Contribuições à Qualidade de um Produto com Testadores de Diferentes Perfis: Um Relato de Experiência

Relato de Experiência

Mirian Souza[1], Ellen Barroso[1], Andréia Vieira[1], Adriane Almeida[1], Lucas Gomes[1], Arilo Dias Neto[2]

[1]FPF Tech - Fundação Desembargador Paulo Feitoza
Manaus – Amazonas – Brasil

[2]Instituto de Computação (IComp) – Universidade Federal do Amazonas (UFAM)
Manaus – AM – Brasil

{msouza, ellen.barroso, avieira, adriane.almeida, lucas.gomes}@fpf.br
arilo@ icomp.ufam.edu.br

Abstract. *This paper presents an experience report of a software testing project that had as main objective to verify the acceptance of a product in the Brazilian market. The testing of this product should be executed by different user profiles, in five regions of Brazil. As result, it was observed that the collaboration of different profiles testing the same product was important to identify faults and improvements, adding value to product quality.*

Resumo. *Este artigo apresenta um relato de experiência de um projeto de testes de software que tinha como principal objetivo verificar a aceitação de um produto no mercado brasileiro. Os testes desse produto deveriam ser realizados por diferentes perfis de usuários, nas cinco regiões do Brasil. Como resultado, foi possível observar que a colaboração de diferentes perfis testando o mesmo produto foi importante na identificação de falhas e melhorias, agregando valor à qualidade do produto.*

1. Introdução

As grandes empresas, antes de oferecerem seus produtos ou serviços no mercado, procuram conhecer melhor o nicho que irão trabalhar a fim de oferecer um produto ou serviço adequado às características locais. Este cenário não é diferente em empresas que desenvolvem software. Estas visam à aplicação de estratégias de teste de software como mecanismo para garantir a qualidade de seus produtos.

Cada nível de teste (unidade, integração, sistema ou aceitação) possui um propósito diferenciado na busca pela avaliação da qualidade de um software, pois consideram diferentes perspectivas. Sob o ponto de vista do desenvolvedor, são realizados testes de unidade e integração. Nestes níveis, a qualidade está na facilidade de se manter esse produto e atender aos requisitos definidos. Já na perspectiva do cliente, podem ser executados os níveis de teste de sistema e aceitação, que além dos requisitos, buscam ainda avaliar outras características, como por exemplo, o retorno financeiro ou marketing por trás de um novo software.

Assim, uma das formas de validar a aceitação do produto a ser inserido no mercado é realizando testes de aceitação com um público diversificado, de forma a identificar pontos fortes e pontos a serem melhorados. Neste sentido, executar testes de aceitação de software utilizando diferentes perfis de usuário auxilia na definição do conceito de qualidade para aquele produto. Isso é importante, pois existem usuários com

diferentes perfis, como por exemplo, aqueles que esperam que o software funcione perfeitamente, seja intuitivo, eficiente e seguro. Assim, pode ser difícil definir de forma exata a qualidade de software, que segundo Pressman (2011) pode ser definida como: "uma gestão de qualidade efetiva aplicada de modo a criar um produto útil que forneça valor mensurável para aqueles que produzem e para aqueles que utilizam".

Neste artigo é apresentado um relato de experiência conduzido com uma empresa internacional de grande porte, de nome fictício *Mobile Apps*, que possui uma loja de aplicativos para dispositivos na plataforma Android disponível e utilizada em diferentes países. Antes de lançar seu produto no mercado brasileiro, houve a preocupação com a aceitação desse produto no mercado local. Desse modo, essa empresa contratou o Instituto de Pesquisa e Desenvolvimento FPF Tech (ou simplesmente FPF Tech), localizado em Manaus/Amazonas, para coletar a percepção dos potenciais usuários desse produto utilizando diferentes dispositivos e conexões. Assim, este relato de experiência descreve o trabalho realizado pela FPF Tech nessa loja de aplicativos, no qual foram realizados testes com perfis diversificados de usuários, utilizando diferentes modelos de smartphones e tablets, diversas operadoras telefônicas, diferentes tipos de conexão de dados em diferentes regiões do Brasil. O artigo apresenta ainda todas as etapas realizadas no projeto, identificando a contribuição que diferentes visões de usuários trouxeram para o produto.

Além desta seção introdutória, este artigo possui mais quatro seções. A Seção 2 apresenta uma breve revisão de literatura sobre testes de software e alguns tipos de testes utilizados no relato. A Seção 3 apresenta o relato de experiência propriamente dito, descrevendo todas as etapas conduzidas ao longo do projeto. A Seção 4 apresenta os resultados obtidos no decorrer do projeto e uma análise desses resultados. Por fim, a Seção 5 descreve as considerações sobre o trabalho desenvolvido e trabalhos futuros.

2. Testes de Software

Um dos principais objetivos da execução de testes é garantir a qualidade de software. Por isso, pode-se dizer que ao executar testes há a garantia de que o software terá um baixo índice de falhas quando entrar em produção, atingindo assim os requisitos estabelecidos. Segundo Sommervile (2007), "[...] a meta do teste de software é convencer os desenvolvedores e clientes do sistema de que o software é bom o suficiente para o uso operacional. O teste é um processo voltado a atingir a confiabilidade do software".

Dessa forma, para que o projeto relatado fosse concluído com êxito, foram utilizados os conceitos de testes de aceitação, testes exploratórios e testes distribuídos de software, visto que supriam as exigências do cliente. Neste projeto, não foi viável aplicar os critérios tradicionais de testes funcionais descritos na literatura técnica, em função de a documentação do produto não ter sido disponibilizada pelo cliente/desenvolvedor.

2.1 Teste de Aceitação

O principal conceito de teste de aceitação é verificar se o que foi entregue para testes satisfaz a real expectativa e necessidade do usuário com uma comparação aos requisitos do projeto. Para isso, geralmente são criados casos de teste que avaliam situações anormais de uso (exceções) das funcionalidades contratadas. Assim, se os testes forem aprovados, significa que o programa está aceito. Segundo Myers (2011), "Teste de aceitação é o processo de comparação que ocorre no programa entre os requisitos iniciais do projeto e a necessidade atual dos usuários finais".

No projeto em questão, foram utilizados alguns conceitos de teste de aceitação, já que o aplicativo tinha uma gama de funcionalidades já prontas, no qual era dever do analista de teste verificar se o aplicativo atingia as necessidades do usuário de acordo com o que foi contratado. Como o aplicativo também foi testado por pessoas que não tinham experiência em teste de software, o *feedback* do teste de aceitação retratou diretamente a satisfação do usuário sobre o aplicativo de forma geral.

2.2 Testes Exploratórios

Segundo Whittaker (2009), testes exploratórios são descritos da seguinte forma: "Quando os scripts são totalmente removidos, o processo é chamado de teste exploratório. Testadores podem interagir com a aplicação da maneira que eles quiserem e utilizar as informações que o aplicativo fornece para reagir, mudar a direção dos testes e, geralmente, explorar a funcionalidade da aplicação sem restrições".

O teste exploratório tende a ser mais ágil que os outros. Porém, nem sempre utilizando este tipo de teste é possível ter o controle dos testes, especialmente quando vários testadores forem envolvidos nos testes do mesmo software. Assim, Bach (2010) sugere a utilização da técnica de testes exploratórios baseados em sessões, possibilitando o gerenciamento dos testes. Nessa técnica, cada sessão equivale a uma missão a ser cumprida durante o teste. Antes de iniciar os testes, o testador reserva um tempo para navegar pelo software com a finalidade de conhecer seus recursos e áreas que poderão ser abordadas durante a execução dos testes e documenta de forma sucinta os pontos que serão explorados. O resultado deste trabalho é uma documentação chamada *charter*.

No projeto relatado neste artigo, foram especificados *charters* definindo apenas as áreas a serem exploradas, deixando livre o roteiro seguido para execução. Isso tornou possível o controle dos testes que seriam executados em diferentes regiões do Brasil.

2.3 Teste Distribuído de Software

Para que se compreenda a dinâmica do Teste Distribuído de Software, Collins et al. (2012) esclarecem que em um cenário de Desenvolvimento Distribuído de Software (DDS) existe uma divisão das tarefas realizadas pelas equipes que trabalham em conjunto, porém de forma descentralizada, no desenvolvimento de um projeto. Assim, Teste Distribuído de Software (TDS) segue a mesma estrutura citada do DDS.

Segundo Audy e Prikladnicki (2007), apesar das atividades de TDS serem similares às atividades tradicionais de testes, existem desafios que vão além de executar os testes para reportar falhas, sendo indispensável o planejamento e controle das tarefas a fim de sempre manter a comunicação, colaboração e controle da equipe distribuída.

No caso do projeto relatado neste artigo, a finalidade foi a distribuição de atividades para um time de testadores separados geograficamente e que possuem a função de realizar e seguir procedimentos específicos para a execução de testes no projeto.

3. Contextualização do Relato

De acordo com o exposto anteriormente, a FPF Tech foi contratada para realizar um projeto cujo objetivo era executar testes em um produto de venda/download de aplicativos para dispositivos Android, que já era utilizado no mercado estrangeiro, visando validar o funcionamento e aceitação dele no mercado brasileiro, considerando:

- Diferentes tipos de usuários usando o produto (mesclar analistas de teste, profissionais da área de software e usuários que não eram da área de software);
- Diferentes configurações de *smartphones* e *tablets* da plataforma Android;

- Diferentes níveis de conexões de internet;
- Diferentes regiões do Brasil.

O planejamento dos testes e definição dos perfis das pessoas que executariam os testes foi baseado nas solicitações do cliente e será detalhado nas seções seguintes.

3.1 Planejamento dos Testes

Foi planejada a execução de 1.360 horas de testes de aceitação no produto em 17 dias, sendo 80 horas diárias de testes que deveriam ser executadas por diferentes pessoas em diferentes regiões do Brasil. Cada pessoa teria que executar apenas 1h de teste por dia utilizando diferentes dispositivos e conexões.

Foi estabelecido pelo cliente que os testes deveriam ser executados não apenas por profissionais da área de testes, mas também pessoas com e sem habilidade na utilização de dispositivos móveis e pessoas da área computação sem especialidade em testes. Tais pessoas estariam distribuídas em diferentes regiões. Mediante esta premissa, foram definidos os perfis de pessoas que deveriam testar o produto em questão:

- **Analistas de teste:** Profissionais formados na área de Computação ou afins, com experiência entre 6 meses e 10 anos na área de testes.
- **Desenvolvedores:** Profissionais formados na área de Computação ou afins, com experiência entre 6 meses e 7 anos na área de desenvolvimento de software.
- **Alunos de Centros de Treinamento em Informática (CTI):** Estudantes de curso técnico de informática, sem nenhuma experiência em testes. A maioria nunca tinha tido contato com os dispositivos usados para execução dos testes.
- **Universitários:** Alunos de cursos de Computação sem experiência profissional na área de teste, mas que já haviam cursado a disciplina de testes de software, e com conhecimento prévio em dispositivos Android.
- **Outras áreas:** Profissionais graduados em áreas diferentes de software, com habilidades para usar dispositivos Android.

A Tabela 1 apresenta o percentual de pessoas envolvidas nos testes por perfil.

Tabela 1: Distribuição de pessoas por perfil.

Perfil	Quantidade de pessoas	Percentual de Pessoas
Analistas de Teste	23	~28%
Desenvolvedores	13	~16%
Universitários	15	~18%
Alunos dos CTIs	20	~25%
Outras Áreas	9	~11%

A divisão de pessoas por região ficou estabelecida da seguinte forma: Norte (70 pessoas); Nordeste e Sul (3 pessoas, cada); Sudeste e Centro Oeste (2 pessoas, cada). Os testes foram concentrados na região Norte, uma vez que a FPF Tech está localizada fisicamente nessa região, possibilitando, assim, um melhor gerenciamento dos testes.

Como não havia documentação detalhada do produto a ser testado, foi definido que o tipo de teste a ser executado seria o exploratório. Para tanto, foi planejada a especificação de *charters* por profissionais de testes da FPF Tech. Eles serviriam de base para a execução dos testes de modo a uniformizar as atividades a serem realizadas pelos testadores, visto que nem todos eram profissionais da área de testes.

Ainda na fase de planejamento, foi criada uma planilha contendo o planejamento do ambiente de execução dos testes, indicando qual dispositivo deveria ser utilizado nos testes, conexão, operadora, login e usuário de acesso. A definição do ambiente de execução foi estabelecida pelo cliente. Nesta fase, também foi definido um padrão para reporte de falhas, de forma que fosse possível obter informações detalhadas sobre falhas/melhorias identificadas, conforme estabelecido na Tabela 2. Como nem todos os envolvidos eram Analistas de Testes, algumas informações do registro de falhas eram revisadas e ajustadas por profissionais da FPF Tech.

Tabela 2: Informações reportadas para cada falha.

Dado	Descrição
Data:	Data na qual a falha foi identificada.
Login:	Conta do usuário que identificou a falha.
Versão do aplicativo:	Versão do produto testado em que a falha foi identificada.
Tipo de falha identifica:	Falha ou melhoria.
Categoria:	Categoria da falha identificada (funcional, usabilidade, localização).
Região:	Região do país na qual a falha foi identificada.
Resumo:	Breve descrição da falha identificada.
Nome do aplicativo:	Indicação do aplicativo onde a falha ocorreu, quando a falha era resultante da compra/download de aplicativo.
Frequência:	Frequência de ocorrência (Sempre ou intermitente).
Passos:	Passos detalhados para reproduzir a falha.
Resultado encontrado:	Resultado anormal encontrado após a execução dos passos.
Resultado esperado:	Resultado esperado para a situação avaliada.

3.2 Execução dos Testes Distribuídos

Antes de iniciar a execução dos testes no projeto, foi realizada uma reunião com todos os envolvidos localizados na região Norte para explicar de forma detalhada o projeto, as metas, objetivos, como os testes seriam executados, como as falhas identificadas seriam reportadas e as regras definidas, como por exemplo:

- Ao iniciar os testes, não pode haver interrupções durante 1 hora;
- Os testes devem ser realizados nos dispositivos (modelo e fabricante) e conexões definidas previamente;
- As falhas e/ou melhorias identificadas devem ser registradas no mesmo dia e no formato definido;
- Envio de relatório diário com reporte das atividades realizadas as pessoas localizadas nas demais regiões, foi enviada uma apresentação detalhada explicando os mesmos itens.

Os testes exploratórios baseados em *charters* foram executados por todos os envolvidos nos testes, que também executaram a exploração improvisada (sem orientação de *charters*).

As falhas e melhorias identificadas eram registradas no mesmo dia e seguindo o padrão definido, contendo *screenshots* e logs, quando aplicável, conforme apresentada na Tabela 2. O monitoramento e controle ocorreu em paralelo às atividades executadas.

3.3 Gerenciamento dos Testes

Foi definido um plano de comunicação, onde constava a lista de todos os envolvidos e seus contatos. O contato com pessoas localizadas em outras regiões foi realizado por e-mail, mensageiros e por telefone.

Para fazer o acompanhamento e condução dos testes realizados por pessoas localizadas em diferentes áreas na região norte e para monitorar os testes realizados nas demais regiões, foram alocados neste projeto, em tempo integral, três Analistas de Teste da FPF Tech.

Diariamente, a execução dos testes era acompanhada de forma a verificar se os testes estavam sendo executados conforme o planejamento ou se algum executor estava tendo algum tipo de impedimento.

As falhas identificadas também eram monitoradas, era realizada uma análise de modo que as falhas duplicadas eram desconsideradas e apenas as falhas sem duplicidade eram encaminhadas ao cliente. Diariamente, era realizada uma reunião via conferência telefônica com o cliente, para reporte da situação da execução dos testes. Ao final dos testes de cada versão, um relatório era enviado ao cliente passando um *feedback* acerca do resultado dos testes, contendo algumas métricas, como: Quantidade de falhas e melhorias por versão; Percentual de horas de testes executados por tipo de conexão de dados; Percentual de aplicativos comprados, percentual de aplicativos baixados gratuitamente por região.

4 Resultados Obtidos

Ao longo do projeto, quatro versões do produto foram testadas. Todas as informações sobre as falhas encontradas pelos diferentes perfis de usuário foram registradas em uma ferramenta interna da FPF Tech, tornando possível a coleta de métricas.

A Figura 1 mostra que a grande maioria (94%) dos itens identificados eram falhas. Pelo fato do número de sugestões de melhorias ser pequeno, todos os itens foram referenciados como falha neste relato. Foi observado também que do total de falhas identificadas, 81% eram válidas e 19% eram duplicadas, conforme pode ser observado na Figura 2. Quando a mesma falha era identificada mais de uma vez, o crédito da identificação era atribuído ao perfil que a identificou primeiro, pois foi acordado com o cliente que as falhas identificadas fossem enviadas para ele imediatamente. Por isso, a primeira identificação de uma falha foi chamada de "válida" neste projeto.

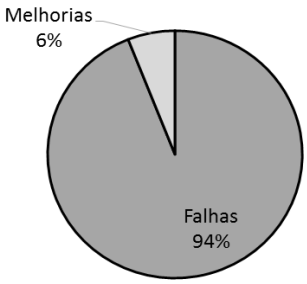


Figura 1: Percentual de Falhas x Melhorias

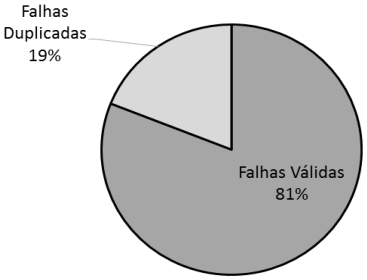


Figura 2: Percentual de falhas válidas

A partir das métricas coletadas, foi possível fazer uma análise paralela sobre os perfis de usuários que identificaram falhas. Considerando o total de falhas identificadas, a Figura 3 mostra que os Analistas de Testes identificaram a maioria das falhas (40%), o que já era esperado, por terem conhecimento técnico sobre teste de software. Porém, os universitários (29%) e desenvolvedores (24%) também contribuíram de forma

significativa com a qualidade do produto.

A Figura 4 mostra o percentual de falhas identificadas unicamente por um determinado perfil. Desta forma, pode-se observar que o perfil Analistas de Teste foi novamente o que mais contribuiu na identificação de falhas (42%), evidenciando que a experiência em testes de software é importante. Os perfis Desenvolvedores (27%) e Universitários (26%) juntos também contribuíram de forma significativa com a qualidade do produto, sendo que, neste cenário, desenvolvedores identificaram mais falhas que os universitários, diferentemente do cenário da Figura 3. Por fim, o perfil Outras Áreas teve uma contribuição menor (5%) e os Alunos dos CTIs não tiveram contribuição, visto que todas as falhas identificadas por eles também foram identificadas por outros perfis.

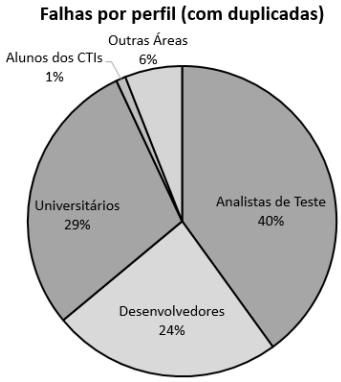


Figura 3: Falhas por perfil (com duplicadas)

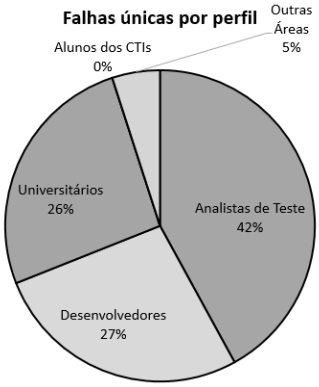
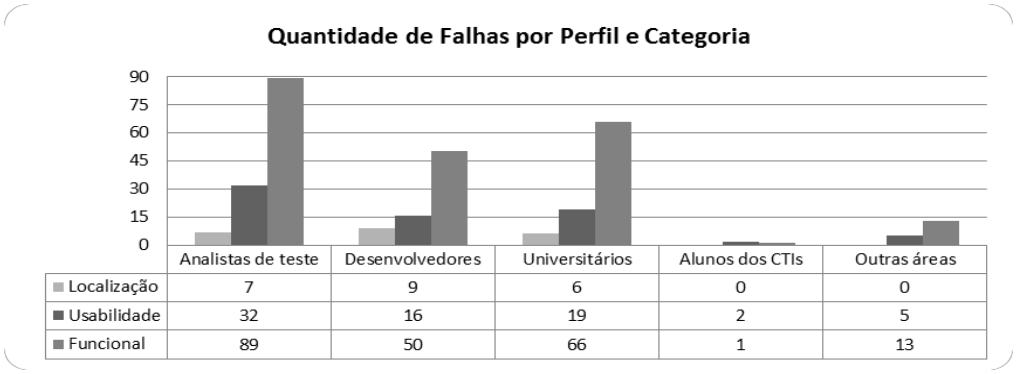


Figura 4: Falhas únicas por perfil

Ao observar a Figura 5, é possível verificar que a quantidade de falhas identificadas em aspectos funcionais do software foi maior, seguida pelas falhas em aspectos de usabilidade. Os seguintes critérios foram considerados para categorizar:

- **Usabilidade:** falhas referentes à falta de clareza e dificuldade de entendimento das funcionalidades e problemas na navegação;
- **Funcional:** falhas referentes a erros nas funcionalidades, comportamento diferente do esperado;
- **Localização:** falhas referentes aos erros de tradução e texto sem tradução.



	Analistas de teste	Desenvolvedores	Universitários	Alunos dos CTIs	Outras áreas
Localização	7	9	6	0	0
Usabilidade	32	16	19	2	5
Funcional	89	50	66	1	13

Figura 5: Quantidade de falhas por perfil e categoria

5 Conclusão e Trabalhos Futuros

Esse artigo apresentou um relato de experiência sobre a realização de testes de softwares, num produto em que o cliente tinha como principal objetivo ter um feedback sobre a aceitação desse produto no mercado brasileiro. Dessa forma, os testes foram

realizados com diferentes perfis de usuários, envolvendo profissionais da área de computação e usuários leigos, distribuídos geograficamente nas cinco regiões do Brasil.

Tendo em vista esse cenário, foi observado que a colaboração de diferentes perfis testando o mesmo produto foi importante, uma vez que cada pessoa testou sob o aspecto atrelado ao seu perfil, tal fato ficou evidente através das falhas reportadas. Um exemplo disso é que um desenvolvedor tentou fazer injeção de código SQL nos campos de entrada, os Analistas de Testes se preocuparam em validar as funcionalidades, os usuários de outras áreas se preocupavam em concluir uma operação desejada, e assim por diante.

Conforme pode ser observado na Figura 5, nem todos os perfis envolvidos nos testes contribuíram de forma significativa com a qualidade do produto. Dessa forma, pode-se concluir que não seria necessário envolver todos os perfis, e sim, somente perfis testadores, universitários e desenvolvedores, e o resultado alcançado seria o mesmo.

Além do mencionado acima, outros pontos foram considerados de sucesso, como por exemplo, o estabelecimento de uma estratégia de comunicação e padrão de registro de falhas, uma vez que existiam envolvidos espalhados pelas cinco regiões do Brasil. Essas estratégias foram fundamentais para o gerenciamento das atividades diárias.

Como trabalho futuro, para projetos similares, pretende-se aplicar avaliações de usabilidade em ambiente controlado a fim de avaliar a eficácia da interação humano computador mediante a realização de ações por parte dos usuários; avaliar a satisfação ou insatisfação dos usuários com o produto, analisando face aos recursos empregados como tempo, quantidade de incidentes, busca de ajuda dentre outros. Outro ponto que pode ser melhorado é a escolha dos perfis de usuários, uma vez que os usuários leigos demonstraram bastante dificuldade em manusear os dispositivos, o que demandou um esforço extra para contextualizá-los de como funciona um dispositivo móvel.

Referências

AUDY, J., PRIKLADNICK, R. Desenvolvimento Distribuído de Software - Desenvolvimento de Software com Equipes Distribuídas. Série Campus. Elsevier Editora, Rio de Janeiro, 2007.

BACH, J. Session-based test management. Software Testing and Quality Engineering, v.2, n.6, 2000.

COLLINS, E.; MACEDO, G.; MAIA, N.; DIAS-NETO, A. C. *An Industrial Experience on the Application of Distributed Testing in a Agile Software Development Environment*. 7th IEEE International Conference on Global Software Engineering. Aug 30, 2012.

LYNDSAY, J.; N. van E. Adventures in session-based testing. Workroom Productions Ltd. Maio 27, 2003.

MYERS, Glenford J.; SANDLER, Corey; BADGETT, Tom. The art of software testing. John Wiley & Sons, 2011.

PEZZÈ, M.; YOUNG, M.. Teste e Análise de Software: processos, princípios e técnicas. Porto Alegre: Bookman, 2008.

PRESSMAN, R. S. *Engenharia de Software: Uma Abordagem Profissional*. 7a Ed. São Paulo: McGraw-Hill, 2011.

SOMMERVILLE, I. *Engenharia de Software*. 8a Ed. São Paulo: Pearson Addison Wesley, 2007.

WHITTAKER, J. A. (2009) Exploratory Software Testing: Tips, Tricks, Tours, and Techniques to Guide Test Design. Addison-Wesley Professional, 2009.