

## AVALIAÇÃO DO PROCESSO DE TESTES DE SOFTWARE DE UMA EMPRESA CANADENSE

### VALUATION OF THE SOFTWARE TESTING OF A CANADIAN COMPANY

Rodrigo Augusto Peres Velozo<sup>1</sup>

Gustavo Kimura Montanha<sup>2</sup>

#### RESUMO

O acesso rápido e a alta disponibilidade de informações são cada vez mais importantes à sociedade, onde empresas almejam alta eficiência de seus processos através da tecnologia, enquanto usuários comuns buscam maior comodidade ao seu cotidiano. Devido tal necessidade, a tecnologia móvel tornou-se atrativa, fazendo-a uma das principais mídias computacionais da atualidade. Como essa tecnologia tornou-se parte do cotidiano das pessoas, o desenvolvimento para estas plataformas precisam estar adequadamente alinhado para proporcionar uma experiência agradável e suprir as necessidades dos usuários, de modo que os testes de software devem ser adequadamente alinhados ao consumidor para um controle de qualidade eficaz. Este trabalho teve como objetivo realizar uma análise sobre o processo de testes de softwares de uma empresa canadense, identificando e propondo melhorias. O estudo avaliou a organização por um período de quinze semanas, de modo a acompanhar a implementação do processo de testes na organização. Concluiu-se que a nova estrutura de testes trará benefícios à organização, mas ainda há espaço para melhoria, contudo, observou-se que novas melhorias podem ainda ser implementadas.

**Palavras-chave:** Computação em Nuvem. Computação Móvel. *Robotium Framework*. Testes de Software.

#### ABSTRACT

The easy access and high availability of information is increasingly important for society, where companies seek for high efficiency of its processes through technology while users are looking for greater convenience to their daily lives. Because of this need, the mobile technology has become attractive, making it one of the main computational media nowadays. As this technology has become part of the daily lives of people, development for these platforms need to be properly aligned to provide a pleasant experience and meet the user's needs, in a way that software testing must be properly aligned to the consumer for an effective quality control. This study aimed to carry out an analysis of the software testing process of a Canadian company, identifying and proposing improvements. The study evaluated the organization for a period of fifteen weeks in order to monitor the implementation of the testing process. It was concluded that the new testing structure will benefit the organization, however, it is noted that further improvements may also be implemented

**Keywords:** cloud computing, mobile computing, *robotium framework*, software testing.

<sup>1</sup> Bolsista do Ciência sem Fronteiras/CAPES – Brasil e Graduando em Informática para Negócios na Faculdade de Tecnologia de Botucatu (FATEC). E-mail: rodrigo\_velozo@outlook.com

<sup>2</sup> Doutor em Agronomia e Professor da Faculdade de Tecnologia de Botucatu (FATEC)

## 1 INTRODUÇÃO

O acesso rápido à informação é cada dia mais importante, não apenas para as organizações, mas incluindo também os usuários finais, em que a tecnologia pode auxiliar na resolução de problemas de seu cotidiano, além de trazer comodidade e facilidade para a realização de atividades específicas. Com a possibilidade de centralizar todas as suas informações e gerenciá-las facilmente através de seus dispositivos, a tecnologia móvel possibilitou um mundo onde as pessoas estão continuamente conectadas e a troca e a consulta de informações são constantes, permitindo a tomada de decisão com base em melhores resultados e menor tempo de resposta nas requisições do usuário.

O conceito de “ter informação em suas mãos em qualquer lugar e a qualquer momento” impulsionou o crescimento da tecnologia móvel. Buscando esta visão, inovações quanto à conectividade sem fio, eficiência energética, redução do espaço e peso de *hardware*, interfaces com o dispositivo e softwares permitiram a criação da atual tecnologia, tornando realidade essa visão de anos atrás (SATYANARAYANAN, 2011).

Nunca houve tanto acesso à informação como atualmente. A internet já fora uma revolução quanto ao compartilhamento de informações, todavia, transpor esta tecnologia de maneira a se tornar algo portátil gerou uma nova necessidade para a sociedade moderna, tornando dispositivos móveis parte essencial desse novo cenário.

A tecnologia móvel traz novas formas de interação e comunicação, diferentes maneiras de organização e condução da vida cotidiana, novos meios de administrar um negócio e acessar serviços. Deste modo, tecnologias em dispositivos móveis são mutuamente de interesse econômico e político, pelo fato de serem o centro das transformações em comunicação e mídia digital, em que estes aparelhos já se tornaram híbridos com as demais tecnologias, como dispositivos de câmeras, localização ou assistentes pessoais (GOGGIN, 2012).

Deste modo, o desenvolvimento de sistemas que sejam compatíveis para dispositivos portáteis torna-se um desafio fundamental para as empresas, de maneira que elas passem a disponibilizar suas informações ou mesmo os seus produtos neste ambiente.

Contudo, as plataformas para dispositivos móveis se diferem dos sistemas computacionais convencionais, não apenas na interface fornecida ao usuário, como também nos recursos dos dispositivos. Assim, surge a necessidade da adequação para desenvolvimento em sistemas móveis, de maneira a garantir uma boa interação do usuário e

funcionamento adequado do software sobre os diferentes cenários criados com a falta de controle sobre as condições de uso da tecnologia.

Deve-se considerar também a competitividade do mercado que apresenta múltiplos sistemas para plataformas móveis, onde cada um disponibiliza características próprias quanto ao *design* de softwares, tendo alterações também na interação do usuário e fluxo de navegação. Tal divergência, apesar de ser sadia ao mercado e dar mais opções aos consumidores, cria mais especificações nas quais os desenvolvedores devem estar atentos no momento da implementação de seus sistemas.

Devido a tantos fatores envolvendo a tecnologia móvel, a realização de testes de software para tal é complexa de forma a contemplar o maior número de cenários possíveis para averiguar corretamente o funcionamento do sistema.

Deste modo, o presente trabalho teve como objetivo analisar a estrutura de testes de software de uma empresa canadense de desenvolvimento de softwares móveis, de modo a averiguar sua eficácia e identificar melhorias neste processo.

## **2 MATERIAL E MÉTODOS**

Para o levantamento da metodologia de testes adotada pela organização, além de exemplificação dos processos pelos gestores, foi efetuado um monitoramento das atividades do setor,. Todavia, durante o período de análise, a organização teve a alteração deste processo, onde foram estudadas as ferramentas envolvidas e escolhas adotadas juntamente à empresa.

A partir da nova estrutura da organização, através de levantamento bibliográfico e estudo das tecnologias no mercado, foi realizada uma avaliação, buscando-se a eficiência dos métodos e ferramentas adotados, além de possíveis melhorias.

A empresa alvo do estudo foi uma *startup* canadense que opera desenvolvendo soluções para veículos elétricos, trabalhando em prototipação, design, coleta e análise de dados em tecnologias relacionadas, além de contar com um departamento de desenvolvimento de aplicações móveis e *web*, a fim de desenvolverem soluções sob encomenda para outros negócios.

Esta mesma organização mantém um sistema próprio de localização de estações de carregamento para veículos eletrônicos que visa fornecer os dados mais precisos e compreensíveis para a América do Norte. Tal software reúne informações sobre estações de

múltiplas empresas, facilitando o descobrimento destes recursos para usuários de veículos elétricos e opera sob dispositivos *Apple*, *Android* e como serviço *web*

### 3 RESULTADOS E DISCUSSÃO

A estrutura inicial de testes da organização se tratava da própria equipe de desenvolvedores realizando testes em seus dispositivos móveis, de modo a identificar possíveis falhas no sistema, sendo estes, então, encaminhados a um grupo de *beta testers* que a organização mantém, para o qual eram encaminhadas também instruções do que deveria ser averiguado na nova versão do software.

Todavia, mesmo com a existência de terceiros realizando uma avaliação sobre o software, o ponto de vista principal que a organização mantinha era sobre a busca de erros no programa, de modo que raramente a empresa recebia resultados referentes à usabilidade do software. No entanto, a empresa adotava da ferramenta *Google Analytics* de modo a inserir pontos de verificação para a contagem do uso de modo a criar estatísticas sobre o sistema, o que lhe permitia recolher informações sobre como os usuários finais operavam o sistema. Entretanto, mesmo possuindo tais dados, não era realizada uma análise detalhada para compreensão de como os usuários utilizavam o programa.

Conforme Myers, Sandler e Badgett (2012) explicam, além da necessidade de averiguar o funcionamento correto do sistema, as empresas devem testar seus sistemas considerando o fator humano, de forma a confrontar o real uso do sistema pelo usuário contra o cenário ao qual fora projetado para ser executado. Ainda assim, a validação sob este fator é altamente subjetiva, sendo necessária a interpretação dos dados coletados para poder encontrar a real causa de um problema encontrado.

Sendo assim, como a parte central dos testes se realizava com base na própria equipe de desenvolvimento, a qual já possuía conhecimento de como o sistema funciona e o que lhe era esperado realizar, presumia-se que o usuário possuiria igualmente o conhecimento necessário para a utilização do software, o que não só se mostrou incorreto futuramente, mas também era uma aproximação inadequada, tendo em vista que o sistema era voltado para usuários randômicos e que não receberiam treinamento prévio.

Certa vez, durante a averiguação dos resultados de uso gerado pelo *Google Analytics*, um gestor identificou que o uso das ferramentas contidas dentro da página de detalhes de uma estação de carregamento era muito inferior ao esperado. A partir desta descoberta, foram

introduzidos outros indicadores ao sistema de modo a traçar como o usuário utilizava o sistema, os quais revelaram que muitos dos usuários sequer entravam na página de detalhes.

Como esta página é o que proporciona grande número de informações e ferramentas ao usuário, sendo classificada pela própria organização como uma peça chave de seu negócio, fora despertada a urgência de se entender como o usuário atua pela empresa.

A partir deste ocorrido, não só foram repensados os indicadores utilizados pelo *Google Analytics*, como também foi adotada uma nova ferramenta para a coleta de informações a respeito da usabilidade: a plataforma *User Testing*.

Por meio desta plataforma, a empresa consegue vídeos de usuários finais utilizando seu sistema, bem como fornecendo um *feedback* com base em perguntas pré-estipuladas de modo a entender como as pessoas utilizam e o que pensam sobre o software.

Com o uso desta plataforma e de ferramentas similares, uma organização consegue a obtenção de dados aliviando possíveis alterações pelo efeito da pressão do usuário estar em um ambiente não familiar, conforme explicado na obra de Myers, Sandler e Badgett (2012), já que, o usuário realiza as gravações em seu ambiente de escolha, apenas encaminhando-as à plataforma e sendo pago por fazê-lo. Alinhando esta ferramenta junto às informações remotas que podem ser obtidas via *Google Analytics*, pode-se traçar melhor o perfil de usabilidade do sistema, já que tais ferramentas passam a suprir a falta de detalhamento ao utilizar um método remoto anônimo, tal qual o sistema da *Google*.

A partir do momento em que a empresa alinhou ambas as tecnologias, foi identificada uma grande falha na interface do sistema, em que o usuário não compreendia como proceder para se chegar a tal página de detalhes.

A adoção de um sistema para testes de usabilidade de software também se mostrou eficaz na ação de correção da organização para o mesmo problema. Como resposta, fora criado um tutorial que, após ser avaliado, demonstrou ineficácia na correção do problema. Como ação final, foi decidido alterar a interface do software.

A problemática que a empresa possuía por não ter conhecimento da ação de seus usuários impactava seu negócio, de modo que diversas ferramentas foram implementadas desde a criação do sistema, mas algumas não possuíam uso adequado vide a necessidade de interação com os detalhes de uma estação. Com uma nova estrutura de testes, a organização conseguiu identificar um problema importante e averiguar rapidamente o resultado das medidas corretivas adotadas.

Tendo então se atentado à importância dos testes de software, a organização visou buscar a otimização em seu processo, adotando testes automáticos como forma de reduzir os recursos empregados na realização dos testes que o objetivo de encontrar falhas técnicas, direcionando melhor seus recursos aos problemas lógicos que o sistema pode apresentar, de modo a aumentar a qualidade de seu software.

Conforme Castro (2013) explica, testes manuais não apenas requerem a alocação de mais recursos frente aos testes automáticos, mas também implicam na possibilidade do erro humano durante o procedimento, uma vez que o agente que realiza o teste pode vir a esquecer de validar um aspecto. Ainda, como forma a diminuir o uso de recursos, empresas passam a avaliar apenas as modificações mais recentes, desconsiderando possíveis efeitos que tais mudanças podem gerar sob outras áreas do sistema. Deste modo, com a automação dos testes, mesmo que parcial, passa a gerar ganhos à organização.

Inicialmente, optou-se por efetuar a automação apenas a plataforma Android, como forma de averiguação do potencial da tecnologia. Para tal, foi decidido o uso do *framework Robotium*, ferramenta de código aberto que permite simular interações com um aplicativo junto de verificações da condição atual do software e que, além de utilizar da mesma linguagem de programação que a plataforma Android, não necessita do código fonte do aplicativo para a execução de um teste (ZADGAONKAR, 2013).

Visando melhorar o processo de criação dos testes, foi estudada a utilização de geradores automáticos de testes para o framework, os quais capturariam as interações realizadas com o dispositivo e fornecendo um *script* para reprodução as mesmas ações. No entanto, os sistemas experimentados (*Robotium Recorder* e *Testroid Recorder*) não satisfaziam as necessidades da organização.

Ambas tecnologias, geravam *scripts* fiéis para a reprodução da ação primeiramente executada, desconsiderando variações no tempo necessário para o carregamento das funções do aplicativo, de modo que, quando um teste era migrado para um dispositivo com menor potência ou mesmo por irregularidades na velocidade da conexão da internet, este desconsiderava o tempo de espera para a criação dos elementos da interface para interação, resultando em um falso-negativo.

Os softwares, durante a gravação das ações, mediam o tempo entre cada ação e aplicavam-no ao script gerado como um código de pausa, em que o aplicativo aguarda pelo tempo estipulado e, quando atingido, executa a próxima linha de comando. A aproximação

adequada para avaliação de sistemas a ser empregada deve operar de modo a aguardar até que determinado evento ocorra, antes de executar a próxima linha de comando.

Esta segunda aproximação traz o benefício de que se estipula um tempo máximo de espera do software para a conclusão do evento verificado de maneira que, caso ultrapassado, o teste acusa o erro sobre o carregamento do evento, ao invés de determinar que o próximo comando apresente falha, como no método criado pelos softwares avaliados, o que traz localização apropriada dos erros do sistema avaliado. Ainda, utilizando desse método, o teste prossegue assim que o evento é concluído, evitando tempo ocioso dos testes e permitindo acompanhamento apropriado da performance de cada dispositivo para o aplicativo.

Sendo assim, os scripts criados necessitavam ser reajustados para atender apropriadamente o software da organização, já que a mesma visa à utilização de seu sistema sob os mais diversos cenários.

Os gravadores também se limitavam a apenas reproduzir um fluxo de ações e garantir que as atividades seriam executadas, onde era necessário adicionar posteriormente validações desejadas sobre os dados gerados por cada ação e averiguação das informações apresentadas ao usuário.

Devido a tais fatores, foi rejeitada a utilização destes softwares, já que a alta carga de manutenção não compensaria o investimento da organização, passando assim a adotar a criação manual dos cenários de testes para seus aplicativos.

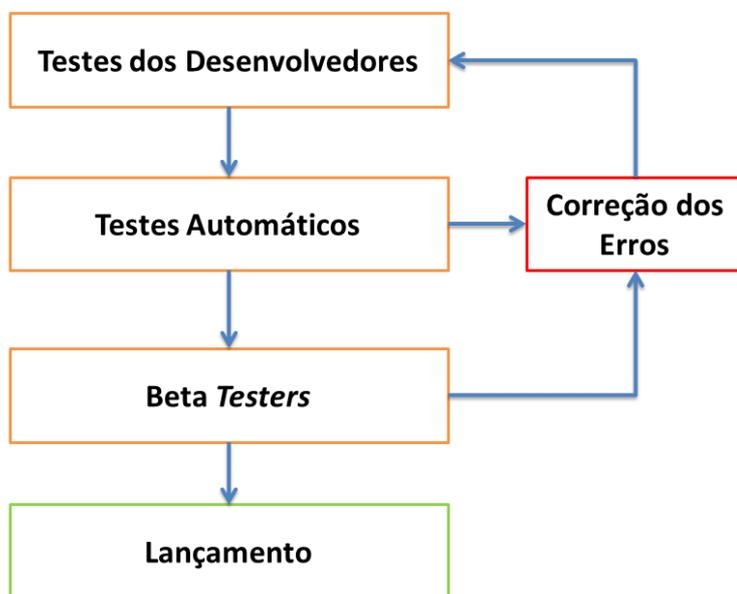
Sendo assim, o novo modelo de testes da organização (Figura 1) se desmembrou em três etapas até o lançamento:

- Primeiro, os desenvolvedores avaliam as alterações no sistema, de maneira a não apenas encontrar possíveis erros, mas também identificando otimizações. Tais testes ocorrem durante o próprio desenvolvimento das alterações no sistema.
- A segunda etapa envolve a utilização e testes automáticos para averiguar não apenas as novas modificações no sistema, mas envolvendo o software como um todo, de maneira a garantir que tais mudanças não impactaram outras áreas do aplicativo. Esta segunda etapa tem objetivo de garantir que o sistema esteja funcionando adequadamente, sem erros de código, para a próxima etapa de validação.
- Por fim, a última etapa consiste na avaliação do sistema pelos *beta testers*, utilizando plataformas de testes de usuários e de um grupo de usuários do

aplicativo dispostos a realizar a avaliação do sistema. Esta etapa busca principalmente erros lógicos e críticas para melhoria da aplicação.

- Finalmente, após a conclusão de todas as etapas, a nova versão do software é disponibilizada e começa-se uma nova coleta de dados de usabilidade remotamente via *Google Analytics*

Figura 1: Novo ciclo de testes da organização



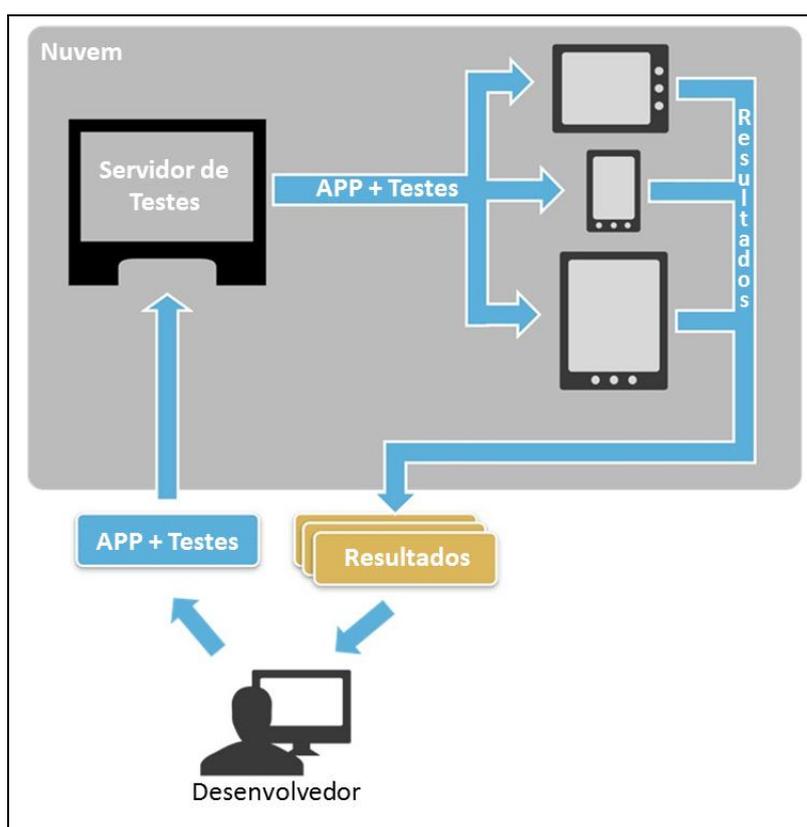
Contudo, apesar do aumento da eficiência nos testes com o modelo implementado, o mesmo ainda pode ser otimizado de maneira a não só reduzir os recursos empregados na automação dos mesmos, mas também aumentando significativamente o número de cenários em que o software será avaliado.

Segundo Pitoura e Samaras (2012), o ambiente da computação móvel é restritivo de diversas formas. Os elementos que compõem a tecnologia móvel em si são limitados e/ou instáveis. Como exemplo, a conexão do sistema costuma ser de baixa capacidade na maior parte do tempo, além da conectividade ser perdida frequentemente por períodos de tempo irregulares, dados a cada cenário que o dispositivo é exposto. Tais restrições trouxeram grande impacto na estruturação e design para a tecnologia, de modo que os sistemas precisam providenciar um serviço eficiente indiferente das variáveis que englobam o ambiente de uso e atributos do dispositivo.

Desta forma, um sistema para dispositivos móveis necessita ter testes realizados em diversos aparelhos e condições diferentes, como forma de assegurar sua qualidade ao consumidor, independente do modelo de seu equipamento e local de uso, já que tais condições são incontroláveis pelo desenvolvedor (CHANDRA *et al.*, 2014).

Uma tecnologia que permite que a empresa possa ampliar os dispositivos e cenários disponíveis é a contratação de um sistema de testes automáticos em nuvem. Tal sistema consiste no conceito de computação em nuvem aplicada a testes de software, no qual são providos equipamentos e ferramentas para a realização de testes através de uma conexão via internet. Com esta modalidade, é possível reduzir os custos do ciclo de testes, além de amplificar a variedade de equipamentos no qual a validação será executada, tornando-se um serviço interessante para desenvolvedores em tecnologias móveis (GAO; BAI; TSAI; 2011).

Figura 2: Processo de testes em um serviço de testes em nuvem



Fonte: adaptado de DEVELOPER XAMARIN, 2015.

Como exemplo do benefício que esta alteração pode gerar, durante o desenvolvimento de uma ferramenta, foi identificado que os aparelhos da marca *Samsung* rotacionavam inadequadamente imagens. Contudo, a assunção de que todos os dispositivos da marca estariam sujeitos a esta irregularidade foi baseada na utilização apenas de dispositivos

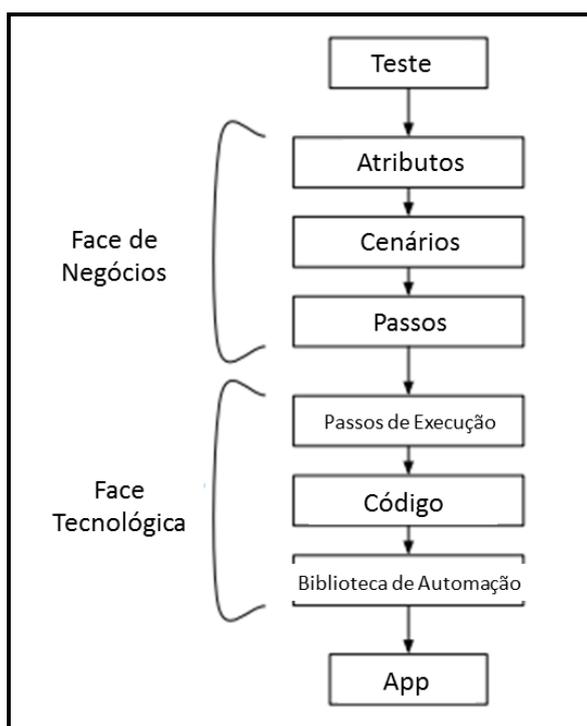
disponíveis pela organização, sendo um número bem limitado referente ao total disponível no mercado.

Desta forma, com a contratação de um serviço de testes em nuvem, a empresa poderia validar se o ocorrido estaria realmente ligado a uma característica dos modelos desta marca, bem como identificar outros dispositivos que possam apresentar a mesma falha antes que um usuário identifique o problema e tenha que reportá-lo a empresa.

Alguns exemplos de serviços de testes automáticos em nuvem são *Testroid Cloud* e *Xamarin Cloud*.

Além da contratação de tal serviço, a empresa ainda pode melhorar seu processo de testes migrando do framework *Robotium* para uma linguagem multiplataforma. Como exemplo, há o *framework Cabalash* que, segundo explicado na obra de Wynne e Hellesoy (2012), consiste em um *framework* que opera em *Cucumber*, uma linguagem de desenvolvimento orientada ao comportamento, em que se descreve o comportamento do sistema do ponto de vista do usuário, alinhada a outras linguagens para a criação das interações com o software a ser testado. Tal visão cria uma divisão do teste em duas faces, em que a primeira está voltada a entender como o usuário agiria no sistema e a segunda seria a parte técnica que cria o *script* para realizar a ação no aparelho (Figura 3).

Figura 3: Passos de um teste em *Cucumber*



Fonte: Adaptado de WYNNE; HELLESOY, 2012.

Com esta alteração, a empresa consegue utilizar um mesmo teste para mais de uma plataforma, reduzindo a manutenção necessária, já que o modelo atual funciona exclusivamente para Android, sendo necessário adaptar futuramente a mesma alternativa aos dispositivos Apple. Todavia, há a necessidade de ambas as plataformas possuírem nomenclaturas ou elementos em comum para que a interação do *script* seja possível.

#### 4 CONCLUSÃO

Conclui-se que, a adoção de testes de usabilidade é essencial, vide o impacto da má informação exposto no trabalho, para assegurar a qualidade de um sistema.

A nova metodologia de testes implementada pela empresa mostrou-se eficaz para a descoberta de falhas, bem como abre espaço para melhorias na eficiência de execução dos testes.

Contudo, ainda existem otimizações necessárias para a organização atingir excelência na validação de seu sistema, já que o mesmo é voltado as mídias móveis, de maneira que a necessidade de se testar o sistema sobre os mais diferentes cenários possíveis é uma preocupação necessária às organizações neste ramo. Deste modo, a contratação de um serviço de testes em nuvem se mostra uma ótima alternativa.

#### REFERÊNCIAS

CASTRO, A. **Test Automation in Enterprise Application Development**. 2013.

CHANDRA, R. *et al.* **Towards Scalable Automated Mobile App Testing**. MSR-TR-2014-44. Microsoft Research, 2014.

DEVELOPER XAMARIN. Disponível em:

<<http://developer.xamarin.com/guides/testcloud/calabash/introduction-to-calabash/>>. Acesso em: 20 ago. 2015.

GAO, J.; BAI, X.; TSAI, W. Cloud testing-issues, challenges, needs and practice. **Software Engineering: An International Journal**, v. 1, n. 1, p. 9-23, 2011. Disponível em:

<<http://seij.dtu.ac.in/Paper%201.pdf>>. Acesso em: 15 ago. 2015

GOGGIN, G. **Cell phone culture: Mobile technology in everyday life**. Routledge, 2012.

MYERS, G. J.; SANDLER, C.; BADGETT, T. **The art of software testing**. 3 ed. John Wiley & Sons, 2012.

PITOURA, E; SAMARAS, G. **Data management for *mobile computing***. Springer Science & Business Media, 2012.

SATYANARAYANAN, M. *Mobile computing: the next decade*. **ACM SIGMOBILE Mobile Computing and Communications Review**, v. 15, n. 2, p. 2-10, 2011. Disponível em: <<http://www.cse.chalmers.se/research/group/idc/ituniv/courses/12/mc/p2-satyanarayanan.pdf>>. Acesso em: 20 ago. 2015

WYNNE, M; HELLESoy, A. **The cucumber book: behaviour-driven development for testers and developers**. Pragmatic Bookshelf, 2012.

ZADGAONKAR, H. **Robotium Automated Testing for Android**. Packt Publishing Ltd, 2013.