

Uma Avaliação de Ferramentas para Testes em Sistemas de Informação Móveis baseada no Método DMADV

Ismayle S. Santos^{1,*}, Carla I. M. Bezerra^{1,2}, Gustavo S. Monteiro, Ítalo L. Araújo^{1,*},
Thalisson A. Oliveira^{1,*}, Rainara M. Santos^{1,*}, Valéria L. L. Dantas, Rossana M. C.
Andrade^{1,†}

Universidade Federal do Ceará (UFC), ¹ Mestrado e Doutorado em Ciência da Computação (MDCC), Departamento de Computação (DC), Grupo de Redes, Engenharia de Software e Sistemas (GREat), Fortaleza, CE – Brasil

² Campus Quixadá, CE – Brasil

{ismaylesantos, gustavomonteiro, valerialelli, italoaraujo, rainarasantos, thalissonoliveira, carlailane, rossana}@great.ufc.br

Abstract. *Mobile devices have become popular, and along with them, the Mobile Information Systems (MIS). Due to some devices constraints, the development of these systems becomes complex. Software testing is one of the activities aiming MIS quality assurance. Nevertheless, this activity is very costly. Aiming cost reduce, it is necessary make use testing tools. However, there are many commercial options, making difficult choose the greater tool. This paper presents a comparison between the main testing tools for MIS, using a DMADV-based method, defined by the Six Sigma methodology. The observed results should help to guide the choice of testing tools, as well as identify unsupported features.*

Resumo. *Os dispositivos móveis tem se popularizado e com eles os sistemas de informação móveis (SIM). Devido às restrições desses dispositivos, o desenvolvimento desses sistemas é complexo. Logo, para garantir qualidade aos SIMs é importante que se realize testes de software, mas essa atividade é bastante onerosa. Para reduzir os custos, pode-se usar ferramentas de teste. Contudo, existem várias opções no mercado dificultando a escolha da ferramenta ideal. Este artigo apresenta uma avaliação entre as principais ferramentas de teste para SIMs baseado na metodologia Six Sigma utilizando o DMADV. Os resultados obtidos podem guiar a escolha da ferramenta de teste, bem como são úteis para identificar as necessidades não supridas.*

1. Introdução

A complexidade para se testar um sistema de informação móvel (SIM), que é um sistema de informação que executa em um dispositivo móvel, exige uma adaptação no processo de teste tradicional [Dantas et al. 2009]. Essa adaptação decorre do fato de que não é possível executar testes em SIMs da mesma maneira que se realiza testes em outros tipos de aplicações. Isso é resultado das limitações do dispositivo móvel (e. g. restrição de

* Bolsista de Mestrado da CAPES

† Bolsista do CNPq de Produtividade DT-2 com o número de processo 314021/2009-4

hardware e limitações do ambiente de comunicação sem fio) [Dantas et al. 2009] e da própria natureza móvel [Ferreira 2007]. Além disso, vale ressaltar que o processo de teste de *software* requer tempo e recursos, o que torna o teste uma etapa onerosa do desenvolvimento de *software* [Shamsoddin-Motlagh 2012].

Com o intuito de reduzir os custos do teste, pode-se utilizar de ferramentas de apoio ao teste de software. Existem várias ferramentas de testes para sistemas de informação móveis disponíveis no mercado [Google 2013a, 2013b; Pivotal Labs 2013]. Contudo, não há na literatura uma avaliação e comparação dessas ferramentas de forma a guiar a escolha da ferramenta adequada de acordo com as necessidades dos testadores.

É importante destacar que a escolha de ferramentas de testes inadequadas pode trazer desperdício de tempo, além de poder influenciar negativamente no sucesso do produto [Veloso et al. 2010].

Nesse cenário, o Six Sigma [Tayntor 2003], que é uma metodologia de qualidade, pode ser utilizado por uma organização para seleção de ferramentas que atendam as suas necessidades. O Six Sigma apresenta alguns métodos de apoio a melhoria de processos como o DMAIC (*Define, Measure, Analyze, Improve, Control*) e o DMADV (*Define, Measure, Analyze, Design, Verify*). O DMAIC atua na melhoria de processos já existentes, enquanto que o DMADV, por sua vez, permite a introdução de novos produtos e processos de forma controlada e sistemática.

Dessa forma, neste trabalho é apresentado o uso do DMADV para avaliação de seis ferramentas de testes voltadas para sistemas de informação móveis. Uma vez que o DMADV utiliza critérios para analisar o objeto em avaliação, critérios ou medidas que pudessem ser utilizados para avaliar ferramentas de teste de SIMs precisavam ser identificados.

No trabalho de Velosos et al. (2010), 67 critérios foram utilizados para avaliar ferramentas de testes de sistemas de informação. Entretanto, esses critérios não contemplavam peculiaridades das ferramentas de testes específicas para aplicações móveis. Neste trabalho também é apresentada uma extensão dos critérios descritos em [Veloso et al. 2010] para cobrir os aspectos relevantes ao contexto de sistemas móveis. Esses critérios foram levantados com base na literatura e na experiência profissional do Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat³).

Os resultados deste trabalho são úteis para seleção de ferramentas de testes e para identificar que características consideradas importantes pelos testadores não são contempladas pelas principais ferramentas de testes de sistemas de informação móveis, identificando lacunas de pesquisa.

O restante deste artigo está organizado da seguinte forma: na Seção 2 são apresentados alguns trabalhos relacionados; na Seção 3 são apresentados os critérios adotados neste trabalho para avaliação de ferramentas de teste voltadas ao cenário móvel; na Seção 4 é apresentada a aplicação do DMADV para avaliação de seis ferramentas para testes de sistemas de informação móveis e, na Seção 5 são apresentadas as conclusões e direções para trabalhos futuros.

³ <http://www.great.ufc.br>

2. Trabalhos Relacionados

Embora seja conhecida a importância das ferramentas de testes no processo de testes, não foi encontrado nenhum trabalho relacionado à avaliação de ferramentas de teste para o cenário móvel. Entretanto, com respeito à avaliação de ferramentas foi possível identificar quatro trabalhos [Dórea et al. 2008, Wang 2009, Veloso et al. 2010 e Borjesson 2012] relacionados a esta pesquisa.

Em [Dórea et al. 2008] sete ferramentas de testes foram avaliadas empiricamente com respeito a critérios como desempenho, portabilidade, interoperabilidade e flexibilidade. Além disso, Dórea et al. (2008) realizaram um questionário com algumas empresas e constataram que as organizações procuram ferramentas que apresentam alto grau de desempenho e eficiência.

No trabalho de Wang e Offutt (2009), os autores compararam três ferramentas para testes de unidade. Esse trabalho consistia em injetar falhas dentro das classes na linguagem de programação Java através de uma ferramenta de mutação. Em seguida, um conjunto de dados de testes de unidades foi gerado manualmente e outros três foram gerados automaticamente pelas ferramentas. Logo, o objetivo era verificar quais conjuntos de dados de testes geravam melhores dados para encontrar as falhas injetadas nas classes. Os resultados mostraram, dentre outras conclusões, que as ferramentas de testes de unidades ainda são muito pobres em detectar falhas.

Em [Veloso et al. 2010], por sua vez, os autores apresentaram uma abordagem baseada em SQFD (*Software Quality Function Deployment*) [Haag 1996] para analisar e comparar ferramentas de teste. O processo de avaliação se iniciou com o levantamento de requisitos e aspectos técnicos que seriam necessários em uma ferramenta de teste. No trabalho de Veloso et al. (2010) foram avaliadas três ferramentas gratuitas voltadas ao teste funcional e três direcionadas ao teste de desempenho/estresse. Entre as ferramentas de teste funcional a que mais se adequou aos critérios identificados foi a *Selenium*⁴ e das ferramentas de teste de desempenho a que obteve melhor resultado foi a *Webload*⁵.

Por fim, o outro trabalho identificado foi o de Borjesson e Feldt (2012), no qual foram comparadas duas ferramentas (uma comercial e outra *open source*) que automatizam testes de GUI (*Graphical User Interface*). Os testes com essas ferramentas foram conduzidos em um ambiente industrial, em um sistema real e crítico. Foi avaliada principalmente a capacidade de automatizar os testes antes conduzidos manualmente. Após executar vários casos de testes com ambas as ferramentas, os autores obtiveram êxito no uso das duas ferramentas com um ganho de tempo de 78%.

Conforme o que foi apresentado nesta seção, é possível observar que não foi encontrado na literatura um trabalho que trata das peculiaridades do teste de um sistema de informação móvel que tem, por exemplo, que usar simuladores dos dispositivos móveis. Dessa forma, o diferencial deste artigo está no fato dele focar nas ferramentas de testes para sistemas de informação móveis, em utilizar um conjunto de critérios que reflete as necessidades dos testadores e no uso do método DMADV para efetuar a comparação das ferramentas de testes.

⁴ <http://seleniumhq.org/>

⁵ <http://www.webload.org/>

3. Critérios para avaliação de ferramentas de teste para aplicações móveis

Conforme mencionado na Seção de Trabalhos Relacionados, não foi encontrado na literatura nenhum trabalho voltado à avaliação de ferramentas de teste para SIMs. Dessa forma, por meio da definição de um conjunto de critérios e do uso do DMADV este trabalho se enquadra nesta lacuna existente.

O DMADV foi escolhido porque é conhecido na indústria para avaliação sistemática de novos produtos e é caracterizado pelo uso de métodos estatísticos e de engenharia. Mais detalhes sobre o DMADV são apresentados na Seção 4.

Já com relação aos critérios, foi identificado no trabalho de [Velo et al. 2010] um conjunto de 67 aspectos técnicos, que representam características consideradas importantes pelos testadores e que servem como critérios de avaliação de ferramentas de teste. Contudo, esses aspectos não contemplavam características peculiares do teste de aplicações móveis, como por exemplo, suporte ao teste levando em consideração características da bateria do dispositivo móvel.

Logo, baseado nos resultados de uma revisão sistemática sobre testes de aplicações móveis [Santos et al. 2012] e na experiência profissional do Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat) que desenvolve continuamente aplicações móveis, um conjunto de aspectos técnicos próprios para o cenário de sistemas de informação móveis foi consolidado.

A Tabela 1 apresenta os critérios de avaliação de ferramentas de testes para SIMs. No total foram identificados 08 aspectos técnicos:

- Simulador de plataformas de hardware: simulação de plataformas de hardware com menos poder computacional para a execução de testes, através de pausas na execução, controladas a partir de um cálculo comparativo entre o poder computacional da máquina executando os testes e a máquina que se deseja simular a execução;
- Simulador de banda de rede: simulação de banda de rede com menos poder de transmissão de dados para a execução de testes, através de pausas na execução, controladas a partir de um cálculo comparativo entre a largura de banda da rede executando os testes e a largura de banda que se deseja simular a execução;
- Testes em múltiplos dispositivos: suporte a execução dos testes simultaneamente em vários dispositivos móveis diferentes;
- Simulador de plataforma móvel: suporte para que o teste seja executado em mais de uma plataforma (e.g. Android, Iphone, Windows Mobile);
- Suporte ao teste em ambiente móvel: para possibilitar o teste da aplicação móvel simulando características específicas do ambiente móvel, tais como meios de comunicação (e. g. Wifi, 3G, Bluetooth) e restrições de memória;
- Verificação de Layout: simulação do layout da interface gráfica verificando fatores como o texto exibido na tela, os formatos retrato e paisagem, a renderização do conteúdo. Logo, com essa característica espera-se que a ferramenta de teste possibilite meios para identificar se as informações (texto e imagem) serão apresentadas integralmente na tela ou se serão cortadas;

- Teste de usabilidade: é interessante que a ferramenta de teste possua meios para coletar os dados de utilização do usuário de um sistema de informação móvel para que seja possível verificar algumas métricas de usabilidade, como o tempo que o usuário leva para executar determinada tarefa;
- Suporte ao teste do software embarcado: suporte ao teste nos aplicativos do próprio dispositivo, como por exemplo, o *menu* do dispositivo móvel.

Tabela 1. Grupo de critérios de avaliação direcionados ao contexto móvel

Grupo	Aspecto Técnico
Contexto móvel	Simulador de plataformas de hardware
	Simulador de banda de rede
	Teste em múltiplos dispositivos
	Simulador de plataforma móvel
	Suporte ao teste do ambiente móvel
	Verificação de Layout
	Teste de usabilidade
	Suporte ao teste do software embarcado

4. Uso do DMADV para avaliação de ferramentas de teste para SIMs

Com base nos critérios definidos em Veloso et al. (2010) e nos novos critérios apresentados na Seção 3, é apresentada na subseção 4.1 o projeto DMADV, derivado da metodologia Six Sigma, para avaliação de ferramentas de testes para SIMs. O detalhamento das fases do DMADV definido por Werkema (2013) é apresentado na Tabela 2. Os resultados da avaliação das ferramentas de testes são apresentados na subseção 4.2.

O método DMADV, é utilizado para implantação do *Design for Six Sigma* (DFSS), em projetos cujo escopo é a introdução de novos produtos e processos [Carvalho et al. 2007]. O DMADV é constituído por cinco etapas: a) *Define*: definir o novo produto ou processo a ser projetado; b) *Measure*: identificar as necessidades e traduzi-las em Características Críticas para a Qualidade (CTQs) - mensuráveis e priorizadas; c) *Analyze*: selecionar a melhor solução dentre as alternativas desenvolvidas com o uso de critérios estabelecidos; d) *Design*: desenvolver o novo produto ou processo, realizar os testes necessários e preparar a produção; e e) *Verify*: implantar e validar o projeto, implantar as melhorias e lançar o novo produto ou processo.

O método DMADV foi escolhido, porque é bastante conhecido e difundido em diversas áreas e que possibilita a seleção de novos produtos de forma sistemática e controlada. Adicionalmente, ele se baseia em ferramentas analíticas, possibilitando aos usuários a capacidade de prever e prevenir defeitos (ou problemas com esses produtos) ainda na fase de projeto de um produto, ou no caso deste trabalho, ainda na fase de seleção da ferramenta de teste.

Tabela 2. Principais objetivos e resultados das fases do DMADV [Werkema 2013]

		Objetivo	Principais resultados esperados
Etapa do DMADV	Define	Definir claramente o novo produto ou processo a ser projetado.	Justificativa para o desenvolvimento do projeto. Potencial de mercado para o novo produto. Análise preliminar da viabilidade técnica. Análise preliminar da viabilidade econômica. Previsão da data de conclusão do projeto. Estimativa dos recursos necessários.
	Measure	Identificar as necessidades dos clientes/consumidores e traduzi-las em características críticas para a qualidade (CTQs) - mensuráveis e priorizadas - do produto.	Identificação e priorização das necessidades dos clientes/consumidores. Análise detalhada do mercado. Características críticas do produto para o atendimento às necessidades dos clientes/consumidores.
	Analyze	Selecionar o melhor conceito dentre as alternativas desenvolvidas e gerar o <i>Design Charter</i> do projeto.	Definição das principais funções a serem projetadas para o atendimento às necessidades dos clientes/consumidores. Avaliação técnica dos diferentes conceitos disponíveis e seleção do melhor. Análise financeira detalhada do projeto.
	Design	Desenvolver o projeto detalhado (protótipo), realizar os testes necessários e preparar para a produção em pequena e em larga escala.	Desenvolvimento físico do produto e realização de testes. Análise do mercado e <i>feedback</i> de clientes/consumidores sobre os protótipos avaliados. Planejamento da produção. Planejamento do lançamento no mercado. Análise financeira atualizada do projeto.
	Verify	Testar e validar a viabilidade do projeto e lançar o novo produto no mercado.	Lançamento do produto no mercado. Avaliação da performance do projeto.

Na próxima subseção é apresentado como o DMADV pode ser utilizado para a seleção e avaliação de ferramentas de testes para sistemas de informação móveis.

4.1. Projeto DMADV para Seleção de Ferramentas de Testes para Aplicações Móveis

Conforme apresentado anteriormente, o DMADV foi originalmente direcionado para criar e avaliar novos produtos ou processos. Dessa forma, foi proposta uma adaptação do DMADV para que seja possível a avaliação de ferramentas de testes para SIMs.

Na fase *Definir*, deve ser elaborado a justificativa e os objetivos do desenvolvimento do novo produto. No caso deste trabalho, a justificativa está na ausência de um mecanismo para avaliar as ferramentas de testes para aplicações móveis existentes. Os objetivos do projeto DMADV para Seleção de Ferramentas de Testes para Sistemas de Informação Móveis são: a) Investigar e analisar as ferramentas de testes para aplicações móveis, disponíveis no mercado; e b) Melhorar os testes (em qualidade e produtividade) e obter uma redução na densidade de defeitos.

Ainda na fase *Definir* foram levantadas e sugeridas ferramentas para o projeto com especialistas da área, considerando as características críticas para a qualidade (CTQs) a partir das necessidades dos testadores. Das ferramentas indicadas pelos especialistas e obtidas mediante pesquisa na literatura, foram consideradas para avaliação apenas as ferramentas gratuitas: Robotium⁶, RoboElectric⁷, MonkeyRunner⁸, MonkeyTalk⁹, Scirocco¹⁰ e Calabash¹¹;

Os CTQs são fatores críticos definidos pelo cliente para o sucesso do produto ou processo, e podem ser representados como medidas [El-Haik e Shaout 2010]. Dos 67 aspectos técnicos do trabalho de Veloso et al. (2010), 35 foram selecionados como CTQs. O restante dos critérios foi excluído porque não representam características relevantes para execução de testes em SIMs com o apoio de uma ferramenta de teste.

Além disso, outros critérios foram definidos especificamente para o contexto móvel (CSM), conforme apresentado na Seção 3. No total foram utilizados 43 critérios, sumarizados na Tabela 3, para avaliar as ferramentas de testes de SIMs. Detalhes sobre os demais critérios (CSI) podem ser obtidos em [Veloso et al. 2010].

Tabela 3. CTQs do projeto

CSI.1	Gerador de povoador	CSI.23	Utilizar terminologia adequada ao contexto
CSI.2	Gerador de valores	CSI.24	Help on-line
CSI.3	Mecanismo de captura-reprodução	CSI.25	Manual de usuário
CSI.4	Uso de linguagens de alto nível	CSI.26	Sítio de apoio com exemplos de uso
CSI.5	Acesso as funções do Software Embarcado	CSI.27	Gerenciador de grids para execução distribuída de testes
CSI.6	Acesso ao mecanismo de persistência	CSI.28	Executor de teste com possibilidade de pausa, cancelamento, anotação e retomada da execução
CSI.7	Integração com ferramentas de acompanhamento de bugs	CSI.29	Agrupador e escalonador de testes
CSI.8	Integração com ferramentas de gerenciamento dos testes	CSI.30	Gerador de log de execução de testes
CSI.9	Integração com ferramentas de gestão de configuração	CSI.31	Integração com uma linguagem de script para configuração do ambiente de teste
CSI.10	Avaliador de cobertura	CSI.32	Comparador de arquivos ignorando padrões configuráveis
CSI.11	Gerador de testes de desempenho e estresse	CSI.33	Gerenciador de transações
CSI.12	Gerador de relatório com formato definido pelo usuário	CSI.34	Analisador de falhas
CSI.13	Gerador de gráficos com fonte de dados e formato definido pelo usuário	CSI.35	Cadastramento automático de falhas
CSI.14	Uso de hiperlinks e agrupadores nos relatórios	CSM.36	Simulador de plataformas de hardware
CSI.15	Cadastro de usuários	CSM.37	Simulador de banda de rede
CSI.16	Cadastro de grupos	CSM.38	Teste em múltiplos dispositivos
CSI.17	Cadastro de projeto	CSM.39	Simulador de plataforma móvel

⁶ <http://code.google.com/p/robotium/>

⁷ <http://pivotal.github.com/roboelectric/>

⁸ http://developer.android.com/tools/help/monkeyrunner_concepts.html

⁹ <https://www.gorillalogic.com/monkeytalk>

¹⁰ <http://code.google.com/p/scirocco/>

¹¹ <https://github.com/calabash/calabash-android>

CSI.18	Cadastro de equipe	CSM.40	Suporte ao teste do ambiente móvel
CSI.19	Mecanismo de busca para qualquer padrão informado pelo usuário	CSM.41	Verificação de Layout
CSI.20	Cadastro de perfis com possibilidade de definição de permissões por grupo, projeto, equipe	CSM.42	Teste de usabilidade
CSI.21	Uso de tecnologia Web	CSM.43	Suporte ao teste do software embarcado
CSI.22	Seguir um guia de estilo	--	--

Outra atividade realizada na fase *Medir* foi à definição e coleta das medições para mensurar os CTQs, conforme a Tabela 4. Tais medições foram coletadas, conforme a Tabela 5, por quatro especialistas na área de teste de aplicações móveis com uma média de três anos de experiência.

Os dados coletados na fase *Medir* avaliando as ferramentas de acordo com as medidas da Tabela 4 serviram como base para fase *Analisar*, onde foram analisadas e priorizadas as ferramentas identificadas de acordo com pesos e critérios. Esses pesos foram definidos baseados na relevância do CTQ no processo de seleção. Os valores informados, por sua vez, foram coletados pelos mesmos especialistas que conduziram as medições da fase *Medir*.

Tabela 4. Medidas definidas na Fase Medir para cada Critério

CTQ	Especificação das Medidas	CTQ	Especificação das Medidas
CSI 1	2 - Gera código de povoamento a partir de uma instância; 1 - Gera código de povoamento; 0 - Não gera código de povoamento.	CSI 23	2 - Uso da terminologia adequada; 1 - Uso parcial da terminologia adequada; 0 - Não utiliza terminologia adequada.
CSI 2	2 - Gera valores aleatórios (na ausência de modelos) E gera com base em modelos; 1 - Gera valores com base em modelo definido pelo usuário OU completamente aleatórios; 0 - Não gera valores.	CSI 24	2 - Possui dois ou mais "modos" de help-online; 1 - Possui fórum OU FAQ OU atendente on-line; 0 - Não possui help on-line.
CSI 3	2 - Grava testes e permite a edição dos scripts de testes gerados; 1 - Grava testes, mas não permite a edição destes; 0 - Não grava testes.	CSI 25	2 - Possui manual de usuário; 0 - Não possui manual de usuário;
CSI 4	2 - Permite o uso de linguagens de alto nível; 0 - Não permite o uso de linguagens de alto nível;	CSI 26	2 - Possui Sítio de apoio com exemplos de uso; 1 - Possui Sítio de apoio, mas sem exemplos de uso; 0 - Não possui Sítio de apoio
CSI 5	2 - Acessa o software embarcado; 0 - Não acessa o software embarcado;	CSI 27	2 - Possibilita a distribuição do teste para executar em vários celulares ou emuladores; 0 - Não possibilita a distribuição dos testes
CSI 6	2 - Permite acesso ao mecanismo de persistência para leitura e escrita; 1 - Permite acesso ao mecanismo de persistência, mas somente para leitura; 0 - Não permite acesso aos mecanismos de persistência;	CSI 28	2 - Testes executados com o controle do usuário; 0 - Testes executados sem controle do usuário.
CSI 7	2 - Possui integração com uma ferramenta de acompanhamento de bugs; 0 - Não possui integração com uma ferramenta de acompanhamento de bugs.	CSI 29	2 - Com agrupador E escalonador de testes; 1 - Com agrupador OU escalonador de testes; 0 - Sem agrupador E sem escalonador de testes
CSI 8	2 - Possui integração com uma ferramenta de gerenciamento dos casos e procedimentos de testes; 0 - Não possui integração com uma ferramenta de gerenciamento dos testes.	CSI 30	2 - Apresenta log e tira <i>screenshots</i> da tela; 1 - Apresenta log de execução dos testes; 0 - Não apresenta log de execução dos testes.
CSI 9	2 - A ferramenta apresenta uma gestão de configuração; 0 - A ferramenta não apresenta uma gestão de configuração.	CSI 31	2 - Possui suporte de uma linguagem de script; 0 - Não apresenta integração com uma linguagem de script.
CSI 10	2 - A ferramenta analisa a cobertura de execução dos testes, apresenta resultado de métricas e uma representação visual; 1 - A ferramenta analisa a cobertura de execução dos testes e apresenta resultados de métricas de cobertura; 0 - A ferramenta não analisa a cobertura dos testes.	CSI 32	2 - Possui comparador de arquivos ignorando padrões configuráveis; 1 - Possui comparador de arquivos; 0 - Não possui comparador de arquivos.

CSI 11	2 - Apresenta suporte para execução de testes de desempenho E os gera automaticamente 1 - Apresenta suporte para execução de testes de desempenho, mas não os gera automaticamente 0 - Não gera teste de desempenho.	CSI 33	2 - Possui gerenciador de transição; 0 - Não possui gerenciador de transição
CSI 12	2 - Gera relatório com formato definido pelo usuário; 1 - Gera relatórios em um formato pré-definido; 0 - Não gera relatórios.	CSI 34	2 - Possui avaliador de falhas; 0 - Não possui avaliador de falhas.
CSI 13	2 - Gera gráficos com formato definido pelo usuário; 1 - Gera gráficos em um formato pré-definido; 0 - Não gera gráficos.	CSI 35	2 - Com cadastramento automático de falhas; 0 - Sem cadastramento automático de falhas.
CSI 14	2 - Com hiperlinks e agrupadores de relatório; 1 - Com hiperlinks ou agrupadores de relatório; 0 - Sem hiperlinks ou agrupadores de relatório.	CSM 36	2 - Com simulador de hardware específico para celulares (ex.: celular Motorola X120) 1 - Com simulador de hardware para uma linha (ex.: modelos LG 200); 0 - Sem simulador de hardware.
CSI 15	2 - Com cadastro de usuários; 0 - Sem cadastro de usuários.	CSM 37	2 - Com simulador de banda de rede; 0 - Sem simulador de banda de rede.
CSI 16	2 - Com cadastro de grupos; 0 - Sem cadastro de grupos.	CSM 38	2 - Permite execução em múltiplos dispositivos; 0 - Não permite a execução em múltiplos dispositivos;
CSI 17	2 - Com cadastro de projeto; 0 - Sem cadastro de projeto.	CSM 39	2 - Possui simulador para duas ou mais plataformas móveis (JME, Android, Windows Mobile); 1 - Possui simulador para apenas uma plataforma móvel; 0 - Não possui simulador para plataforma móvel.
CSI 18	2 - Com cadastro de equipe; 0 - Sem cadastro de equipe.	CSM 40	2 - Apresenta suporte para o teste de comunicação E de armazenamento; 1 - Apresenta suporte para o teste de comunicação OU de armazenamento; 0 - Não apresenta suporte ao teste do ambiente móvel;
CSI 19	2 - Mecanismo de busca para qualquer padrão; 1 - Mecanismo de busca limitado; 0 - Sem mecanismo de busca.	CSM 41	2 - Apresenta suporte para verificação de layout 0 - Não apresenta nenhum meio para verificação de Layout
CSI 20	2 - Permite definição de permissão por grupos e por equipes; 1 - Permite definição de permissão por grupos ou por equipes; 0 - Não existe mecanismo de definição de permissão	CSM 42	2 - Permite coleta de dados (log) de utilização do usuário com a aplicação móvel em qualquer situação de uso; 0 - Não permite coleta de dados.
CSI 21	2 - Funcionamento possível via web; 1 - Funcionamento somente via web; 0 - Não possui funcionamento via web.	CSM 43	2 - Permite a execução do teste no próprio dispositivo; 0 - Não permite a execução no próprio dispositivo
CSI 22	2 - Possui interface de usuário e segue um guia de estilo; 1 - Possui interface de usuário, mas não segue um estilo; 0 - Não possui uma interface de usuário (própria);		

A análise das ferramentas de acordo com os pesos e critérios é apresentada na Tabela 5. A porcentagem de atendimento dos critérios de cada ferramenta foi dada conforme a fórmula:

$$r_f = \frac{\sum_{i=1}^{43} (p_i * m_i)}{\sum_{i=1}^{43} p_i}$$

r_f = resultado da ferramenta
 p_i = peso da medida i
 m_i = resultado da medida i

Dessa forma, conforme pode ser verificado na Tabela 5, as ferramentas que possuíram melhores avaliações de acordo com os critérios e a importância dos pesos dados, foram Scirocco, RoboElectric e Robotium, com percentuais de atendimento aos critérios de 79%, 77% e 76% respectivamente.

As fases Projetar e Verificar do método DMADV não foram executadas, pois estes devem se utilizar de projetos reais para análise das medições. No entanto, todas as medições coletadas de acordo com os critérios disponíveis foram analisadas por especialistas diretamente nas ferramentas.

Tabela 5. Análise das ferramentas de acordo com os critérios e pesos

Critérios	Pesos	Ferramentas de testes para Sistemas de Informação Móveis					
		Robotium	Robolectric	MonkeyRunner	MonkeyTalk	Scirocco	Calabash
CSI 1	2	0	0	0	0	0	0
CSI 2	2	1	1	1	1	1	0
CSI 3	4	0	0	1	2	0	0
CSI 4	5	2	2	2	2	2	2
CSI 5	3	0	0	0	0	0	0
CSI 6	4	2	2	0	0	2	0
CSI 7	3	0	0	0	0	0	0
CSI 8	3	0	0	0	0	0	0
CSI 9	3	2	2	2	2	2	2
CSI 10	4	1	2	1	1	1	1
CSI 11	3	0	0	0	0	0	0
CSI 12	4	1	1	1	1	1	1
CSI 13	3	0	0	0	0	0	0
CSI 14	2	1	1	1	0	1	0
CSI 15	1	0	0	0	0	0	0
CSI 16	1	0	0	0	0	0	0
CSI 17	1	0	0	0	0	0	0
CSI 18	1	0	0	0	0	0	0
CSI 19	3	1	1	0	1	1	0
CSI 20	1	0	0	0	0	0	0
CSI 21	3	0	2	0	2	0	2
CSI 22	2	1	1	1	1	1	1
CSI 23	2	1	1	1	1	1	1
CSI 24	4	2	2	0	2	1	1
CSI 25	5	2	2	2	2	0	0
CSI 26	4	2	2	2	2	2	2
CSI 27	3	0	0	0	0	0	0
CSI 28	4	0	0	0	2	0	0
CSI 29	2	1	1	1	1	1	0
CSI 30	5	0	0	0	0	2	2
CSI 31	3	2	2	2	2	2	2
CSI 32	2	0	0	1	0	0	0
CSI 33	2	0	0	0	0	0	0
CSI 34	3	0	0	0	0	0	0
CSI 35	2	0	0	0	0	0	0
CSM 36	5	0	0	0	0	0	0
CSM 37	4	0	0	0	0	0	0
CSM 38	4	0	0	2	0	2	0
CSM 39	5	1	0	1	2	1	2
CSM 40	5	1	2	0	0	1	0
CSM 41	3	2	2	0	0	2	0
CSM 42	4	0	0	0	0	0	0
CSM 43	4	2	0	2	0	2	2
Resultado		0,76	0,77	0,64	0,74	0,79	0,60

4.3. Discussão dos resultados

Diante dos resultados obtidos com a avaliação das ferramentas de testes é possível observar que nenhuma das ferramentas de testes avaliadas atende uma porção significativa dos critérios usados para avaliação. De fato, boa parte dos critérios (18) não foi atendida por nenhuma das ferramentas analisadas, mesmo sendo critérios relevantes como, por exemplo, integração com ferramentas de bugs (CSI7) e integração com ferramentas de gerenciamento dos testes (CSI8).

Com relação aos critérios que foram criados especificamente para tratar do contexto móvel, as ferramentas avaliadas não atenderam critérios considerados importantes como “Simulador de plataforma de hardware” (CSM36), “Simulador de Banda de Rede” (CSM37) e “Suporte ao teste do software embarcado” (CSM43).

Dessa forma, assim como Veloso et al. (2010) apresenta indícios de que ainda há muito o que fazer com relação às ferramentas de apoio aos testes funcionais e de desempenho, este trabalho apresenta indícios que também há muito o que fazer com

relação a ferramentas de apoio ao teste de sistemas de informação móveis. Contudo, é importante ressaltar que é difícil que haja uma única ferramenta que atenda a todos os critérios apresentados neste trabalho devido à complexidade de desenvolvimento envolvida. O mais provável, é que uma cobertura maior dos critérios possa ser atendida utilizando um conjunto de ferramentas de testes.

Quanto ao uso do DMADV, com poucas adaptações ele se mostrou apropriado para a realização da avaliação das ferramentas de testes. Além disso, embora neste trabalho não tenha sido realizado as últimas duas etapas do método (*Projetar e Verificar*), por necessitar do uso de cada ferramenta em projetos reais, acredita-se que os resultados são válidos, pois todo o processo foi acompanhado por especialistas.

Por fim, vale mencionar que a avaliação de ferramentas de teste depende do contexto no qual ela será aplicada e das necessidades dos testadores. Logo, o propósito deste trabalho não foi indicar a “melhor ferramenta”, mas sim apresentar um meio para que empresas de desenvolvimento de software possam fazer essa avaliação. Além disso, os resultados apresentados neste artigo indicam lacunas de pesquisas.

5. Conclusões

Este artigo apresenta o uso do DMADV para avaliação de ferramentas de testes para sistemas de informação móveis. O método DMADV foi escolhido porque possibilita a seleção das ferramentas de forma sistemática.

Com o DMADV e os critérios apresentados neste trabalho é possível avaliar e comparar ferramentas de testes de aplicações móveis. Essa avaliação/comparação é relevante porque fornece informações úteis para identificar a ferramenta que melhor se adequa as necessidades de uma determinada empresa ou determinado projeto de desenvolvimento. É importante mencionar que o uso da ferramenta adequada tende a facilitar o projeto e execução dos testes e a proporcionar uma maior qualidade aos testes.

Seis ferramentas gratuitas de teste de sistemas de informações móveis foram avaliadas usando um conjunto de 43 critérios, sendo oito próprios para aplicações móveis. Os resultados indicaram que as ferramentas que mais atendem aos critérios utilizados para análise são o *RoboEtric*, o *Scirocco* e o *Robotium*. Contudo, a avaliação também revelou que boa parte dos critérios não são atendidos pelas ferramentas atuais e que somente uma combinação das ferramentas possibilita uma cobertura maior das necessidades dos testadores.

Uma limitação deste trabalho foi a não execução de todas as fases do método DMADV. Dessa forma, como trabalhos futuros pretende-se executar as fases *Projetar* e *Verificar* desse método, de forma a executar as ferramentas avaliadas em projetos reais e fazer novas medições e algumas análises estatísticas. Os resultados dessas fases serviriam então, para averiguar em projetos reais que os resultados obtidos com a avaliação realmente indicaram quais as melhores ferramentas de testes para os critérios analisados.

Agradecimentos

Este artigo é um resultado parcial do projeto UbiStructure com apoio do CNPq (MCT/CNPq 14/2011 - Universal) com o número 481417/2011-7 e do projeto Maximum da FUNCAP/FAPs/INRIA/INS2i-CNRS com o número 0064-000120100/12.

Referências

- Borjesson, E.; Feldt, R. (2012). Automated System Testing Using Visual GUI Testing Tools: A Comparative Study in Industry, In IEEE Fifth International Conference on Software Testing, Verification and Validation, vol., no., pp.350-359, 17-21.
- Dantas, V. L.L. ; Marinho, F. G.; Costa, A. L.; Rossana M. C. Andrade (2009). Testing Requirements for Mobile Applications. In Proceedings of the 24th International Symposium on Computer and Information Sciences. Guzelyurt: IEEE, p. 555-560.
- Dórea, A. D. O.; Carvalho, F. S.; Santos, M. T.; Neto, M. C. M.; Moises, D. (2008). Avaliação de Ferramentas de Automação para Engenheiros de Testes. In: Anais do IV Simpósio Brasileiro de Sistemas de Informação. RJ: SBC. v. 1. p. 23-34.
- El-Haik, Basen S. e Shaout, Adnan. (2010) “Software Design for Six Sigma: A Roadmap for Excellence”. Wiley, 1 edition.
- Ferreira, D. S. (2007) Abordagem Híbrida para Avaliação da Usabilidade de Dispositivos Móveis. Dissertação (Mestrado), Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande.
- Google (2013a). Robotium: Android test automation framework. Disponível em: <http://code.google.com/p/robotium/>. Último acesso em janeiro de 2013.
- Google (2013b). MonkeyRunner. Disponível em: http://developer.android.com/tools/help/monkeyrunner_concepts.html. Último acesso em janeiro de 2013.
- Haag, S.; Raja, M. K.; Schkade, L. L. (1996). Quality function deployment usage in software development. In Communications of the ACM, 39(1):41 – 49.
- Pivotal Labs (2013). Robolectric. Disponível em: <http://pivotal.github.com/robolectric/>. Último acesso em janeiro de 2013.
- Santos, I. S.; Dantas, V. L. L.; Santos, R. M. (2012). Teste de aplicações móveis: uma análise das pesquisas científicas via revisão sistemática. Anais do Simpósio Brasileiro de Qualidade de Software.
- Shamsoddin-Motlagh, E. (2012). A Review of Automatic Test Cases Generation. In International Journal of Computer Applications, v. 57, n. 13, p. 25–29
- Tayntor, C. B. (2003). Six Sigma Software Development, Flórida, Auerbach
- Veloso, J. ; Santos Neto, P.; Santos, I. S. ; Britto, R.S. (2010) Avaliação de Ferramentas de Apoio ao Teste de Sistemas de Informação. In iSys: Revista Brasileira de Sistemas de Informação, v. 3, p. 4, 2010.
- Wang, S. and Offutt, J. (2009). Comparison of Unit-Level Automated Test Generation Tools. In IEEE International Conference on Software Testing Verification and Validation Workshops, pp. 210-219.
- Werkema, C. (2013) Integração entre o Design for Six Sigma (DFSS) e a Metodologia de Gerenciamento de Projetos PMBoK. Disponível em: <http://www.leansixsigma.com.br/>. Último acesso em janeiro de 2013.