Uma Abordagem para Apoiar Avaliações de Usabilidade em Sistemas Web com base em Mineração de Dados

Matheus Souza[1], **Rafael Ribeiro**[1], **Pedro Almir Oliveira**[2], **Pedro Santos Neto**[1]

[1]Universidade Federal do Piauí
Teresina - Piauí - Brasil

[2]Instituto Federal do Maranhão
Pedreiras - Maranhão - Brasil

{matheusmmcs,raffael404}@gmail.com, pedro.oliveira@ifma.edu.br

pasn@ufpi.edu.br

Abstract. *The usability is a relevant characteristic regarding the quality of a system. However, the usability evaluation of systems is usually very costly and complex. Although there are many proposed methods with the intention of mitigating such disadvantages of traditional evaluations, they usually require the allocation of time and participants exclusively for their achievement. This work presents an approach that aims to eliminate these disadvantages, allowing the execution of remote evaluations on systems through the analysis of data generated by their daily use. The approach uses data mining and clustering techniques to identify possible problem areas. An evaluation of the approach showed that it was capable of finding important usability problems and that directly impact the quality of use of the system.*

Resumo. *A usabilidade é uma característica muito relevante em relação à qualidade de um sistema. Entretanto, avaliar a usabilidade de sistemas geralmente é muito oneroso e complexo. Apesar de existirem diversos métodos propostos com o intuito de mitigar tais desvantagens das avaliações tradicionais, geralmente eles necessitam da alocação de tempo e de participantes exclusivos para a sua realização. Este trabalho apresenta uma abordagem que visa eliminar essas desvantagens, possibilitando a realização de avaliações remotas em sistemas por meio da análise dos dados gerados pelo seu uso cotidiano. A abordagem faz uso de técnicas de agrupamento e mineração de dados para identificar possíveis pontos problemáticos. Uma avaliação da abordagem mostrou que ela foi capaz de encontrar problemas de usabilidade importantes e que impactam diretamente na qualidade de uso de um sistema.*

1. Introdução

Devido as facilidades que proporcionam, os sistemas *Web* difundiram-se rapidamente e hoje fazem parte do dia-a-dia das pessoas. Para as empresas que utilizam esses sistemas como meio de distribuir seus produtos e serviços, é de fundamental importância que eles sejam desenvolvidos com foco na aceitação dos clientes. A aceitação de serviços e dispositivos tecnológicos depende de vários fatores tais como o *design*, os recursos financeiros

disponíveis, o contexto dos utilizadores, as próprias funções disponibilizadas e o seu mapeamento com as capacidades e competências dos utilizadores finais, ou seja, o seu grau de usabilidade [Martins et al. 2013].

Um alto grau de usabilidade influencia diretamente na satisfação do usuário e, consequentemente, na sua fidelidade a um *Website* [Flavián et al. 2006]. Nesse cenário, vários métodos de avaliação foram desenvolvidos para melhorar a experiência do usuário através da localização e correção de problemas de usabilidade em sistemas *Web* [Fernandez et al. 2011]. O teste com usuários reais é um dos métodos mais utilizados para avaliar usabilidade, já que ele provê informações diretas sobre como as pessoas usam o sistema e quais são seus problemas com a interface [Nielsen 1994a].

Por conta de alguns fatores como custo elevado, número limitado de participantes [Tullis et al. 2002], complexidade de organização e alta demanda de tempo do teste laboratorial tradicional, tem havido bastantes trabalhos relacionados ao teste de usabilidade com o objetivo de melhorar o seu desempenho [Ahmad et al. 2010]. Apesar de conseguirem dirimir algumas desvantagens, boa parte desses trabalhos ainda é baseado na reprodução de testes em ambiente laboratorial, o que demanda a alocação de tempo dos participantes exclusivamente para a realização do teste. Esse não é um cenário desejável, por exemplo, em empresas onde os funcionários teriam de interromper a execução de suas funções para dedicar-se a participar do teste.

Este trabalho apresenta uma abordagem para permitir a realização de testes de usabilidade sem a necessidade de alocação de tempo de teste por parte dos participantes, uma vez que utiliza os dados gerados pela própria operação do sistema em ambiente de produção. A abordagem utiliza técnicas de mineração de dados para analisar os *logs* (dados de utilização) do sistema e apresentar de forma simplificada (por meio de gráficos e tabelas) fluxos de utilização que contêm possíveis problemas de usabilidade. Além de facilitar o trabalho do especialista em usabilidade, capturando e sumarizando os dados, o uso da abordagem permite que os testes sejam realizados com maior frequência, consequentemente acelerando as melhorias na usabilidade do sistema.

O restante deste trabalho está estruturado como segue. Na Seção 2 são apresentados alguns conceitos que servem como base para o trabalho; na Seção 3 o método e a ferramenta que compõem a abordagem proposta são descritos em detalhes; na Seção 4 o estudo experimental realizado e os resultados obtidos com a abordagem são descritos; na Seção 5 são apresentados alguns trabalhos relacionados; e, por fim, na Seção 6 são discutidas as conclusões e perspectivas para trabalhos futuros.

2. Referencial Teórico

Na área de Interface Humano-Computador (IHC), o conceito de usabilidade mais amplamente aceito é o da norma ISO 9241-11: "a capacidade de um produto ser usado por um conjunto de usuários para alcançar objetivos determinados, com eficácia, eficiência e satisfação, em um contexto específico de uso" [ISO 1998, Fernandez et al. 2011].

Dentre os tipos de métodos de avaliação de usabilidade, eles podem ser classificados em: teste, inspeção, investigação, modelagem analítica e simulação [Ivory and Hearst 2001]. No teste de usabilidade tradicional, geralmente os participantes têm de realizar uma série de tarefas pré-determinadas, enquanto um ou mais observadores gravam suas ações, o tempo e quaisquer comentários relevantes sobre a utilização.

Toda essa informação é usada para desenvolver uma lista de problemas de usabilidade ou potenciais pontos problemáticos na aplicação [Tullis et al. 2002].

A eficácia dos testes como uma forma de descobrir problemas de usabilidade em *Websites* e outras aplicações é amplamente aceita [Tullis et al. 2002]. O problema é que geralmente esses testes são caros, demorados e trabalhosos [Mueller et al. 2009], pois exigem a contratação de especialistas em usabilidade, compra de equipamentos, organização do ambiente laboratorial, etc., mas são fundamentais para um software ser aprovado pelos utilizadores. Além disso, os usuários tendem a agir de forma diferente do habitual por estarem sendo observados por avaliadores e por estarem em um ambiente não familiar, dificultando a obtenção de dados reais nos testes [Castillo et al. 1998].

Os testes com usuários podem ser executados remotamente, de maneira que os usuários não necessitam se deslocar para um laboratório de testes. A principal diferença entre testes remotos e presenciais é a separação espacial entre os especialistas e os usuários. Dentre os testes remotos, eles podem ser síncronos ou assíncronos. Nesse caso, a separação temporal entre os especialistas e os usuários é a principal diferença, permitindo que os usuários executem os testes a qualquer momento, independente do acompanhamento de especialistas no momento da realização do teste [Bruun et al. 2009].

Para a análise dos *logs*, utilizam-se técnicas de *Web Usage Mining* (WUM), que consiste na aplicação de técnicas de mineração de dados para descobrir padrões de uso a partir de dados *Web*, com o objetivo de entender e melhor servir as necessidades de aplicações *Web* [Srivastava et al. 2000].

3. Abordagem Proposta

Com o intuito de reduzir os custos e a complexidade envolvidos em avaliações de usabilidade de sistemas *Web*, este trabalho propõe uma abordagem composta por um método e uma ferramenta para automatizar parte desse tipo de avaliação. O método da abordagem propõe as diretrizes para automatizar avaliações com base na captura e análise de *logs* de utilização. A ferramenta, denominada UseSkill, é responsável por implementar os conceitos presentes no método, possibilitando a sua utilização em ambientes reais, além de permitir avaliar os resultados da abordagem.

3.1. Método

O método proposto neste trabalho baseia-se na captura remota de ações realizadas por usuários em determinado sistema *Web* e na comparação entre as utilizações “boas” e “ruins”. A ideia por trás do método é identificar quais partes das funcionalidades influenciam negativamente na utilização dos usuários e fazem eles divergirem entre “bons” e “ruins”, apontando possíveis pontos problemáticos.

Para que seja possível capturar os dados corretos, analisá-los e apoiar na identificação de problemas de usabilidade, o método proposto foi dividido em quatro etapas: captura dos *logs* de utilização, preparação dos dados, análise dos dados e geração de relatórios. A Figura 1 apresenta as etapas do método proposto.

Figura 1. Abordagem proposta para avaliar a usabilidade de sistemas *Web*.

3.1.1. Captura de ações

Diferentemente de testes laboratoriais, que avaliam um usuário por vez, esse método propõe que as avaliações sejam remotas e assíncronas, o que dispensa a presença de especialistas e usuários no local e no instante da avaliação. Essas características permitem que diversos usuários sejam avaliados simultaneamente, sem acrescer custos com translado e preparação de ambiente, além de simplificar a logística durante a avaliação [Bruun et al. 2009].

Para permitir que as avaliações sejam remotas, assíncronas e com etapas automatizadas, os *logs* devem possuir informações importantes sobre as ações realizadas durante as utilizações do sistema. A captura dessas ações ocorre por meio de um componente de captura de *logs* desenvolvido em *Javascript* e que executa nos *Web browsers*. Ele é personalizável, permitindo capturar informações específicas que podem ser úteis durante a análise dos *logs*. Entretanto, dentre as informações capturadas, algumas são obrigatórias por serem cruciais para diferenciá-las e permitirem realizar comparações entre as utilizações do sistema. São elas:

- **Tempo**: horário que a ação ocorreu;
- **Tipo da ação realizada**: clique, preenchimento de campo, *mouseover*, etc.;
- **Elemento que sofreu a ação**: botão, *link*, campo de texto, etc.;
- **Onde a ação ocorreu**: qual página (geralmente utiliza-se a URL, mas também podem ser utilizados metadados);
- **Quem fez a ação**: um identificador de qual usuário realizou a ação.

3.1.2. Preparação dos dados

A preparação dos dados proposta no método baseia-se em definições do pré-processamento de dados em *Web Usage Mining* [Mobasher 2006]. A primeira fase da preparação é a identificação de quais ações capturadas fazem parte da funcionalidade a ser analisada. Essa delimitação dos dados é importante para que cada funcionalidade possa ser analisada isoladamente.

Em seguida ocorre a limpeza dos dados, removendo tipos de ações indesejadas. Por exemplo, caso ao avaliar uma funcionalidade não haja a necessidade de acompanhar

quando há a ação *mouseover* em algum elemento da interface, esse tipo de ação pode ser desconsiderado para diminuir os dados ruidosos.

Com os dados delimitados e limpos, ocorre a identificação das sessões de uso. Cada sessão representa um conjunto de ações realizadas por determinado usuário ao utilizar uma funcionalidade de um sistema. Seu conceito é semelhante à de uma utilização da funcionalidade. Um usuário, por exemplo, pode utilizar diversas vezes a mesma funcionalidade do sistema, gerando assim diversas sessões de uso.

Resumidamente, a preparação dos dados corresponde à etapa em que as ações capturadas são transformadas nas sessões de uso a serem avaliadas.

3.1.3. Análise dos dados

Com as sessões (conjuntos de ações) capturadas, o próximo passo é calcular métricas para medir a qualidade de uso de cada sessão. O método proposto sugere o cálculo de duas métricas: eficácia e eficiência. Elas foram selecionadas por serem relacionadas diretamente ao conceito de usabilidade proposto na ISO 9241-11 e por serem amplamente utilizadas na literatura [ISO 1998].

O cálculo da eficácia baseia-se na quantidade de ações obrigatórias (AO) realizadas pelo usuário ao utilizar determinada funcionalidade. Por exemplo, caso uma funcionalidade possua 5 ações a serem realizadas obrigatoriamente, mas o usuário realizou apenas 3, a eficácia seria de 60%. A eficácia é o percentual de ações obrigatórias realizadas na sessão. A Equação 1 apresenta como a eficácia é calculada.

$$Efica_s = \frac{AO_s * 100}{AO} \tag{1}$$

A variável AO_s representa a quantidade de ações obrigatórias contidas na sessão e a variável AO representa o total de ações obrigatórias da funcionalidade. O valor da eficácia da sessão ($Efica_s$) varia entre 0 e 100. Para a funcionalidade, o cálculo da eficácia é a média das eficácias das sessões, como pode ser visto na Equação 2, onde s representa as sessões dos usuários que utilizaram a funcionalidade.

$$Efica_f = \frac{\sum_{s=1} Efica_s}{s} \tag{2}$$

A eficiência é a proporção entre a eficácia e o esforço demandado, nesse caso medido em tempo e quantidade de ações realizadas. Caso um usuário atinja todos os objetivos, mas demore muito, sua sessão de uso terá eficácia alta e eficiência baixa. Para calcular a eficiência de uma sessão de uso é necessário calcular a eficácia e dividir sobre a quantidade de ações e de tempo despendido na sessão, como pode ser visto na Equação 3.

$$Efici_s = \frac{Efica_s}{(\frac{A_s}{mA_ok}) * (\frac{T_s}{mT_ok})} \tag{3}$$

A variável A_s equivale à quantidade de ações da sessão, mA_ok é quantidade de ações da sessão que foi realizada corretamente e com menor número de ações, T_s é o

tempo despendido durante a sessão, mT_ok é tempo da sessão correta que foi realizada mais rapidamente. Para o cálculo da eficiência de uma funcionalidade é utilizada a média das eficiências das sessões ($Efici_s$), segundo a Equação 4.

$$Efici_f = \frac{\sum_{s=1} Efici_s}{s} \quad (4)$$

Em seguida, de acordo com as métricas calculadas para cada uma das sessões, elas são classificadas como "boas" ou "ruins". Caso a sessão tenha bons índices de eficácia e eficiência, ela será classificada como uma "boa" sessão. Caso contrário, se os índices forem baixos, a sessão será classificada como "ruim".

Com as sessões classificadas, a ideia é agrupá-las em: Grupo de Sessões Referência (GSR), contendo as utilizações das funcionalidades do sistema de maneira esperada; ou Grupo das Demais Sessões (GDS), que não lograram êxito, realizaram ações demasiadamente ou demoraram muito para finalizar a sessão. Após o agrupamento das sessões em GSR e GDS é possível comparar tais grupos a fim de encontrar diferenças entre eles. Essas comparações servem como base para a classificação de cada uma das ações contidas nas sessões.

As ações que foram mais frequentes no GSR são classificadas como "obrigatórias" (AO), ou seja, passos que os usuários devem realizar para utilizar a funcionalidade corretamente. As demais ações contidas no GSR e que não foram classificadas como AO, são as ações "corretas" (AC). As ações mais frequentes no GDS e que não estão entre as mais frequentes no GSR são as "problemáticas" (AP). Por fim, as ações contidas no GDS, que não fazem parte do GSR e que não foram classificadas como AP, são consideradas "alertas" (AA).

A classificação das ações é uma etapa importante para gerar relatórios que apontem para as partes de funcionalidades com maior possibilidade de possuírem problemas de usabilidade. As ações AP e AA servem para dar indícios de onde estão os locais problemáticos e as métricas apontam quais sessões enfrentaram mais dificuldades.

3.1.4. Geração de Relatórios

A partir das métricas e classificações realizadas, são gerados relatórios que apoiam a análise e interpretação dos dados por parte de especialistas em usabilidade. A proposta baseia-se na possibilidade de ter uma visão geral da usabilidade e ao mesmo tempo permitir análises aprofundadas em determinadas utilizações.

O método sugere a utilização de listas contendo as ações realizadas sequencialmente e a possibilidade de selecionar ações para visualizar as informações capturadas, como tipo de ação, elemento, local, horário e usuário. Com isso é possível, por exemplo, verificar quanto tempo o usuário demorou entre uma ação e outra, ou identificar quais elementos receberam mais ações repetidamente.

As análises rápidas que permitem aos especialistas terem noções gerais da usabilidade são baseadas em grafos. Os nós dos grafos representam as ações realizadas, sendo que os nós redondos correspondem às ações mais frequentes e os quadrados são as ações

que não estão entre as mais frequentes. As arestas do grafo apontam quais caminhos foram percorridos pelos usuários, as cores dos nós indicam a classificação de cada ação, a largura das arestas apontam quais caminhos foram mais percorridos e o tamanho de cada nó (diâmetro ou tamanho do lado) representa quais ações foram mais realizadas.

3.2. Ferramenta

A ferramenta UseSkill *OnTheFly* baseia-se no método proposto para auxiliar avaliações de usabilidade. Ela permite a realização de avaliações de usabilidade em contextos controlados e em ambientes de produção. Inicialmente foi proposta a ferramenta UseSkill *Control* (USC) [Souza et al. 2015], que visa auxiliar a realização de testes de usabilidade remotos apenas em contextos controlados, necessitando da definição de roteiros, tarefas e questionários. Esse tipo de avaliação, em um contexto controlado, é classificada como formal, requerendo a execução de tarefas específicas, previamente selecionadas por um especialista [Ivory and Hearst 2001].

Para implementar o novo método proposto, foi desenvolvida a ferramenta UseSkill *OnTheFly* (USOTF), que contempla avaliações de usuários em seu ambiente de produção, executando suas atividades do dia-a-dia. Essa necessidade surgiu pois convidar usuários, criar roteiros e preparar o sistema para ser testado envolve custos e complexidade logística. Dessa forma, a USOTF não necessita da definição de roteiros, nem convidar usuários a realizarem um conjunto de tarefas visando apenas avaliar suas interações. O componente de captura de *logs* deve ser inserido no código fonte do sistema a ser avaliado, podendo assim ser restrito a partes do sistema ou a todas as funcionalidades dele. A avaliação ocorre com base nos *logs* capturados enquanto os usuários utilizam o sistema em ambiente de produção, sem a definição de tarefas específicas para a avaliação.

3.2.1. Funcionamento da Ferramenta

Para avaliar a usabilidade de sistemas com a USOTF é necessário realizar as seguintes etapas: cadastrar testes e funcionalidades; definir ações iniciais e finais; executar algoritmos de mineração e agrupamento; e por fim avaliar os relatórios gerados.

A ferramenta USOTF permite a criação de testes, que são compostos por um conjunto de funcionalidades e de janelas temporais, que possibilitam avaliar versões específicas do sistema. O cadastro de testes necessita apenas do título e de um código que identificará os *logs* de cada teste. Cada funcionalidade possui um título, quais tipos de ações a serem desconsiderados e um limiar de tempo máximo que o usuário pode demorar entre uma ação e outra.

Após o cadastro das funcionalidades é necessário identificar quais são suas ações iniciais e finais. Essa etapa é importante para identificar onde a funcionalidade inicia e termina. Por fim é necessário cadastrar janelas temporais, que possuem uma data inicial e final. As avaliações das funcionalidades ocorrem dentro de janelas temporais. Com os dados de utilização capturados, a ferramenta também identifica quais funcionalidades foram as mais utilizadas do sistema.

A análise dos dados capturados é baseada na mineração e agrupamento de dados. A primeira etapa é a mineração de padrões sequenciais frequentes (*Frequent Sequential*

Patterns ou FSP) no universo de todas as sessões identificadas. A ideia dessa etapa é encontrar subsequências de ações frequentes em determinado conjunto de sessões. Essas ações servem de indícios de pontos onde os usuários devem passar obrigatoriamente para realizar a funcionalidade corretamente (ações obrigatórias) ou de pontos problemáticos onde muitos usuários estão enfrentando dificuldades.

Devido ao grande volume de dados, é inviável identificar os padrões sequenciais frequentes por meio da geração de todas as combinações possíveis e em seguida identificar a melhor. Há um grupo de algoritmos de mineração de dados voltados especificamente para tal necessidade. Dentre os diversos algoritmos de mineração de padrões sequenciais frequentes, foi utilizado o algoritmo CM-SPADE. Ele é baseado no algoritmo Apriori e mescla estratégias dos algoritmos ClaSP, SPADE e SPAM. O CM-SPADE foi selecionado por ser mais rápido que os algoritmos originais e consumir menos memória ao minerar padrões sequenciais frequentes [Fournier-Viger et al. 2014].

Ao executar o CM-SPADE com todas as sessões identificadas de determinada funcionalidade em uma janela temporal, a ferramenta apresenta um grafo contendo as ações mais realizadas sequencialmente na funcionalidade. Com o grafo em mãos, o avaliador deve classificar as ações identificadas, geralmente sendo classificadas como ações obrigatórias (AO) ou ações problemáticas (AP).

Em seguida, com as métricas de eficácia e eficiência de cada sessão calculadas, a ferramenta agrupa as sessões em busca de um grupo de sessões referência. Para realizar esse agrupamento foi utilizado o algoritmo k-means [MacQueen et al. 1967], que agrupa as sessões de acordo com suas similaridades e dissimilaridades. A ideia é identificar diversos grupos de sessões e selecionar o grupo com centróide mais próximo do ponto máximo de eficácia e eficiência como GSR. As demais sessões são classificadas como pertencentes ao grupo GDS.

Após a definição dos grupos, a ferramenta utiliza novamente o algoritmo para mineração de FSPs. Entretanto, agora a mineração ocorre isoladamente para cada um dos grupos. A ideia é identificar as ações mais realizadas apenas em sessões de referência e nas demais sessões. O CM-SPADE é executado duas vezes, com valores pré-definidos empiricamente, mas que o avaliador pode ajustar de acordo com sua necessidade. Os valores iniciais são: suporte mínimo de 75% e quantidade mínima de 4 ações nos FSPs. O resultado dessa etapa é a classificação das ações.

Por fim, são gerados os relatórios baseados em grafos e listagens de ações realizadas. A Figura 2 exemplifica um grafo gerado pela UseSkill, que apresenta uma sessão de uso. Nele há nós e arestas maiores em relação aos demais, pois foram mais realizados. As cores representam a classificação das ações, sendo as azuis “obrigatórias”, as verdes “corretas” e as amarelas “alertas”. É perceptível também que há ciclos no grafo, apresentando conjuntos de ações que foram repetidas. A existência de ciclos, o tamanho dos nós e a classificação das ações são fundamentais para apoiar uma visualização geral da qualidade de uso da funcionalidade.

3.3. Desafios e Limitações

Dentre os desafios e limitações da UseSkill *OnTheFly*, o apoio fornecido para avaliar a usabilidade de sistemas *Web* leva em consideração apenas eficácia e eficiência, não conseguindo avaliar a satisfação dos usuários. Outro ponto importante inerente à captura

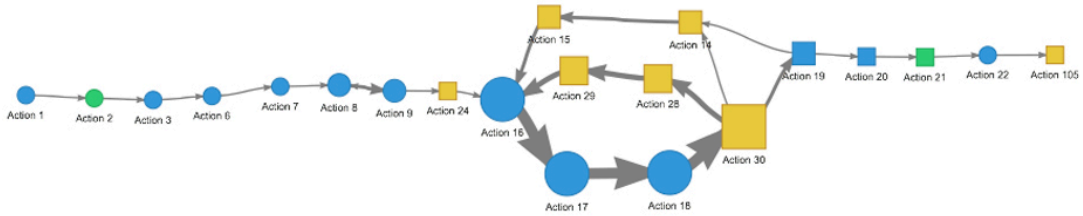

Figura 2. Grafo exemplificando uma sessão de uso. Os nós azuis são AO, os amarelos AA e os verdes AC. Esse grafo não possui AP.

de *logs* em ambientes reais é que, apesar dos usuários estarem no contexto de uso do dia-a-dia, nem sempre o usuário está focado ao realizar as funcionalidades, impactando na quantidade de ações realizadas e no tempo para finalizá-las.

Para gerar relatórios completos é necessário que haja uma quantidade razoável de *log* capturado. Caso o sistema seja novo e possua poucos usuários, avaliar a usabilidade de suas funcionalidades baseado na captura de *logs* em ambiente real pode demorar tanto quanto outros métodos de avaliação de usabilidade. A necessidade de alterar o sistema a ser testado também é uma limitação da ferramenta proposta, pois nem sempre é simples e de fácil acesso ao código fonte do sistema a ser testado, embora a alteração a ser realizada seja bastante simples e limita-se à inclusão do componente de captura no sistema.

Outro desafio para o método proposto é a definição de ações iniciais, finais e obrigatórias. Essa definição impacta diretamente no cálculo da eficácia e eficiência das sessões, ou seja, influenciando diretamente nos relatórios gerados. A identificação dessas ações nem sempre é simples, podendo ser complexa e com custo benefício baixo durante a primeira avaliação realizada com a ferramenta.

4. Estudo Experimental

Com o objetivo de comparar os resultados da UseSkill em relação a outro método de avaliação de usabilidade relevante foi realizado um Estudo Experimental. O método selecionado para comparar com a UseSkill foi o *Cognitive Walkthrough* (CW) ou Passo-a-passo Cognitivo [Polson et al. 1992]. O método CW foi selecionado por ser capaz de avaliar funcionalidades específicas em um software e por possuir baixo custo [Fernandez et al. 2012].

Para alcançar o objetivo da pesquisa, as seguintes questões guiaram as avaliações realizadas:

- **QP1**: a UseSkill identifica problemas de usabilidade, segundo especialistas na área?
- **QP2**: a quantidade de problemas de usabilidade distintos identificados com apoio da UseSkill é diferente se comparado com avaliações baseadas em CW?

- **QP3**: a relevância dos problemas de usabilidade identificados com apoio da UseSkill é diferente dos problemas encontrados em avaliações baseadas em CW?

4.1. Desenho experimental

A definição dos participantes foi realizada de acordo com os dois tipos de avaliações distintas. O Grupo X avaliou o sistema utilizando um método baseado em CW. Ele foi composto por dois *designers* de interface que trabalham na empresa responsável pelo sistema sob avaliação. O Grupo Y realizou avaliações com apoio da UseSkill *OnTheFly*, sendo composto de apenas um avaliador que possui conhecimento sobre a ferramenta, sobre o sistema a ser avaliado e sobre usabilidade. Devido à pequena quantidade de profissionais com conhecimento sobre usabilidade, a seleção dos participantes foi baseada na conveniência, o que classifica esta pesquisa empírica como um *Quasi-experiment* [Wohlin et al. 2012].

O sistema *Web* selecionado é utilizado para gerir planos de saúde. Ele foi avaliado tanto com apoio da UseSkill, como por meio do CW. Para avaliar com a UseSkill *OnTheFly* foi necessária a inserção do componente de captura de *logs* no código fonte do sistema. Foram capturados dados de utilização do sistema em ambiente de produção durante 4 meses, totalizando aproximadamente 3 milhões de ações capturadas. Nesse estudo experimental foi avaliada apenas a versão do sistema correspondente a duas semanas de uso, tanto na UseSkill quanto por meio do CW.

Após a definição da janela temporal de 2 semanas, foram escolhidas quais funcionalidades seriam avaliadas. A seleção objetivou escolher as partes do sistema mais utilizadas pelos usuários no dia-a-dia. A UseSkill auxiliou essa seleção apresentando uma lista de funcionalidades de acordo com o número de ações realizadas durante o período selecionado. Foram selecionadas para o estudo experimental as 6 funcionalidades mais utilizadas.

A relevância dos problema foi subdividida em 4 métricas. A primeira, sobre a severidade do problema, considerou a escala proposta por Nielsen [Nielsen 1994b] e que foi sumarizada na Tabela 1.

Tabela 1. Escala de severidade de problemas de usabilidade.

Severidade	Tipo	Descrição
0	Sem importância	Não afeta a operação da interface
1	Cosmético	Não há necessidade imediata de solução
2	Simples	Problema de baixa prioridade (pode ser reparado)
3	Grave	Problema de alta prioridade (deve ser reparado)
4	Catastrófico	Muito grave, deve ser reparado de qualquer forma

Após a definição do grau de severidade, cada problema de usabilidade também foi classificado quanto a sua frequência, impacto e persistência. As notas foram valores inteiros entre 0 e 4, de forma análoga à avaliação de severidade, onde quanto maior a nota, mais relevante é o atributo. Para evitar problemas de interpretação, a definição desses valores seguiram os seguintes conceitos [Nielsen 1994b]:

- **Frequência**: se é comum ou raro, se acontecem em muitas funcionalidades ou muitas etapas de uma funcionalidade. Considera em quantos locais esse problema ocorre;

- **Impacto**: se é fácil ou difícil de ser superado pelos usuários;
- **Persistência**: se os usuários podem superar apenas uma vez, quando eles sabem sobre o problema, ou se os usuários são incomodados repetidamente pelo problema.

4.2. Resultados

4.2.1. Identificação de Problemas com apoio da UseSkill (QP1)

Ao final da avaliação realizada com apoio da UseSkill, o avaliador do Grupo Y identificou 10 problemas de usabilidade. Para validar tais problemas, eles foram analisados pelo Grupo X logo após a avaliação das mesmas funcionalidades com o método *Cognitive Walkthrough*. Para cada um dos 10 problemas, os avaliadores informaram se concordam ou não, além de atribuir notas sobre a relevância dos problemas.

O avaliador B concordou com todos os problemas identificados com apoio da UseSkill e o avaliador A discordou de apenas um deles. De acordo com o avaliador A, um dos problemas era apenas uma proposta de melhoria para aumentar a flexibilidade de uso.

O índice Kappa foi utilizado para medir a concordância entre os avaliadores. Os valores podem variar entre -1 e 1, onde 1 representa uma concordância perfeita, 0 é o que seria esperado por acaso, e valores negativos indicam potencial desacordo sistemático [Viera et al. 2005]. O índice Kappa entre os avaliadores foi 0,94, representando uma concordância próxima à ideal.

Considerando os resultados obtidos, a resposta para a QP1 é: sim, segundo profissionais com experiência em usabilidade, a ferramenta apoia a identificação de problemas de usabilidade em sistemas *Web*.

4.2.2. Quantidade (QP2)

O avaliador A identificou 11 problemas e o avaliador B identificou 10, ambos do Grupo X, que basearam-se no método CW. Dentre os 21 problemas encontrados, 5 eram iguais entre os dois avaliadores, restando assim 16 problemas distintos identificados por meio de CW.

O avaliador C, do Grupo Y, identificou 10 problemas com apoio da UseSkill. Desses problemas, 3 eram iguais a problemas identificados pelos avaliadores do Grupo X, totalizando assim 7 novos problemas não encontrados por meio de CW. A quantidade de problemas durante o estudo experimental totalizou 23 problemas distintos identificados com os dois métodos.

Apesar da quantidade de problemas identificados por cada avaliador ser parecida, de acordo com a quantidade de problemas identificados com apoio de cada método, a resposta da QP2 é: sim, a quantidade de problemas distintos identificados com apoio da UseSkill foi menor que a quantidade de problemas identificados com CW.

4.2.3. Relevância (QP3)

A pontuação de cada problema de usabilidade no que se refere à sua severidade, frequência, impacto e persistência foi definida em uma escala de 0 a 4. A Tabela 2 sumariza os resultados obtidos com as notas dadas pelos avaliadores.

Tabela 2. Média das notas atribuídas para severidade, frequência, impacto e persistência. As notas foram separadas por cada avaliador e por método que apoiou a identificação. As "em comum" são as notas dos problemas que foram encontrados pelos avaliadores A e B.

Avaliador	Prob.	Sever.	Frequênc.	Impacto	Persist.
Avaliador A (*Cognitive Walkthrough*)	11	2,27	2,64	2,55	3,00
Avaliador B (*Cognitive Walkthrough*)	10	1,40	3,50	1,80	1,60
Média dos Avaliadores (*Cognitive Walkthrough*)	-	1,83	3,07	2,17	2,30
Avaliador A (CW em comum com Aval. B)	5	2,20	3,80	2,40	3,00
Avaliador B (CW em comum com Aval. A)	5	1,20	3,80	1,80	1,40
Média dos Avaliadores (CW em comum)	-	1,70	3,80	2,10	2,20
Avaliador A (UseSkill)	9	3,56	3,44	3,56	3,44
Avaliador B (UseSkill)	10	1,90	3,10	2,00	1,80
Média dos Avaliadores (*Cognitive Walkthrough*)	-	2,73	2,77	3,27	2,78

Ao comparar as notas atribuídas pelos dois avaliadores nos 5 problemas identificados por ambos, percebe-se que o avaliador A atribuiu notas maiores que o avaliador B. Entretanto, mesmo com essa diferença nas notas, ambos consideraram os problemas identificados com a UseSkill mais severos, impactantes e persistentes, mas que ocorrem com menor frequência em relação aos identificados por eles mesmo durante avaliação do sistema com CW.

Dessa forma, a resposta da QP3 é: sim, a severidade, o impacto e a persistência dos problemas identificados com a UseSkill aparentam serem maiores, enquanto a frequência é menor que os problemas identificados com CW.

4.3. Limitações e Ameaças à validade

A primeira limitação é que os tratamentos aplicados (UseSkill e CW) pertencem a "tipos" e "classes" de métodos de avaliação de usabilidade distintos. O primeiro pertence à classe "teste", enquanto o segundo faz parte da classe "avaliação heurística" [Ivory and Hearst 2001]. Substituir o método de "avaliação heurística" por outra ferramenta do mesmo "tipo de método" (teste de usabilidade remoto) implica em:

- Selecionar outra ferramenta: identificar artigos e ferramentas do mesmo "tipo de método" e que estivessem disponíveis para utilização. A seleção seria realizada por autores da UseSkill, o que poderia enviesar a escolha;

- Selecionar especialistas: após a seleção da ferramenta, seria difícil encontrar especialistas nela. Pesquisadores ligados à UseSkill não poderiam se aprofundar na ferramenta, pois seria outra ameaça à validade.

Além das limitações sobre os métodos utilizados, a quantidade de avaliadores experientes em usabilidade disponíveis para o experimento é pequena, sendo outra ameaça à validade. Por conta da quantidade de participantes fica inviável realizar análises estatísticas nos resultados, impactando na generalização dos resultados obtidos.

5. Trabalhos Relacionados

O principal trabalho relacionado é a ferramenta UseSkill *Control* [Souza et al. 2015], que permite a realização de testes semi-automatizados remotos e assíncronos. A captura dos *logs* ocorre por meio de um *plugin* para *browsers*, não exigindo modificações no sistema testado, entretanto necessita da definição prévia de quais usuários são "iniciantes" e "experientes". Apesar de facilitar a realização de testes, a ferramenta *Control* ainda necessita da definição de tarefas e da alocação de tempo exclusivo dos participantes. A USOTF necessita apenas do mapeamento da funcionalidade e os dados do sistema em produção servem para avaliar a usabilidade. Os trabalhos relacionados presentes no trabalho da UseSkill *Control* também serviram de base para a concepção da ferramenta UseSkill *OnTheFly*, especialmente as ferramentas WELFIT [de Santana and Baranauskas 2015] e USABILICS [de Vasconcelos and Baldochi Jr 2012].

Geng e Tian [Geng and Tian 2015] propõem um método para identificar problemas de usabilidade relacionados à navegação baseado na comparação de padrões de uso reais e preditos. Os padrões de uso reais são extraídos utilizando-se algoritmos de *Web Usage Mining* em *logs* de servidores *Web*. Os padrões de uso preditos são obtidos através da simulação do comportamento ideal do usuário utilizando-se modelos cognitivos. As diferenças encontradas entre esses dois padrões de uso são usadas para descobrir problemas e sugerir ações corretivas para melhorar a usabilidade. Porém, a utilização de *logs* do servidor, apesar de mais simples, não fornece informações detalhadas, limitando-se a identificar os caminhos de navegação do usuário. Além disso, o desenvolvimento de modelos cognitivos é uma tarefa complexa, que exige o auxílio de especialistas no assunto.

6. Conclusões e Trabalhos Futuros

Este trabalho apresentou uma abordagem para apoiar avaliações de usabilidade de sistemas *Web* remotamente baseada em mineração de dados. A abordagem proposta é composta por um método e uma ferramenta, denominada UseSkill *OnTheFly*. O método baseia-se na comparação das melhores utilizações em relação às demais, comparando semi-automaticamente as ações realizadas e indicando possíveis problemas de usabilidade. A ferramenta que implementa esse método captura *logs* de interação dos usuários diariamente na aplicação em ambiente de produção.

Como resultado do estudo experimental realizado, percebe-se que a ferramenta foi responsável por apoiar a identificação de 7 problemas que não foram encontrados por meio de CW. A abordagem proposta não substitui avaliações já existentes, mas complementa com problemas que impactam diretamente na utilização do sistema. As notas atribuídas à severidade demonstram que os problemas identificados com apoio da UseSkill são tão relevantes quanto os por meio de CW.

Apesar das limitações e ameaças à validade deste estudo experimental, percebe-se que a abordagem proposta tem potencial para apoiar a avaliação de usabilidade em sistemas *Web*. O método proposto possibilita a avaliação de como usuários se comportam na aplicação sem a necessidade de toda a complexidade e custos de testes de usabilidade laboratoriais. Além disso, a ferramenta permite visualizar quais funcionalidades são mais utilizadas, apoiando na priorização de avaliações de usabilidade.

Ressalta-se que outros estudos ainda não publicados já foram desenvolvidos envolvendo a ferramenta UseSkill. Porém, esses estudos têm um foco diferente do apresentado neste trabalho. O trabalho submetido à Sessão de Ferramentas do CBSoft 2016 foca em apresentar os componentes de software da ferramenta, como arquitetura, interface, funcionalidades, etc. Este trabalho foca em apresentar o método que serve de base para a ferramenta e como ele é utilizado. Como trabalhos futuros, pretende-se estudar o uso de técnicas de aprendizado de máquina para ajudar na detecção de padrões de uso incorreto e os problemas de usabilidade associados a a eles, facilitando ainda mais a identificação de problemas.

Referências

Ahmad, W. F. W., Sulaiman, S., and Johari, F. S. (2010). Usability management system (usemate): A web-based automated system for managing usability testing systematically. In *User Science and Engineering (i-USEr), 2010 International Conference on*, pages 110–115. IEEE.

Bruun, A., Gull, P., Hofmeister, L., and Stage, J. (2009). Let your users do the testing: A comparison of three remote asynchronous usability testing methods. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1619–1628, New York, NY, USA. ACM.

Castillo, J. C., Hartson, H. R., and Hix, D. (1998). Remote usability evaluation: can users report their own critical incidents? In *CHI 98 Cconference Summary on Human Factors in Computing Systems*, pages 253–254. ACM.

de Santana, V. F. and Baranauskas, M. C. C. (2015). Welfit: A remote evaluation tool for identifying web usage patterns through client-side logging. *International Journal of Human-Computer Studies*, 76:40–49.

de Vasconcelos, L. G. and Baldochi Jr, L. A. (2012). Towards an automatic evaluation of web applications. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 709–716. ACM.

Fernandez, A., Abrahão, S., and Insfran, E. (2012). A systematic review on the effectiveness of web usability evaluation methods. In *Evaluation & Assessment in Software Engineering (EASE 2012), 16th International Conference on*, pages 52–56. IET.

Fernandez, A., Insfran, E., and Abrahão, S. (2011). Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, 53(8):789 – 817. Advances in functional size measurement and effort estimation - Extended best papers.

Flavián, C., Guinalíu, M., and Gurrea, R. (2006). The role played by perceived usability, satisfaction and consumer trust on website loyalty. *Information & Management*, 43(1):1–14.

Fournier-Viger, P., Gomariz, A., Campos, M., and Thomas, R. (2014). Fast vertical mining of sequential patterns using co-occurrence information. In *Advances in Knowledge Discovery and Data Mining*, pages 40–52. Springer.

Geng, R. and Tian, J. (2015). Improving web navigation usability by comparing actual and anticipated usage. *IEEE Transactions on Human-Machine Systems*, 45(1):84–94.

ISO (1998). Ergonomic requirements for office work with visual display terminals (VDTs) - part 11: Guidance on usability, ISO 9241-11. *International Organization for Standardization*.

Ivory, M. Y. and Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33(4):470–516.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

Martins, A. I., Queirós, A., Rocha, N. P., and Santos, B. S. (2013). Avaliação de usabilidade: uma revisão sistemática da literatura. *RISTI-Revista Ibérica de Sistemas e Tecnologias de Informação*, (11):31–43.

Mobasher, B. (2006). Web usage mining. *Web data mining: Exploring hyperlinks, contents and usage data*, 12.

Mueller, C., Tamir, D., Komogortsev, O., and Feldman, L. (2009). An economical approach to usability testing. In *Computer Software and Applications Conference, 2009. COMPSAC '09. 33rd Annual IEEE International*, volume 1, pages 124–129.

Nielsen, J. (1994a). *Usability engineering*. Elsevier.

Nielsen, J. (1994b). Usability inspection methods. In *Conference companion on Human factors in computing systems*, pages 413–414. ACM.

Polson, P. G., Lewis, C., Rieman, J., and Wharton, C. (1992). Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of man-machine studies*, 36(5):741–773.

Souza, M. M. C., Oliveira, P. A., Ribeiro, R. F., Britto, R. S., and Neto, P. S. (2015). Useskill: uma ferramenta de apoio à avaliação de usabilidade de sistemas web. *XIV Simpósio Brasileiro de Qualidade de Software (SBQS)*.

Srivastava, J., Cooley, R., Deshpande, M., and Tan, P.-N. (2000). Web usage mining: Discovery and applications of usage patterns from web data. *Acm Sigkdd Explorations Newsletter*, 1(2):12–23.

Tullis, T., Fleischman, S., McNulty, M., Cianchette, C., and Bergel, M. (2002). An empirical comparison of lab and remote usability testing of web sites. In *Usability Professionals Association Conference*.

Viera, A. J., Garrett, J. M., et al. (2005). Understanding interobserver agreement: the kappa statistic. *Fam Med*, 37(5):360–363.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.