra ferramenta, a *Just Mind* (JUSTMIND, 2014) com versão *on line*, tem semelhança com um ambiente de desenvolvimento, o que pode dificultar, visto que todos os

componentes possuem muitas opções e atributos. A ferramenta oferece um editor de regras de negócio, no qual é possível adicionar controle, como por exemplo, um botão tomar uma decisão baseada no estado e valores selecionados nos componentes. É possível desta forma, simular o funcionamento do sistema, a navegação entre telas mesmo que de maneira pré-definida pode se criar um roteiro e apresentar o sistema dentro deste contexto semelhante à cena criada na utilização da técnica de Magico de Oz, porém de forma automatizada. Através da apresentação dos *Mocks*, os usuários podem refletir sobre as funcionalidades do sistema permitindo a elicitación dos requisitos.

A técnica de quadro de histórias Snyder (2003) é uma forma de elicitar e comunicar requisitos funcionais e não funcionais com os *stakeholders* do projeto e os projetistas do sistema. A documentação das propriedades e comportamento do sistema é feita através do registro de uma história de uso do sistema. Ela pode envolver apenas uma funcionalidade ou um conjunto de funcionalidades correlacionadas para atingir um objetivo dentro do *software* que está sendo especificado. A representação gráfica desta história não tem a necessidade de ser esteticamente perfeita ou possuir uma concepção artística (SNYDER, 2003).

Uma forma eficaz de formular as histórias em quadros é a técnica denominada por Klemmer *et al* (2006) de representação em estrela. Os usuários são representados na história em uma forma que se assemelha em uma estrela (Figura 2).

**Figura 2 – Exemplo do uso da técnica de representação em estrela**



Fonte: elaborado pelos autores

Esta forma de representação de baixa fidelidade não necessita curva de aprendizado ou habilidade para ser reproduzida. Desta forma, mudando o preenchimento do interior do desenho do personagem, é fácil reproduzir diferentes usuários (SNYDER, 2003).

Cada quadro da história, a começar pelo quadro inicial, retrata um passo para completar uma tarefa dentro do sistema. Cada etapa é representada em seu próprio quadro, de

forma sequencial, semelhante à forma como são diagramadas as histórias em quadrinhos. Todos os passos chaves para se atingir um objetivo dentro do sistema serão retratados. A técnica sempre ilustra o caminho de sucesso da aplicação e tem o objetivo de elicitar os requisitos e verificar se o sistema é fácil de usar (SNYDER, 2003).

O primeiro quadro a ser retratado é o “gatilho” para a tarefa especificada. Deve responder a pergunta ”Porque o Sr. Estrela deseja realizar esta tarefa?”. Assim, são colocados no quadro, os demais usuários envolvidos, o sistema que será acessado, o ambiente, e o que motivou o acesso. Este é um diferencial desta forma de documentação. Ela especifica de forma rápida e simples o contexto, do usuário quando a ação foi realizada, o tipo de usuário, suas motivações e até mesmo o sentimento que ele tem em relação a aplicação (animação, ansiedade, preocupação). Com este tipo de protótipo, é possível comunicar ideias ao cliente, equipe e demais *stakeholders* do *software* (SNYDER, 2003).

Uma das técnicas amplamente aconselhadas e utilizadas são os protótipos de papel (NIELSEN, 1993), (SNYDER, 2003). Snyder (2003) propõe uma forma de elicitação de requisitos através da criação de protótipos em papel dispondo apenas de materiais de escritório como papéis, canetas coloridas, tesouras, durex e cola. Segundo Snyder (2003), esta técnica é diferente de desenho ou esboço de telas em papel, como as *MockUps*. O autor define o protótipo como um conjunto de telas mais funcionalidades representadas por desenhos à mão livre. É proposto que nas fases anteriores a codificação e documentação os *stakeholders*, possam realizar uma reunião com o intuito de discutir e esboçar o sistema realizando desenhos a mão livre das telas que irão compor o sistema. A Figura 3 apresenta um usuário testando um protótipo em papel.

**Figura 3 – Usuário utilizando um protótipo em papel fonte Site UseIt**



Fonte: USEIT (2012)

Durante o teste, alguns pontos podem ser verificados. Por exemplo, se os componentes

utilizados na montagem da tela permitem atingir o objetivo proposto, se o comportamento sistêmico da funcionalidade está claro a todos os participantes, se existem problemas de *layout* na forma como os componentes foram organizados na tela e se algum requisito funcional não foi levantado. As informações provenientes da reunião entre os participantes servem para a especificação dos requisitos do sistema (SNYDER, 2003).

Um debate à respeito das alternativas de *design* pode ser fomentado, os programadores poderão induzir a escolha de soluções tecnológicas que lhes tragam alguma vantagem durante fase de desenvolvimento, seja motivado pela facilidade de implementação que um *framework* pode oferecer ou outra razão dependendo das características do projeto. A opinião dos usuários tem um peso maior de decisão do que a opinião dos analistas (chamado de desenvolvimento centrado no usuário pela ISO 13407 (*International Organization for Standardization*)). (SNYDER, 2003).

Com a devida maturidade, que pode ser adquirida pela repetição constante da prática, os participantes poderão desenvolver um alto nível de abstração, sendo capazes, até mesmo, de prever problemas que só poderiam ser observados por uma equipe de teste após alguma entrega do *software*. Segundo Netto (2010), em geral, 30% das funcionalidades atendem a 70% dos usuários. Esta proporção também é mencionada nos processos de desenvolvimento ágeis, de forma que encontrar as funcionalidades que representam estes 30% podem garantir o sucesso de um projeto (SNYDER, 2003).

Um fator positivo desta técnica é que ela não gera nenhuma curva de aprendizado sistêmico e não utiliza nenhum recurso computacional. Portanto, qualquer cliente, possuindo domínio em informática ou não, está apto a participar da atividade. Esta prática ajuda os participantes a exercitarem a criatividade, e pode até mesmo servir como um momento de descontração (SNYDER, 2003).

Um ponto negativo desta abordagem é que ela necessita de uma parceria sólida entre equipe de desenvolvimento e cliente. Pelo fato de que no papel teoricamente tudo é possível, a experiência da equipe, especialmente dos especialistas em tecnologia da informação (TI) é exigida para direcionar o debate para requisitos que sejam aceitáveis de acordo com a tecnologia atual. A maturidade do cliente também é fundamental para que possa abstrair as representações e consiga imaginar em algum nível o produto real, transmitindo sua visão aos demais membros.

Em casos onde este cenário não for possível ferramentas como JustMind (JUSTMIND, 2014) ou protótipos incompletos segundo as práticas de Nielsen (1993) podem ser úteis.

### **3. A técnica protótipos de papel**

No cenário de desenvolvimento de projetos de *software*, um grande avanço é a aplicação de práticas de desenvolvimento ágil para reduzir os custos de desenvolvimento e entregar soluções em menor tempo. Qualquer investimento gasto no levantamento de requisitos, embora extremamente necessário no projeto como um todo, pode ser visto como um risco. Se o requisito for, por algum motivo, alterado o investimento em levantar este requisito é perdido. A técnica proposta por Snyder (2003) foi escolhida tendo em consideração um projeto com pouco investimento. Apresenta-se promissora por ser simples e econômica. A base desta técnica é a comunicação através de papel. A escrita e a representação por desenhos faz parte do cognitivo humano. O hábito de fazer anotações pessoais é comum entre as pessoas, em especial entre os projetistas. Desta forma, a mente humana já está condicionada a aprender e abstrair desenhos simples e anotações manuscritas. No mundo corporativo e em comunicações formais, os trabalhos impressos são uma preferência. Mesmo assim, é comum o papel ainda estar presente.

Segundo um estudo apresentado por Snyder (2003), ao entrarem em contato com uma tela ou mesmo com uma *Mock* de computador impressa em papel, é comum as pessoas darem à representação o valor de documentos, de forma a não se sentirem a vontade em rascunhá-la a mão ou alterar o seu conteúdo. Do ponto de vista de apresentação de conceitos de *softwares*, isso é ruim, uma vez que a equipe de desenvolvimento precisa constantemente do *feedback* do cliente.

Algumas ferramentas de *Mocks*, como o Balsamic (2014), procuram assemelhar-se a desenhos manuscritos justamente com a intenção, de quebrar este paradigma entre objeto apresentado e um documento em sua versão final. Porém a criação das *Mocks*, utilizando ferramentas informatizadas, ainda requer conhecimento na ferramenta selecionada. Neste aspecto, o uso de papel economiza recursos do projeto.

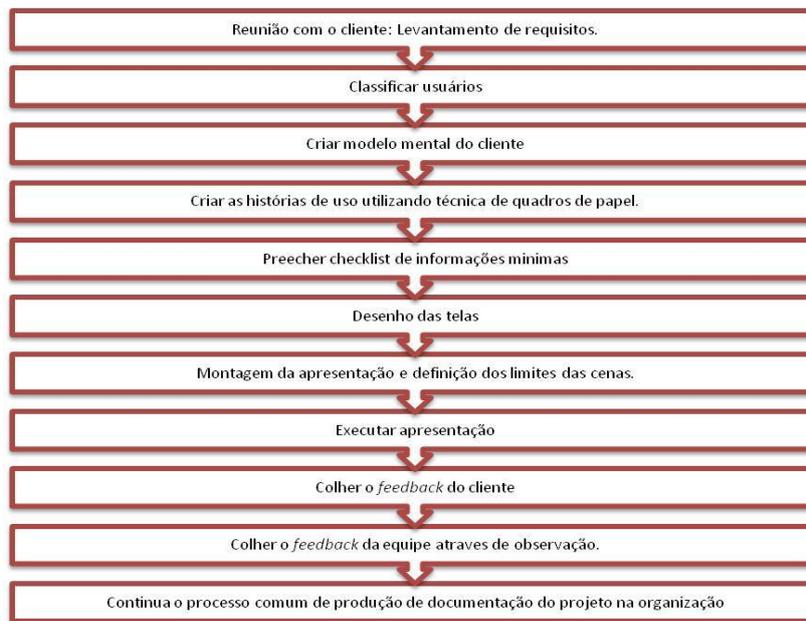
Outro fator relevante da escolha da técnica é que o cliente não necessita ter conhecimento técnico para manipular as ferramentas utilizadas no desenvolvimento do protótipo. Neste ponto, a utilização de papel se demonstra um ganho. Para representar o conceito de uma tela, não é necessária uma habilidade artística específica. Da mesma forma que na criação de histórias em quadros, com um pouco de tempo e material de escritório, as *Mocks* de tela podem ser produzidas sem maiores dificuldades pela equipe, apenas mantendo o foco na comunicação e abstraindo o conceito de perfeição. Conforme Nielsen (1993), protótipos são feitos para serem incompletos, logo para expressar uma ideia, uma folha com

alguns desenhos de tela são mais do que suficientes.

#### 4. Metodologia

Para o presente estudo foram realizadas atividades de levantamento e especificação de requisitos de IHC conforme apresentado na Figura 4.

**Figura 4 – resumo das atividades de levantamento de requisitos de IHC**



Fonte: elaborado pelos autores

A primeira atividade é a reunião com o cliente. Independentemente do ciclo de vida adotado para o projeto, os *designers* têm por objetivo obter um modelo metal do sistema, uma visão do negócio e da demanda do cliente. A atividade de classificação de usuários é destacada por diversos autores como Nielsen (1993), Shneiderman (1998), Shneiderman e Plasaint (2010) e Netto (2010), entre outros, como de extrema importância. É necessário conhecer o ambiente onde o sistema será aplicado, o perfil do usuário, seus conhecimentos e sua rotina de trabalho. Posteriormente os *designers* desenvolvem o modelo de interação do usuário (modelo mental).

Após as três atividades iniciais, uma boa técnica para validar os requisitos, antes mesmo da criação de documentos, é a aplicação da técnica de Quadros de História em Papel. Através do desenho da história durante a reunião, de forma ágil, os integrantes discutem os principais pontos do sistema, e as funcionalidades. É importante ressaltar que os quadros

deverão conter os pontos de início, pessoas e ambiente envolvidos na funcionalidade retratada.

Para a atividade de *Checklist* que valida a completude da especificação, foram selecionadas as seguintes informações:

- x Quais são os passos descritivos para se alcanças esta regra de negócio?
- x Que domínio está envolvido?
  - o O que o usuário ou sistema tem que fazer?
  - o Entradas e saídas do sistema
    - Levantamento de listas de domínios e valores pré-definidos.
    - Quais entradas serão controladas?
    - Quais restrições serão aplicadas as entradas?
    - Quais campos em formulários serão obrigatórios?
  - o Quais são os artefatos envolvidos?
  - o Quais artefatos estão envolvidos? Ex: *mouse*, teclado, tela sensível ao toque, sensor de movimentos, pranchas de desenho em alta resolução, outros.
  - o Suporte específico a hardware
    - Em uma tela sensível ao toque, os usuários têm dedos maiores do que o comum? Ex: Motoristas de caminhão, eletricitas. Será necessário utilizar apenas controles de tela grandes? Os usuários são idosos?
- x Freqüência de entrada dos dados
  - o O sistema terá entrada de dados, ocasionais ou constantes?
  - o O que é prioridade a agilidade de entrada dos dados ou clareza de informação?
  - o Alguma entrada de dados precisa ser automatizada?
- x Qual a resolução para qual o sistema é projetado?
- x Qual é o principal objetivo da funcionalidade?
  - o Caso de sucesso
  - o Caso de falhas
- x Como mensurar o sucesso da funcionalidade?
  - o Caso de testes que contemplem os valores esperados, limite e inesperados da funcionalidade.
  - o Especificação dos critérios de testes.
  - o Tempo limite para o usuário completar a tarefa.

Após estudar o usuário, entender suas demandas e documentar os requisitos de forma ágil, através da história em quadros, um protótipo em papel é desenvolvido para que o cliente valide a especificação que está sendo descrita.

Para ganhar mais agilidade, neste primeiro protótipo nenhuma linha de código será desenvolvida. Seguindo a proposta de Snyder (2003), a equipe de desenvolvedores elabora uma apresentação baseada na sequência de telas representadas em papel. Qualquer artifício que permita criar a sequência de telas, consumindo a menor quantidade de recursos, é válida. Como por exemplo, impressões de capturas de telas poderão ser recortadas para extrair componentes de tela, eliminando tempo de desenho, etc. Durante a preparação da apresentação do protótipo as regras de preparação de cena e definição de limites atendidos pelo protótipo descritas na técnica de Mágico de Oz podem ser observadas. Quando a apresentação estiver pronta algum usuário do sistema será convidado a avaliar o protótipo. Durante esta fase, alguns membros executores do projeto estarão presentes desempenhando alguns papéis específicos (papel de observador, papel de computador e papel de facilitador).

O observador analisa de forma silenciosa, todas as interações do usuário procurando falhas ou ponto de melhorias. O membro designado como computador tem o objetivo de guiar o usuário durante a apresentação, explicando as funcionalidades que estão sendo apresentadas, e descrevendo o que for relevante. E o facilitador que tem a responsabilidade de explicar os limites da cena ao usuário.

É também responsabilidade do facilitador incentivar a liberdade do usuário de forma que ele se sinta dono de seu produto. O facilitador deve convidar o usuário a realizar desenhos, rabiscos e anotações diretamente no protótipo. Como o projeto não está em um formato sistêmico, o usuário tem o conhecimento suficiente para alterá-lo. Este é um dos maiores objetivos da cena, recolher este *feedback*. A reunião inclui materiais comuns como folhas em branco e canetas para o usuário.

O *feedback* produzido é registrado pelos observadores. Também participa desta apresentação, o responsável pelo papel de computador, operando as fichas e recortes, simulando o que acontecerá no sistema real. No final desta atividade os requisitos iniciais utilizados para construir a apresentação deverão ser ajustados segundo os *feedbacks*.

O material, da apresentação, pode ser considerado como documentação. Os requisitos foram levantados, validados e aprovados pelo cliente.

O uso de apresentações envolvendo protótipos em papel é uma boa oportunidade para a aplicação da técnica proposta por Nielsen (1993), da criação de *Designs Paralelos*. A equipe pode ser dividida em grupos desafiados a montar mais de uma apresentação, oferecendo

diversas opções ao cliente, por exemplo, versões para diversas plataformas como Móvel, *Web* ou *Desktop* ou cada grupo pode ser responsável por atender as necessidades de uma classificação de usuários do sistema, como por exemplo, deficientes auditivos, deficientes visuais, usuários extremos e usuários comuns.

Desta forma, o escopo do projeto pode ser radicalmente alterado antes da escrita da primeira linha de código, adequando-se a expectativas e necessidades do cliente, já que no contexto atual de *software*, os sistemas possuem integrações complexas e uma constante necessidade de serem desenvolvidos em um prazo menor de tempo.

## **5. Resultados**

As técnicas e o processo levantados neste trabalho foram utilizados, em um projeto de inovação para transporte público desenvolvido pela *Interactive Mobile @dvertising* (Im@). Maiores detalhes sobre o projeto serão omitidos por questões de segurança e confidencialidade, porém os aspectos envolvendo a utilização das práticas serão demonstrados ao longo desta seção.

O projeto surgiu discutindo alternativas legais para a fiscalização do transporte público na cidade de Campinas no Estado de São Paulo. Surgiu a ideia de se aproveitar os sensores dos dispositivos móveis, como *Global Positioning System* (GPS), triangulação da rede e acelerômetros do celular dos passageiros na fiscalização. Através do sistema, o usuário seria capaz de saber a trajetória de uma determinada linha de ônibus, tempo de chegada até o ponto onde o usuário se encontra e outras funcionalidades.

Como o projeto é uma iniciativa de inovação da empresa, e não uma demanda solicitada por um cliente, a equipe não teve acesso a um usuário final para levantamento de requisitos. A equipe então discutiu sobre quem seriam os usuários do sistema e tentou classificá-los, de acordo com suas principais características de uso do sistema. Alguns perfis foram especificados: Estudantes; Pessoas que utilizam o transporte público a lazer; Pessoas que utilizam transporte público a trabalho; Idosos.

Um possível perfil de uso do sistema foi imaginado para cada uma destas classificações. Porém era necessário conhecer ao menos um usuário de cada perfil e entender como era a divisão de uso do sistema entre as classificações para entender qual perfil é o usuário principal da aplicação. Devido às limitações de recurso do projeto, não foi possível realizar entrevista ou estudos de observação destes usuários. Então foi adotada a técnica de aplicação de formulários. A escolha desta técnica foi embasada na grande quantidade de

*feedbacks* que pode ser obtido.

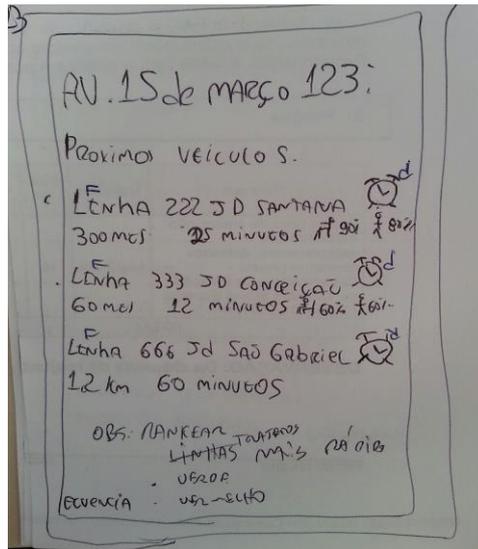
O formulário foi elaborado com perguntas que descreviam o usuário como, por exemplo: idade, sexo, se utiliza o transporte público, se possui acesso a pacote de dados, tipo de sistema operacional do aparelho móvel entre outras. O padrão para resposta adotado foi o tipo de pergunta com resposta fechada para facilitar na transformação das respostas em dados estatísticos.

Outras perguntas foram formuladas com o objetivo colher *feedbacks* e avaliar usabilidade, como funcionalidades que o sistema deveria ter. Este questionário teve o padrão de respostas aberta. Os formulários foram enviados via *e-mail* para todos os funcionários da empresa. Hoje a Im@ conta com cerca de 700 funcionários no total, no qual foi obtida um total de 100 respostas, sendo 4 de funcionários que não utilizam transporte público e, portanto, foram descartadas. A análise dos resultados demonstrou que a maioria dos usuários possuía um dispositivo móvel com sistema operacional Android. Portanto, foi decidido desenvolver de o sistema para esta plataforma. Foi possível perceber também que boa parte dos usuários, 60%, possuía um plano de dados ativo e tinham o costume de utilizar o aparelho durante o uso de transporte público. Desta forma ficou definido que as contribuições dos usuários poderiam ocorrer de duas formas, em tempo real, através de internet móvel ou ficariam retidas no aparelho até o momento que uma rede WIFI estivesse disponível.

Com a contribuição dos entrevistados nas perguntas de resposta aberta foi possível visualizar as funcionalidades que este sistema deveria conter. A equipe então descreveu as histórias de uso, envolvendo as funcionalidades levantadas na forma de quadro de histórias em papel. A Figura 5 demonstra a ilustração de uma destas histórias.

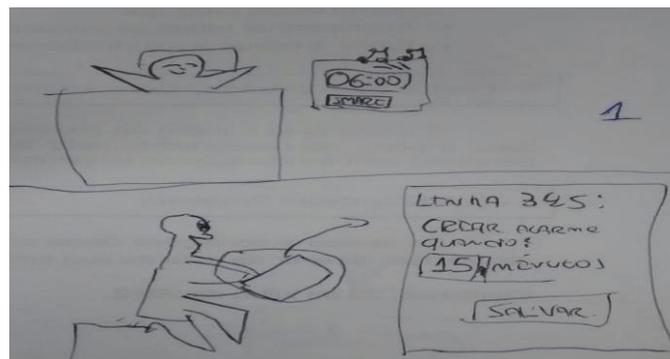
Logo após as ilustrações, uma apresentação utilizando a técnica de protótipos em papel foi montada e apresentada à equipe da Im@, uma vez que os *designers* não possuíam acesso a usuários finais. Esta cena precisou de apenas 3 horas para ser montada. Durante a simulação, algumas melhorias puderam ser feitas, como, por exemplo, surgiu a necessidade de o sistema criar uma nota (de 1 a 10) para todas as linhas de ônibus e exibir qual é a melhor com base nas informações históricas da linha de lotação, naquele horário, tempo médio de deslocamento entre os pontos, e dados como reclamações. Com base nestas informações, o usuário poderia escolher melhor seu trajeto utilizando linhas com as melhores notas. A Figura 6 apresenta a tela original rascunhada durante a apresentação das notas para as linhas de ônibus.

**Figura 5 – História do embarque diário utilizando transporte.**



Fonte: elaborado pelos autores

**Figura 6 – levantamento do requisito de nota das linhas**



Fonte: elaborado pelos autores

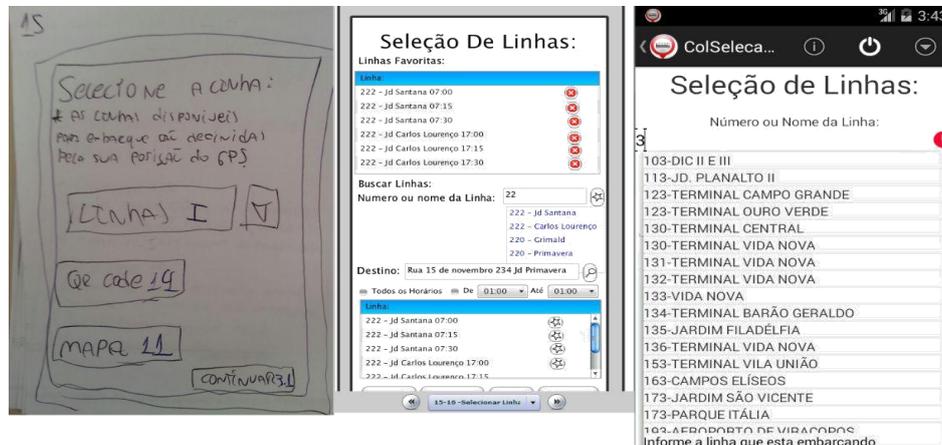
Uma nova apresentação, ainda em papel foi criada contemplando todas as alterações propostas na primeira demonstração. A partir dos resultados desta apresentação as histórias em quadros foram convertidas em histórias uso. A Figura 7 demonstra a história de alarme com a adição da nota atribuída à linha levantada nos testes com protótipos em papel.

Após a documentação do sistema uma nova apresentação foi desenvolvida, porém de forma computacional pois seria exibida a um possível cliente. Esta apresentação foi montada em uma ferramenta de geração de *Mocks* de baixa fidelidade, e levou 16 horas para ser desenvolvida. Este resultado é interessante, uma vez que foi gerado praticamente a mesma apresentação, com inclusão de algumas novas funcionalidades e pequenas melhorias, porém em uma plataforma diferente. Desta forma, foi verificado que, utilizando a plataforma computacional, foi necessário mais do que o dobro de tempo para criar a apresentação. Isto

corroborou a praticidade e economia da técnica de papel.

A Figura 7 demonstra dos protótipos até chegar em sua fase de codificação.

**Figura 7 – Tela de embarque em diferentes fases do projeto.**



Fonte: elaborado pelos autores

Desta forma, os requisitos de negócio e usabilidade foram especificados e o projeto pode seguir para a fase de codificação. Da conversa inicial e entendimento de uma oportunidade até o momento dos requisitos documentados e validados, foram gastas 3 semanas de trabalho, um tempo relativamente curto para a elicitação de requisitos, com poucos recursos econômicos investidos nesta fase.

## 6. Considerações finais

Esta pesquisa pode constatar que a literatura possui as técnicas necessárias para levantamento e modelagem de requisitos para IHC e que, nem sempre, estas são combinadas de forma a potencializar os resultados no que diz respeito à agilidade e qualidade.

Este trabalho propôs uma forma de combinar algumas técnicas de validação dos requisitos funcionais e de IHC antes da fase de codificação. Através de um estudo de caso, demonstrou-se que a validação dos requisitos pode ser feita de maneira simples, econômica, ágil e eficazmente sem o uso de ferramentas computacionais, alcançando, assim, o objetivo inicial deste trabalho. Alias, foi comprovado que o uso de uma ferramenta de geração de protótipos consumiu mais do que o dobro de tempo para criar a apresentação para o usuário que a técnica de papel.

Com as técnicas utilizadas neste trabalho, a fase de apresentação dos requisitos, que

pode ser um ponto sensível em qualquer projeto de *software* é contornada de maneira satisfatória. Isto devido a que o usuário pode sim reconhecer novas demandas causando alterações de escopo sem isso ser dispendioso para o projeto. Esta situação não impacta de forma significativa o projeto, uma vez que pouco recurso foi despendido no levantamento e documentação dos requisitos.

## **Referências**

BALSAMIC. **Balsamic Mockups**. Disponível em: <<http://balsamiq.com/products/mockups/>>. Acesso em: 01 mai 2014.

BARBOSA, S.; SILVA, B. **Interação humano-computador**. Rio de Janeiro: Campos, 2010.  
**JUSTMIND**. Disponível em: <<http://www.justinmind.com>>. Acesso em: 01 mai 2014.

CAETANO, M. L. S. Clareza, atualização, acesso às informações e estética em sites de Organizações Não Governamentais. **Research, Society and Development**, v. 2, n. 1, p. 80-92, set. 2016.

GAMA, M. X. B. A Liderança na Era da Informação e do Conhecimento nas empresas. **Research, Society and Development**, v. 3, n. 1, p. 02-18, nov. 2016

KELLEY, J. F. **A natural language program developed with the OZ Paradigm: implications for supercomputing systems**. First International Conference on Supercomputing Systems, New York: ACM, pp. 238-248. 1985.

LAUREL, B. **The art of human-computer interface design**. Boston: Addison-Wesley-Logman.1990.

**MANIFESTO**. Disponível em: <<http://manifestoagil.com.br/principios.html>>. Acesso em: 01 mai 2014.

NIELSEN, J. **Usability engineering**. New York: Academic Press, 1993.

NETTO, A. A. de O. **IHC e a engenharia pedagógica**. Florianópolis: Visual Books Ltda,

2010.

PRESSMAN, R. **Software engineering: a practitioner's approach**. 6º ed.: Mc Graw Hill, 2005.

SANTOS, C. A. et al. Um modelo de sistema de informação gerencial: vantagem competitiva no processo da logística reversa do óleo de cozinha. **Research, Society and Development**, v. 4, n. 1, p. 62-88, jan. 2017

SHNEIDERMAN, B. **Designing the user interface: strategies for effective human computer interaction**. MA: Addison-Wesley-Logman. 1998.

SHNEIDERMAN, B.; PLAISANT, C. **Designing the user interface: strategies for effective human-computer interaction**. MA: Addison Wesley Publishing Company Incorporated, 2010.

SOMMERVILLE, I. **Engenharia de software**. 9 ed. São Paulo: Pearson Prentice Hall. 2011.

SNYDER, C. **Paper prototyping: the fast and easy way to design and refine user interfaces**. San Francisco: Morgan Kaufmann. 2003.

USEIT. Disponível em: <<http://www.useit.com>>. Acesso em: 20 mai 2012.