

## Definição e Aplicação de um Processo de Testes Ágeis: um Relato de Experiência

Ismayle S. Santos<sup>1,2,a</sup>, Wellington Franco<sup>1</sup>, Bruno Sabóia<sup>1</sup>, Rossana M.C. Andrade<sup>1,2,b</sup>

<sup>1</sup>Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)

<sup>2</sup>Mestrado e Doutorado em Ciência da Computação (MDCC)

Universidade Federal do Ceará (UFC)

Fortaleza– Ceará – Brasil

{ismaylesantos,wellingtonfranco,bruno,rossana}@great.ufc.br

**Resumo.** *Por promover uma entrega rápida de código útil e gerenciar mudanças de prioridades, optou-se por usar o Scrum em vários projetos de pesquisa e desenvolvimento realizados no nosso laboratório de pesquisa em parceria com uma empresa fabricante de celular. Este artigo define um processo de testes ágeis com base nas experiências de usar o Scrum nesses projetos e relata a aplicação deste processo no desenvolvimento de sistemas móveis. Também são descritas as lições aprendidas no uso do processo e que resultaram na melhoria do mesmo, os benefícios alcançados e os resultados de um survey aplicado com a equipe.*

**Abstract.** *Due to Scrum benefits, such as rapid delivery of code, we choose this framework to manage several research and development projects developed in our research lab in partnership with a mobile phone manufacturer. This paper defines an agile testing process based on the experiences of using Scrum in these projects and reports the application of this process in the development of mobile systems. We also describe lessons learned in the use of the process and that have resulted on its improvement as well as the benefits achieved and the results of a survey applied with the team.*

### 1. Introdução

A aplicação de métodos ágeis tem se tornado cada vez mais comum no desenvolvimento de software. Segundo os resultados de um *survey* nacional sobre métodos ágeis [Melo et al. 2012], respondido por 466 participantes, as principais razões que motivam o uso de métodos ágeis são: o aumento de produtividade, o gerenciamento de mudanças de prioridade e o aumento da qualidade de software.

Dentre os métodos ágeis, o *Scrum* [Schwaber e Sutherland 2013] é um dos métodos mais usados pela comunidade brasileira [Melo et al. 2012]. O *Scrum* trabalha com *Sprints*, que são iterações de trabalho de tamanho fixo, e com a definição de três papéis [Schwaber e Sutherland 2013]: i) *Product Owner*, que é o responsável por maximizar o valor do produto; ii) *Time de desenvolvimento*, composto pelos

---

<sup>a</sup> Bolsista de Doutorado da CAPES

<sup>b</sup> Bolsista do CNPq de Produtividade em Desenvolvimento Tecnológico e Extensão Inovadora (DT) 2

profissionais que fazem o trabalho para entregar as versões incrementais do produto no final de cada *Sprint*; e iii) *Scrum Master*, responsável por garantir que o *Scrum* é entendido e que o time de desenvolvimento siga as práticas e regras do *Scrum*.

Além disso, no *Scrum* os testes (agora também ágeis) deixam de ser uma fase e passam a atuar de maneira integrada com os esforços de desenvolvimento sendo, inclusive, obrigatórios para que um item seja considerado “pronto” [Myers *et al.* 2011].

O GREat<sup>c</sup> executa projetos de pesquisa, desenvolvimento e inovação<sup>d</sup>, em parceria com empresas fabricantes de celular, os quais usam o *Scrum* no desenvolvimento de aplicações móveis. Em um desses projetos, em particular, dado restrições de tempo e ausência de pessoas especializadas, não havia um processo de testes bem definido. Desta forma, fomos motivados a observar a aplicação de testes ágeis com o *Scrum* no desenvolvimento de aplicações móveis neste projeto para definir um processo de testes. Este relato tem como objetivo apresentar este processo, que foi definido por meio das nossas experiências. Também são descritas neste artigo: (i) a percepção da equipe, coletada por meio de um *survey*, quanto ao processo definido; (ii) os benefícios alcançados; e (iii) as lições aprendidas durante o período da observação.

O restante deste artigo está organizado em outras cinco seções. Inicialmente, são apresentados os trabalhos relacionados na Seção 2. A seguir, na Seção 3, o relato da experiência de aplicação de testes ágeis, o processo proposto e os resultados do *survey* aplicado com a equipe são descritos. Na Seção 4 são apresentadas as lições aprendidas. Por fim, na Seção 5 são apresentadas as conclusões e direcionamentos para trabalhos futuros.

## 2. Trabalhos Relacionados

Na literatura existem trabalhos relacionados à aplicação de testes ágeis [Lima e Travassos 2013, Ribeiro *et al.* 2013, Collins e Lucena Jr. 2012, Filho *et al.* 2012, Santos *et al.* 2011], os quais são descritos sucintamente a seguir.

Collins e Lucena Jr. (2012) descrevem a experiência com o uso de ferramentas de teste *open source* na automação de testes em dois projetos de desenvolvimento utilizando o *Scrum*. Já Lima e Travassos (2013) apresentam o relato da experiência na introdução de práticas ágeis em processos de testes de software em uma organização com modelo de maturidade. Santos *et al.* (2011), por sua vez, descrevem observações feitas no uso de práticas de testes em ambientes ágeis utilizando o *Scrum*. Baseado nestas observações, Santos *et al.* apresentam as dificuldades e decisões tomadas. Por fim, os trabalhos de Ribeiro *et al.* (2013) e de Filho *et al.* (2012) avaliam os benefícios do TDD (*Test Driven Development*), uma prática comum em ambientes ágeis.

Ao contrário dos trabalhos supracitados, este relato não foca na automação dos testes ou em práticas de testes específicas. O diferencial do relato apresentado neste artigo está na descrição das experiências ao longo de mais de um ano que resultaram na definição de um processo de testes ágeis. Além disso, são descritos as lições aprendidas e os resultados de um *survey* aplicado para coletar o *feedback* da equipe.

---

<sup>c</sup> Grupo de Redes de Computadores, Engenharia de Software e Sistemas. <http://great.ufc.br>

<sup>d</sup> Por meio da Lei da Informática (Leis 8.248/91, 10.176/01 e 11.077/04), da Fundação Cearense de Pesquisa e Cultura (FCPC - <http://www.fcpc.ufc.br/>) e do MCTI (<http://www.mcti.gov.br/>)

### 3. Experiência no uso de testes ágeis no desenvolvimento de software móvel

O objetivo deste artigo é apresentar um relato de experiência da aplicação de testes ágeis no desenvolvimento de três sistemas móveis. Sendo assim, quanto à abordagem metodológica utilizada pode-se classificar esta pesquisa como descritiva [Leite 2008], pois a finalidade é observar, registrar e analisar as experiências do uso de testes ágeis.

Quanto ao procedimento metodológico, esta pesquisa pode ser classificada como quantitativa e qualitativa [Leite 2008]. Essa pesquisa é quantitativa, pois utilizamos um *survey* com questões fechadas e também mensuramos a quantidade de testes de sistemas executados e de testes que falharam. Por outro lado, ela tem caráter qualitativo, pois objetiva descrever a experiência relacionada à aplicação de testes ágeis.

#### 3.1. Caracterização dos projetos e da equipe envolvida

O período deste relato é de julho de 2013 a dezembro de 2014, no qual foram desenvolvidos três sistemas de informação móveis para uma empresa privada parceira do GREat. Tais aplicativos são denominados neste artigo de *SIM1*, *SIM2* e *SIM3* por razões de confidencialidade.

Durante o desenvolvimento destes aplicativos, as *Sprints* tinham tamanho fixo de 2 semanas. Na Tabela 1 é apresentado o período de desenvolvimento de cada aplicativo, bem como o número de pessoas relacionadas com cada papel do *Scrum*.

**Tabela 1. Caracterização dos aplicativos da experiência**

Aplicativo	Início	Fim	Time	Scrum Master	Product Owner
<i>SIM1</i>	Julho/2013	Mai/2014	08	1	1
<i>SIM2</i>	Julho/2014	Set/2014	12	1	1
<i>SIM3</i>	Set/2014	Dez/2014	10	1	1

O tamanho do time variou durante o período deste relato, tendo uma média de 10 pessoas. Com relação ao papel do *Scrum Master*, este era executado pelo gerente de projetos com o apoio do membro mais experiente da equipe, que auxiliava o gerente no acompanhamento e na remoção de impedimentos da equipe. Por fim, o *Product Owner* era o próprio cliente com o qual o time se reunia de 15 em 15 dias para revisão da *Sprint* corrente e planejamento da próxima *Sprint*. Quando o cliente não estava disponível para participar de alguma reunião, o time fazia um planejamento prévio, o qual era confirmado posteriormente com o cliente.

#### 3.2. Testes no desenvolvimento do *SIM1*

Durante o desenvolvimento do *SIM1*, os desenvolvedores criavam testes unitários sem qualquer planejamento prévio e alguns testes de sistema eram implementados com a ferramenta Robotium<sup>e</sup>, enquanto que os *bugs* (defeitos) eram registrados no Jira<sup>f</sup>.

Observou-se neste período a ausência de um processo de testes bem definido. A falta de planejamento prévio e de documentação dos testes dificultava a avaliação do que estava sendo testado e de como estava sendo testado.

<sup>e</sup> <https://code.google.com/p/robotium/>

<sup>f</sup> <https://www.atlassian.com/software/jira>

Para melhorar a qualidade dos testes, em janeiro de 2014, foi contratado um analista de testes com mais de quatro anos de experiência no teste de aplicações móveis. Este analista executou testes de sistemas ao final de cada *Sprint* e identificou 148 *bugs* no *SIMI* (47% dos testes de sistema falharam), conforme apresentado na Figura 1. Cabe ressaltar que muitos destes bugs eram do tipo *critical*, que são *bugs* que impedem o uso do aplicativo. Como resultado dos dados e observações, chegou-se à conclusão de que a aplicação dos testes de forma *ad-hoc* (sem seguir um processo de testes bem definido) era ineficiente para o contexto do projeto.

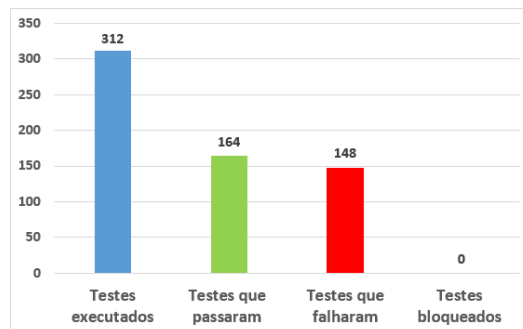


Figura 1. Resultados dos testes de sistema no *SIMI*

### 3.3. Definição do Processo e Testes no desenvolvimento do *SIM2*

Com o intuito de minimizar os problemas identificados, foi definido um processo de testes adaptado às diferentes etapas do *Scrum* (ver Figura 2). Ressalta-se que os testes de sistemas eram feitos pelo analista de testes quando a funcionalidade é finalizada e ao fim de cada *Sprint*.

Além do processo, algumas práticas foram adotadas. Em primeiro lugar, o analista de testes passou a acompanhar de perto a implementação dos testes unitários e de integração junto com os desenvolvedores. Para isso, ele especificava os cenários de testes, os quais os desenvolvedores usavam como base para a implementação dos testes de unidade e de integração. O analista de testes também executou “testes em par”, que é a prática da *Programação em Par* aplicada na implementação dos testes, com cada desenvolvedor até que eles estivessem melhor preparados para criar bons testes.

Além disso, cada atividade da *Sprint Backlog* tinha agora uma validação do analista de testes para ficar como *done* (concluída). Quando a atividade não gerava um código executável, era feita uma revisão do código da aplicação. Por outro lado, se fosse possível executar o código, então testes manuais eram conduzidos para validar a funcionalidade implementada na atividade. Destaca-se que não foram implementados testes automatizados devidas as características do aplicativo *SIM2*. Também foram incluídas atividades específicas para a criação dos testes unitários, enfatizando a importância dos mesmos dentro da *Sprint*.

Adicionalmente, para maximizar a transparência das atividades sendo realizadas, foi instalada uma TV de 42” na sala de desenvolvimento. Por meio do uso desta, todos da equipe puderam acompanhar em tempo real as atividades por meio do Jira.

Por fim, os cenários de testes passaram a ser documentados em “documentos vivos”, documentos que eram sempre atualizados pela equipe e compartilhados entre

todos os envolvidos. Para ser um documento enxuto, optou-se por planilhas eletrônicas com apenas três colunas: cenário, passo-a-passo e resultado. Dessa forma, o time considerou fácil documentar, planejar e priorizar os testes conforme as mudanças nas *Sprints* e solicitações do cliente. Cabe ressaltar que durante o *SIM2*, os *bugs* eram registrados também nesta planilha, mas foi observado que isso não era uma boa prática porque dificultava o acompanhamento da resolução dos defeitos. Na Figura 3 são apresentados os resultados dos testes no *SIM2*.

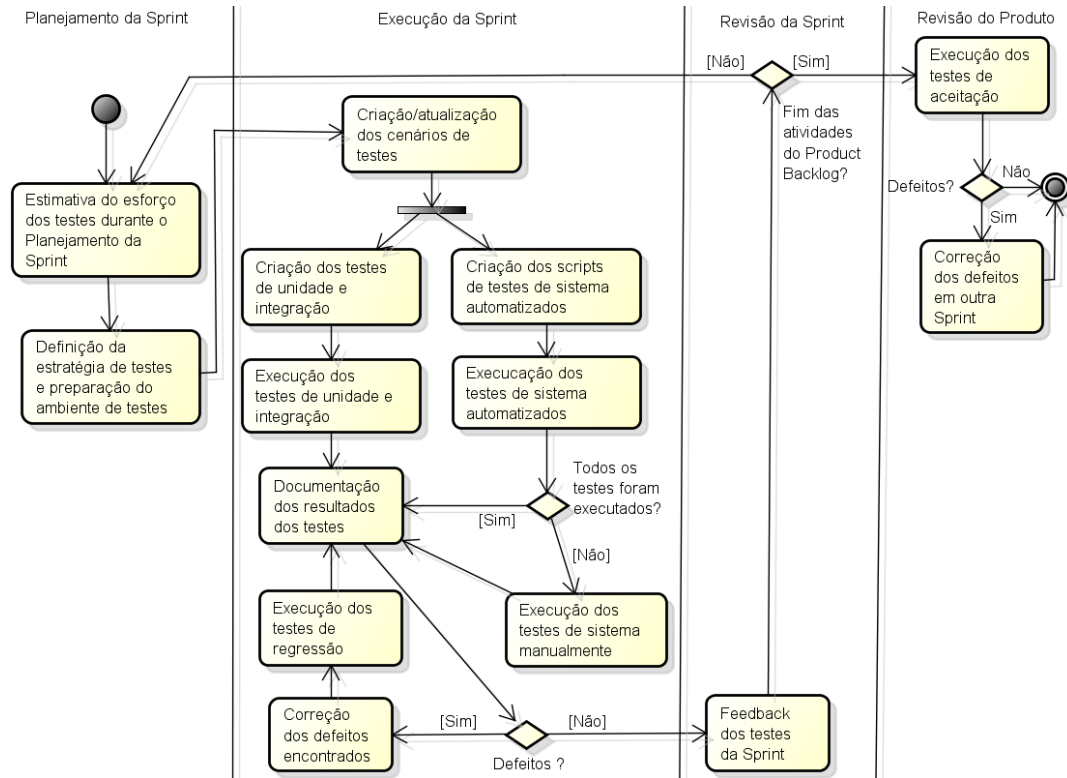


Figura 2. Processo de testes definido

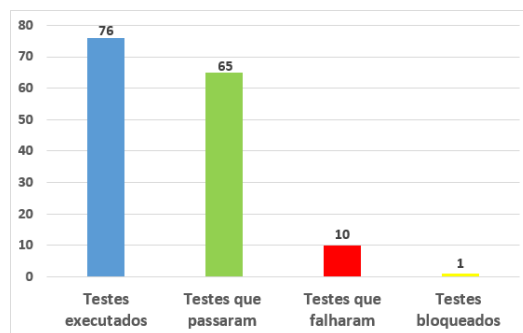


Figura 3. Resultados dos testes de sistema no SIM2

O resultado direto das mudanças aplicadas durante o desenvolvimento do *SIM2* foi: (i) maior participação de todos os desenvolvedores na qualidade do produto; (ii) maior confiança dos desenvolvedores nos testes implementados; e (iii) melhor monitoramento do que estava sendo testado por meio do uso da TV.

### 3.4. Testes no desenvolvimento do *SIM3* e Aplicação do *Survey*

Dado os resultados positivos, o processo e as práticas aplicadas no *SIM2* foram usadas durante o desenvolvimento do *SIM3*. Contudo, a reunião de retrospectiva do *SIM2* revelou a necessidade de melhorias no processo. O time observou que era necessário que o tempo de validação fosse levado em conta nas estimativas de cada atividade. Sendo assim, para cada atividade, 1 ou 0.5 pontos passaram a ser acrescentados na estimativa da atividade para a validação da mesma. Também foi colocado na *Sprint Backlog* uma atividade para o teste de sistema do que foi desenvolvido na *Sprint*.

Outro diferencial é que durante o desenvolvimento do *SIM3* vários testes foram automatizados com o Robotium, o que facilitou a execução de testes de regressão. Por fim, os *bugs* voltaram a ser registrados no Jira em vez de usar a planilha citada na Seção 3.3, e dessa forma foi possível acompanhar a resolução dos mesmos, pois para cada *bug* era aberto uma *issue* no Jira de correção. Na Figura 4 são apresentados os resultados dos testes no *SIM3*, onde 28% dos testes de sistema executados falharam. Cabe ressaltar que nenhum dos *bugs* encontrados foi do tipo *critical*, sendo isso uma possível indicação de que houve uma melhora dos testes feitos nos estágios iniciais de desenvolvimento.

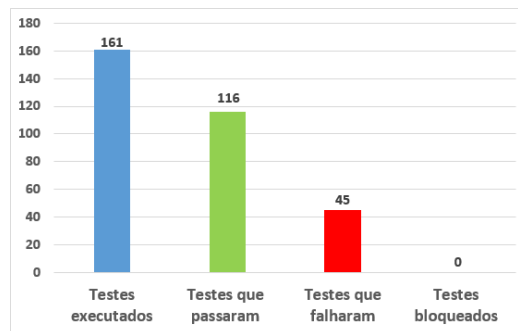


Figura 4. Resultados dos testes de sistema no *SIM3*

Para uma avaliação inicial do processo de testes utilizado, um *survey* foi aplicado (ver resultados na Tabela 2) com o objetivo de captar a percepção da equipe quanto ao mesmo no que se refere aos valores ágeis. O *feedback* da equipe foi positivo, mas destacou-se a necessidade de enfatizar uma melhor interação entre os envolvidos.

Tabela 2. Resultado do *survey* realizado com a equipe

Questão	CT	C	N	D	DT
Os testadores interagem diretamente com o produto	6	3			
Os testadores interagem diretamente com os desenvolvedores	4	5			
O foco está na interação dos envolvidos e não em processos e ferramentas	1	4	1	1	2
Os testadores têm um feedback interativo com o cliente	1	5	2	1	
Os testes são repriorizados conforme a necessidade	1	7	1		
A automação dos testes está adequada as necessidades da equipe	2	4	3		
Você se sente satisfeito com relação as ferramentas de testes usadas	1	6	2		

**Legenda:** CT=Concordo Totalmente; C=Concordo; N=Neutro; D=Discordo; DT=Discordo Totalmente

Por fim, observamos que com as mudanças aplicadas durante o desenvolvimento do *SIM3*, o time de desenvolvimento conseguiu um fluxo leve e efetivo para a realização de testes ágeis. Os principais benefícios observados foram: (i) maior qualidade, visto que nenhum *bug* do tipo *critical* foi encontrado nos testes de sistema; (ii) maior sensação de responsabilidade da qualidade pelos desenvolvedores; e (iii) maior integração entre o código e o teste, uma vez que o teste se tornou parte integrante do desenvolvimento de uma funcionalidade.

#### 4. Análise dos resultados e lições aprendidas

Com base na experiência prática de testes ágeis com o *Scrum* no desenvolvimento de três aplicações móveis em um projeto em parceria com uma empresa fabricante de celular, as seguintes lições foram aprendidas:

- *Sem papéis especializados, mas com um processo de testes e pessoas especializadas.* Durante os testes do *SIM1*, muitos dos defeitos encontrados eram do tipo *critical*. Isso indicou que a imaturidade dos desenvolvedores quanto aos testes unitários e de integração propiciou a identificação tardia de muitos defeitos críticos. Já no desenvolvimento do *SIM2* e do *SIM3*, a presença do analista de testes e o uso de um processo de testes bem definido possibilitou que defeitos críticos fossem identificados em estágios iniciais do desenvolvimento;
- *Documentação de casos de testes é essencial.* A experiência no desenvolvimento do *SIM1* revelou indícios de que a falta de uma documentação dos casos de testes dificulta o acompanhamento e a avaliação da qualidade dos testes;
- *Os defeitos precisam ser acompanhados.* Durante o desenvolvimento do *SIM2*, pode-se notar que o registro dos defeitos em uma planilha eletrônica dificultou o acompanhamento da correção dos defeitos. Logo, o uso do Jira durante o *SIM3* foi importante para o melhor acompanhamento dos defeitos;
- *Nem sempre é possível automatizar, mas é aconselhável.* A automação dos testes é interessante porque facilita o *feedback* contínuo. O uso planejado de uma ferramenta de automação de testes de sistemas, durante o desenvolvimento do *SIM3*, favoreceu a identificação rápida de defeitos;
- *Testes em par.* A aplicação da programação em par para implementar testes durante o desenvolvimento *SIM2* teve uma repercussão muito positiva, pois além de capacitar os desenvolvedores, melhorou a comunicação entre o time; e
- *O cliente participa dos testes.* Quando o cliente não puder participar de uma decisão em determinado momento, é uma boa prática documentar as decisões tomadas relativas à atividade de testes para posteriormente confirmar com cliente. Isso foi feito nos testes dos três aplicativos mantendo assim uma constante interação com o cliente.

#### 5. Conclusões

Este artigo apresentou um relato de experiência da aplicação de testes ágeis com o *Scrum* no desenvolvimento de três sistemas móveis (*SIM1*, *SIM2* e *SIM3*) e a definição de um processo de testes ágeis.

Durante o desenvolvimento do *SIM1* observou-se que a falta de planejamento, de experiência dos desenvolvedores quanto aos testes de unidade e de um processo de testes bem definido foram motivos para a alta incidência de defeitos. Dessa forma, definiu-se um processo de testes ágeis e adotou-se algumas práticas, como o uso de “testes em par”, durante o desenvolvimento do *SIM2* resultando em melhorias da qualidade dos testes. Contudo, percebeu-se a necessidade de mudanças nas práticas, como o registro dos *bugs* no Jira, que foram aplicadas durante o desenvolvimento do *SIM3*. O resultado final foi um fluxo leve e efetivo para a realização de testes ágeis. Este processo teve um *feedback* positivo da equipe e trouxe benefícios como um maior envolvimento de todos na qualidade do produto. Foram também apresentadas neste artigo as lições aprendidas que resultaram na melhoria deste processo de teste.

Como trabalhos futuros seria interessante fazer entrevistas com a equipe para obter mais informações a fim de ajudar na melhoria do processo de testes definido, bem como coletar medidas (e.g. cobertura de código) para avaliar os testes criados.

## Referências

- Collins, E. F.; de Lucena, V. F. (2012) Software Test Automation practices in agile development environment: An industry experience report. In *Proceedings of the 7th International Workshop on Automation of Software Test (AST)*, pp. 57-63.
- Filho, M. C.; Vasconcelo, J. L.; Santos, W. B.; Silva, I. F. (2012) Um Estudo de Caso sobre o Aumento de Qualidade de Software em Projetos de Sistemas de Informação que Utilizam *Test Driven Development*. Anais do VIII Simpósio Brasileiro de Sistemas de Informação (SBSI).
- Lima, C. G. B; Travassos, G. H. (2013). Inserção de Agilidade nos Processos de Teste de uma Organização de Software MPS.BR: Um relato de experiência. Anais do Simpósio Brasileiro de Qualidade de Software, pp. 270-277.
- Leite, F. T. (2012) Metodologia Científica: métodos e técnicas de pesquisa: monografias, dissertações, teses e livros. Ideias & Letras, 3<sup>o</sup> edição.
- Melo, C. O.; Santos, V. A.; Corbucci, H.; Katayama, E.; Goldman, A.; Kon, F. (2012) Métodos ágeis no Brasil: estado da prática em times e organizações. Relatório Técnico RT-MAC-2012-03. Departamento de Ciência da Computação. IME-USP.
- Myers, G. J.; Badgett, T.; Sandler, C. (2011) *The Art of Software Testing*. John Wiley & Sons, 3<sup>o</sup> edição.
- Ribeiro, S; Albuquerque, A.; Filho, L; Júnior, L.; Régia, S; Cavalcante, S. (2013). Uma avaliação da abordagem TDD (*Test Driven Development*) em uma empresa desenvolvedora de software madura. Anais do Simpósio Brasileiro de Qualidade de Software, pp 302-309
- Santos, A. M.; Karlsson, B. F.; Cavalcante, A. M.; Correia, I. B.; Silva, E. (2011) Testing in an agile product development environment: An industry experience report. In *Proceedings of the 12th Latin American Test Workshop*, pp. 1-6.
- Schwaber, K.; Sutherland, J. (2013) *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game*. Disponível em: <http://www.Scrumguides.org/docs/Scrumguide/v1/Scrum-guide-us.pdf> Último acesso em janeiro de 2015.