ADOPTING KEYWORD-DRIVEN TESTING FRAMEWORK IN WEB-BASED PRODUCT DEVELOPMENT

MUHAMMAD AZAM ALIAS

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Software Engineering

Advanced Informatics School
Universiti Teknologi Malaysia

JANUARY 2015

ABSTRACT

Web applications are abundant in the World Wide Web as well as in the private networks. Organisations resort to using Agile Scrum development to shorten their time to deliver their products rapidly. In less mature Agile organisations, testing is typically done in manual way, with lack of test automation practices at unit, integration, and acceptance levels. Lack of documentations for expected behaviours of existing features is also prevalent. These practices cause long delays in testing for new features and regression testing for existing features. Manual testing consumes a lot of time and may cause some tests to be skipped under time constraints. Undocumented existing features also require back and forth discussions among team members to determine the correct behaviour of the product. We propose to adopt the usage of Robot Framework, a Keyword-Driven Testing (KDT) tool along with Selenium2Library. The framework is to be integrated with existing Jenkins Continuous Integration tool. Leveraging Keyword-Driven Testing framework into Continuous Integration architecture provides a maintainable, reusable, understandable test assets that can be run automatically every time a new build of the product is deployed to multiple target environments. This integrated automated infrastructure allows regression testing to be executed rapidly, without sacrificing test coverage. Rapid feedback from the tests and self-documenting test assets will further improve the productivity and efficiency and ultimately, the quality of the delivered product.

ABSTRAK

Aplikasi web banyak terdapat dalam World Wide Web serta dalam rangkaian persendirian. Organisasi menggunakan pembangunan Agile Scrum untuk memendekkan masa untuk menghasilkan produk dengan cepat. Dalam organisasi Agile yang kurang berpengalaman, pengujian biasanya dilakukan secara manual, dengan kekurangan amalan automasi ujian pada tahap unit, integrasi, dan penerimaan. Kekurangan dokumentasi yang menentukan spesifikasi sedia ada untuk sesuatu aplikasi juga menjadi satu kebiasaan. Ini mengakibatkan pengujian mengalami penangguhan dan kelewatan semasa pengujian ciri-ciri baru dan juga ujian regresi ciri-ciri sedia ada. Pengujian secara manual menggunakan banyak masa, menyebabkan beberapa ujian dilangkau akibat kekangan masa. Ciri sedia ada yang tidak mempunyai dokumen spesifikasi juga memerlukan perbincangan berterusan di kalangan kumpulan pembangun untuk menentukan spesifikasi produk yang sepatutnya, sewaktu pengubahan atau penambahan ciri-ciri baru. Kami mencadangkan untuk menggunakan Robot Framework, iaitu satu perisian Keyword-Driven Testing (KDT) bersama-sama dengan Selenium2Library. Perisian ini disepadukan dengan infrastruktur integrasi berterusan Jenkins yang sedia ada dalam organisasi. Menggunakan teknik Keyword-Driven Testing untuk pengujian bersama dengan infrastruktur integrasi berterusan, aset pengujian perisian boleh digunakan semula, mudah difahami dan boleh dijalankan secara automatik untuk pelbagai situasi setiap kali produk diubah. Infrastruktur automatik bersepadu membolehkan ujian regresi dilaksanakan dengan pantas, tanpa melangkau mana-mana ujian akibat kesuntukan masa. Maklum balas yang pantas daripada ujian dan aset ujian yang mudah difahami akan meningkatkan lagi produktiviti dan kualiti produk yang dihasilkan.

TABLE OF CONTENTS

CHAPTER	TITLE		PAGE
	DECLARATION		ii
	ABSTRACT		iii
	ABSTRAK		iv
	TABLE OF CONTENTS		v
	LIST OF TABLES		ix
	LIST OF FIGURES		x
	LIST OF ABBREVIATIONS		xi
	LIST OF APPENDICES		xii
1	**INTRODUCTION**		**1**
	1.1	Company Background	1
	1.2	Project Background	3
	1.3	Motivation of Project	4
	1.4	Project Objectives	5
	1.5	Project Scopes	5
	1.6	Project Deliverables	6
	1.7	Project Planning	6
2	**LITERATURE REVIEW**		**7**
	2.1	Introduction to Software Testing	7
	2.2	Types and Levels of Testing	8
	2.3	Introduction to Agile	9
	2.4	Web-Based Product Development Using Agile	12
	2.5	Keyword-Driven Testing	15
	2.6	Integration with CI Infrastructure	16
	2.7	Comparative Analysis of Current Practices	17
	2.8	Conclusion	20
3	**PROJECT METHODOLOGY**		**21**

3.1 Introduction 21
3.2 Project Methodology 21
3.2.1 Planning 23
3.2.2 Literature Review 23
3.2.3 Gap Analysis 23
3.2.4 Implementation Phase 1 24
3.2.5 Implementation Phase 2 24
3.2.6 Evaluation 24
3.2.7 Reporting 24

4 ADOPTION OF RF IN WEB-BASED PRODUCT DEVELOPMENT 25
4.1 Overview of Current Practice 25
4.2 Other Available Technologies 31
4.2.1 Selenium WebDriver 32
4.2.2 Selenium WebDriver with Page Object Pattern 33
4.2.3 RF with S2L 36
4.3 Proposed Framework 40
4.4 Gap Analysis of Solutions 42

5 PROJECT DISCUSSION - PHASE 1 45
5.1 Installation of RF and its Dependencies 45
5.2 Installation of S2L onto RF 46
5.3 Installation of Jenkins and its Integration with RF 46
5.4 Installation and Configuration of Git Client and GitHub 47
5.5 Creation of Custom Keywords and Test Assets in RF Format 48
5.6 Creation of Test Cases Using Keywords 49
5.7 Notes, Challenges, and Observations for Phase 1 50

6 PROJECT DISCUSSION - PHASE 2 51
6.1 Integration of Jenkins with Git plugin 51
6.2 Variable File Implemented as Python Module 52
6.3 Tagging of Test Cases 54
6.4 Pybot Script in Context of Jenkins 55
6.5 Custom Keywords Implementation 61

7 EVALUATION **65**
7.1 Setup of KDT Framework for Test Status Visibility 65
7.2 Integration of Framework with Existing System for Rapid Test Execution 65
7.3 Development of Reusable Test Assets for Better Maintainability 66
7.4 Creation of Self-Documenting Test Cases 66
7.5 Evaluation Summary 66

8 CONCLUSION **67**
8.1 Benefits Attained from Applying KDT with RF into the Infrastructure 67
8.1.1 Reusable Test Assets across Multiple Configurations 67
8.1.2 Self-Documenting Test Cases for Debugging and Reference 68
8.1.3 Unattended Automated Test Executions 68
8.1.4 Rapid Feedbacks in Forms of Accessible Test Results via Web Interface 68
8.1.5 Detailed Execution Logs with Screenshots of Each Failing Test Steps 69
8.1.6 Selection of Test Cases Execution Rule Simplified by Tags 69
8.1.7 End-to-End Integration Using GitHub, Jenkins, RF and S2L 69
8.2 Limitations 70
8.2.1 Inability to Use InternetExplorerDriver 70
8.2.2 Absence of Time Measures 70
8.2.3 False Positives 70
8.2.4 Broken Compatibility with Certain Components 71
8.3 Areas of Improvements and Future Works 72
8.3.1 Parallelisation of Test Execution 72
8.3.2 Flattening of Test Result Files 72
8.3.3 Time Measurement of Test Activities 73
8.3.4 Moving of Logic and Processing into External Libraries 73
8.3.5 Usage of KDT and RF in Non-HTML Products 73

8.3.6 Investigation on the Effects of Web Locators in S2L Context 74
8.3.7 Strategies to Deal with Ajax-heavy Pages using RF & S2L 74

REFERENCES **75**
Appendices A – C 81 – 88

LIST OF TABLES

TABLE NO.	TITLE	PAGE
1.1	Project Plan	6
2.1	Criteria and Values for Solutions Analysis	18
2.2	Analysis of Solutions Based on Criteria	19
3.1	Project Stages, Activities and Deliverables	22
4.1	Summary of Gap Analysis of Possible Solutions	42
6.1	Pybot Batch Script Configurations for Multiple Jenkins Builds	60

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
2.1	Breakdown of Topics by Conferences	10
2.2	Simplified Architecture of a CI Infrastructure	11
4.1	Current Infrastructure of SDLC in iProperty.com SG IT team	26
4.2	Selenium IDE Script Example	28
4.3	Selenium IDE Test Script in HTML Format	29
4.4	Test Log for Selenium IDE	30
4.5	Selenium WebDriver Script in Python	32
4.6	Selenium WebDriver Script with Page Object-Like Definition	33
4.7	Reusing Function from Page-Object Definition	34
4.8	Selenium WebDriver Execution Log and Error Message	35
4.9	RF Test Case using S2L	36
4.10	Overall Test Execution Log for RF in HTML	38
4.11	Highlighted Failing Test Steps from HTML Result	39
4.12	Proposed Framework of using RF with S2L	40
6.1	Git Plugin for Jenkins	51
6.2	Variables Implemented as Python Module	52
6.3	Self-Testing Environment Variable Module in RF	53
6.4	Setting BaseURL Variable Globally Per-Execution	53
6.5	Tags in Test Cases	54
6.6	Pybot and Rebot Scripts for a Jenkins Build	56
6.7	Pybot Script Setup in Jenkins RF Plugin	57
6.8	Jenkins RF Plugin Setup for Publishing Test Results	58
6.9	Test Execution Results Trend on Jenkins Dashboard	59
6.10	Automatic Trigger of Jenkins Build	60
6.11	Low Level Keywords	62
6.12	Low Level Keywords with Arguments and Return Values	63
6.13	Loop Construct in Keywords	64

LIST OF ABBREVIATIONS

API	-	Application Programming Interface
ATDD	-	Acceptance Test-Driven Development
AUT	-	Application Under Test
CI	-	Continuous Integration
DB	-	Database
CLI	-	Command Line Interface
CSS	-	Cascading Style Sheets
CSV	-	Comma-Separated Value
FF	-	Mozilla Firefox
GC	-	Google Chrome
GUI	-	Graphical User Interface
HTML	-	HyperText Markup Language
IDE	-	Integrated Development Environment / Selenium IDE
IE	-	Microsoft Internet Explorer
IP	-	Internet Protocol
KDT	-	Keyword-Driven Testing
RF	-	Robot Framework
S2L	-	Selenium2Library
SCM	-	Software Configuration Management
SDLC	-	Software Development Life Cycle
SSH	-	Secure Shell
TDD	-	Test-Driven Development
TSV	-	Tab-Separated Value
URL	-	Uniform Resource Locator
XML	-	Extensible Markup Language
XP	-	Extreme Programming

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Resources	81
B	Keywords	83
C	Test Cases	88

CHAPTER 1

INTRODUCTION

1.1 Company Background

The iProperty Group is a business entity in real estate industry. They own a network of various property websites under the iProperty.com brand and operates in the Asia region.

Among the notable websites under the iProperty Group's portfolios are: iProperty.com (Malaysia, Singapore, India, Philippines), Rumah123.com (Indonesia), GoHome.com.hk (Hong Kong) and vproperty.mo (Macau).

Operating since 2003, the Group develop, maintain and acquire various websites to support their vision. There are a few categories of websites and products across the iProperty Group portfolio:

Consumer Portals serve as one-stop centres for mass consumers to:

i. Search and view property listings
ii. Subscribe to property listing alerts
iii. Bookmark and shortlist properties
iv. Get property exhibition and events update
v. Contact property agents
vi. Get loan and financing information
vii. Read tips, guides, reviews and news regarding property sector

iRealtor is a category of websites for property agents to draft, list, and update property listings. This platform also allows the property agents to respond to enquiries made by general consumers. Access to this website is only to paying property agents. The listings uploaded from this website will be accessible to consumers via the Consumer Portal. The number of listings allowed to be uploaded per account depends on the subscription package of the property agent.

iAdmin serves as an internal portal for iProperty's staff and administrators to maintain the property agents accounts, subscription settings, and other maintenance-related tasks for both the Consumer Portal and iRealtor.

Mobile web allows mass consumers to access the core essential features of Consumer Portal via a lightweight mobile web interface. Similar functionalities are also delivered via native mobile applications for popular platforms like Android devices, iPad, iPhone, Windows Phone, Windows 8 Metro, and Nokia Symbian. To cater to the small screen estate, not all of the functions in Consumer Portal are available in the mobile web and native mobile version of the website.

1.2 Project Background

In iProperty, there are separate teams covering different country websites. For iProperty Singapore website, all of the comprising products (Consumer Portal, iRealtor, and iAdmin) are handled by a single Agile team. Although normally seen as separate products, they actually are interfacing with each other. As the author is part of this Agile team for iProperty Singapore products, the focus of this project is based on the context of this team.

For each new feature or bug fix implementation (in form of user stories), they are carefully deployed to various tests and staging environments before being deployed to production environment. In each stage, the user stories are manually tested whether they are correctly implemented, and also checked if they cause other unintended behaviours in the existing product and/or interfacing products.

Sometimes, because of different data in tests, staging and production environments, regression bugs can only be caught in later stages of deployments. When this happens, depending on the cases, the fixes will be applied in stages again and tested accordingly. This in turn increases turnaround time to deliver and deploy the stories to the customers.

Being an Agile team, the documentation is kept to minimum except for big stories or during major site revamps. There is little to no evidence that existing or new user stories are being tested regularly during the software development lifecycle, except for a few descriptions on the Project Tracker. Factoring in rapid development lifecycle, small team and little visibility to the test processes, sometimes it could not be prevented that some of the tests would be skipped due to time constraints. Normally only cursory tests are done for story and regression tests.

There were past efforts on implementing automated test tools like Selenium IDE but it had shown to be not really maintainable and had poor reusability. All test scripts had to be manually started even though the steps are automatically executed, and the operator had to be present to observe the results.

This project is not intended to cover all levels of testing (unit, integration, system tests) and all types of testing (performance, usability, security, *etc.*) but rather the implementation of a set of acceptance-level functional tests using the proposed framework.

1.3 Motivation of Project

Being the leading online advertising business group in property sector in Asia region, there are many competitors attempting to catch up and outdo each other within the business ecosystem. New features and services must be developed, tested, and deployed rapidly to either to respond to the changing market needs as well as to proactively leading the innovation within the industry. With various categories of web products multiplied by various brands and countries, the efforts needed to develop, maintain, and operate these products requires an immense amount of coordination among teams and within team members.

Current practice of using manual regression testing results in long turnaround time to know whether the newly integrated features and defect fixing are affecting the existing features of the product. This project serves as a proof-of-concept of using automated regression testing for acceptance tests using keyword driven approach for improved turnaround time of feedbacks, better visibility of tests done, minimise test script redundancy, and full regression execution.

1.4 Project Objectives

The objectives of the implementation of the automated regression testing framework for web-based product development are:

i. To setup automated keyword-driven testing framework for better visibility of test status

ii. To integrate the framework with existing build automation system for rapid test execution

iii. To develop reusable keywords and/or test libraries for better test scripts maintainability

iv. To create self-documenting test cases for easier troubleshooting and traceability to defects

1.5 Project Scopes

The scopes of the project are:

i. Implementation of automated testing framework within iProperty Singapore IT Team
    a. To integrate Robot Framework (RF) into current build management infrastructure (Jenkins)

ii. Creation and execution of acceptance-level functional regression tests
    a. To create test cases using Selenium2Library (S2L) in RF format

iii. Writing up project's documentations
    a. To prepare documentations for setup and integration of RF into existing SDLC processes and workflows

REFERENCES

1. 29119-1. *Software Testing - Concepts and definitions*. vol. 2013. ISO/IEC/IEEE. 2013.

2. Bansal, A. A Comparative Study of Software Testing Techniques. *International Journal of Computer Science and Mobile Computing*, 2014. 3(6): 579–584. URL http://www.ijcsmc.com/docs/papers/June2014/V3I6201499a11.pdf.

3. Ran, H., Zhuo, W. and Jianfeng, X. Web Quality of Agile Web Development. *2009 IITA International Conference on Services Science, Management and Engineering*, 2009: 426–429. doi:10.1109/SSME.2009.112. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5233257.

4. 29119-4. *Software Testing - Test techniques*. 40. ISO/IEC/IEEE. 2013.

5. Connolly, D., Keenan, F. and McCaffery, F. Developing acceptance tests from existing documentation using annotations: An experiment. *2009 ICSE Workshop on Automation of Software Test*, 2009: 123–129. doi:10.1109/IWAST.2009.5069050. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5069050.

6. Beck, K. Manifesto for Agile Software Development. URL http://agilemanifesto.org/.

7. dos Santos, A. M., Karlsson, B. F., Cavalcante, A. M., Correia, I. B. and Silva, E. Testing in an agile product development environment: An industry experience report. *2011 12th Latin American Test Workshop (LATW)*, 2011: 1–6. doi:10.1109/LATW.2011.5985897. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5985897.

8. Collins, E. F. and de Lucena, V. F. Software Test Automation practices in agile development environment: An industry experience report. *2012 7th International Workshop on Automation of Software Test (AST)*, 2012: 57–63. doi:10.1109/IWAST.2012.6228991. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6228991.

9. Rossberg, J. and Olausson, M. Introduction to Agile. ... *Lifecycle Management with Visual Studio 2012*, 2012: 91–99. URL http://link.springer.com/chapter/10.1007/978-1-4302-4345-8_6.

10. Hellmann, T. D., Chokshi, A., Abad, Z. S. H., Pratte, S. and Maurer, F. Agile Testing: A Systematic Mapping across Three Conferences: Understanding Agile Testing in the XP/Agile Universe, Agile, and XP Conferences. *2013 Agile Conference*, 2013: 32–41. doi:10.1109/AGILE.2013.10. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6612876.

11. Stelzmann, E. Contextualizing agile systems engineering. *IEEE Aerospace and Electronic Systems Magazine*, 2012. 27(5): 17–22. ISSN 0885-8985. doi:10.1109/MAES.2012.6226690. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6226690.

12. Zhang, X. S. and Dorn, B. Agile Practices in a Small-Scale, Time-Intensive Web Development Project. *2011 Eighth International Conference on Information Technology: New Generations*, 2011: 1060–1061. doi:10.1109/ITNG.2011.187. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5945388.

13. Eck, A., Uebernickel, F. and Brenner, W. Fit for Continuous Integration: How Organizations Assimilate an Agile Practice. 2014: 1–11. URL http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1275&context=amcis2014.

14. Chaipraserth, R., Leelasantitham, A. and Kiattisin, S. A test automation framework in POCT system using TDD techniques. *2013 13th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE. 2013. ISBN 978-1-4673-5580-3. 600–604. doi:10.1109/ISCIT.2013.6645931. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6645931.

15. Collins, E., Dias-Neto, A. and Lucena Jr., V. F. D. Strategies for Agile Software Testing Automation: An Industrial Experience. *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops*, 2012: 440–445. doi:10.1109/COMPSACW.2012.84. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6341616.

16. Haugset, B. and Stalhane, T. Automated Acceptance Testing as an Agile Requirements Engineering Practice. *2012 45th Hawaii International Conference on System Sciences*, 2012: 5289–5298. doi:10.1109/HICSS.2012.127. URL http://ieeexplore.ieee.org/lpdocs/epic03/

wrapper.htm?arnumber=6149535.

17. Hu, R., Wang, Z., Hu, J., Xu, J. and Xie, J. Agile Web Development with Web Framework. *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008: 1–4. doi: 10.1109/WiCom.2008.2960. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4681149.

18. de Castro, A. M. F. V., Macedo, G. a., Collins, E. F. and Dias-Neto, A. C. Extension of Selenium RC tool to perform automated testing with databases in web applications. *2013 8th International Workshop on Automation of Software Test (AST)*, 2013: 125–131. doi:10.1109/IWAST.2013.6595803. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6595803.

19. Sampath, S. and Mihaylov, V. Composing a framework to automate testing of operational web-based software. *Software Maintenance, ...*, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1357795.

20. Leotta, M., Clerissi, D., Ricca, F. and Spadaro, C. Repairing Selenium Test Cases: An Industrial Case Study about Web Page Element Localization. *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*, 2013: 487–488. doi:10.1109/ICST.2013.73. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6569767.

21. Xu, D., Xu, W., Bavikati, B. K. and Wong, W. E. Mining Executable Specifications of Web Applications from Selenium IDE Tests. *2012 IEEE Sixth International Conference on Software Security and Reliability*, 2012: 263–272. doi:10.1109/SERE.2012.39. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6258316.

22. Bruns, A., Kornstadt, A. and Wichmann, D. Web application tests with selenium. *Software, IEEE*, 2009: 0–3. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5222802.

23. Wang, X. and Xu, P. Build an Auto Testing Framework Based on Selenium and FitNesse. *2009 International Conference on Information Technology and Computer Science*, 2009: 436–439. doi:10.1109/ITCS.2009.228. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5190273.

24. Ghodeswar, A. and Kulkarni, Y. SMEASY: A GUI Automation Framework

On the Top of the Selenium. *International Journal*, 2013. 3(2): 103–112. URL http://pakacademicsearch.com/pdf-files/com/237/103-112Vol.3,Issue2,Jun2013.pdf.

25. Razak, R. A. and Fahrurazi, F. R. Agile testing with Selenium. *2011 Malaysian Conference in Software Engineering*, 2011: 217–219. doi:10.1109/MySEC.2011.6140672. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6140672.

26. Leotta, M., Clerissi, D., Ricca, F. and Spadaro, C. Improving Test Suites Maintainability with the Page Object Pattern: An Industrial Case Study. *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops*, 2013: 108–113. doi:10.1109/ICSTW.2013.19. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6571616.

27. Wang, F. and Du, W. A Test Automation Framework Based on WEB. *2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, 2012: 683–687. doi:10.1109/ICIS.2012.21. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6211171.

28. Nagowah, L. and Doorgah, K. Improving test data management in record and playback testing tools. *2012 International Conference on Computer & Information Science (ICCIS)*, 2012: 931–937. doi:10.1109/ICCISci.2012.6297159. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6297159.

29. Leotta, M., Clerissi, D., Ricca, F. and Tonella, P. Capture-replay vs. programmable web testing: An empirical assessment during test case evolution. *2013 20th Working Conference on Reverse Engineering (WCRE)*, 2013: 272–281. doi:10.1109/WCRE.2013.6671302. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6671302.

30. Hametner, R., Winkler, D. and Zoitl, A. Agile testing concepts based on keyword-driven testing for industrial automation systems. *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 2012: 3727–3732. doi:10.1109/IECON.2012.6389298. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6389298.

31. Pajunen, T., Takala, T. and Katara, M. Model-Based Testing with a General Purpose Keyword-Driven Test Automation Framework. *2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops*, 2011: 242–251. doi:10.1109/ICSTW.

2011.39. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5954415.

32. Gopularam, B. P., Yogeesha, C. B. and Periasamy, P. Highly scalable model for tests execution in cloud environments. *2012 18th International Conference on Advanced Computing and Communications (ADCOM)*, 2012: 54–58. doi:10.1109/ADCOM.2012.6563584. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6563584.

33. Haugset, B. and Hanssen, G. K. The Home Ground of Automated Acceptance Testing: Mature Use of FitNesse. *2011 AGILE Conference*, 2011: 97–106. doi:10.1109/AGILE.2011.37. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6005490.

34. Hanssen, G. and Haugset, B. Automated acceptance testing using fit. *System Sciences, 2009. HICSS'09. ...*, 2009: 1–8. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4755765.

35. Agarwal, A., Garg, N. K. and Jain, A. Quality assurance for Product development using Agile. *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, 2014: 44–47. doi:10.1109/ICROIT.2014.6798281. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6798281.

36. Sumrell, M. From Waterfall to Agile - How does a QA Team Transition? *Agile 2007 (Agile 2007)*, 2007: 291–295. doi:10.1109/AGILE.2007.29. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4293611.

37. Liu, X., Lan, Y. and Ma, L. Design and Implementation of Automated Testing Framework for Linux Software GUI Testing. *2010 International Conference on Computational Intelligence and Software Engineering*, 2010: 1–4. doi:10.1109/WICOM.2010.5600880. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5600880.

38. Tang, J., Cao, X. and Ma, A. Towards Adaptive Framework of Keyword Driven Automation Testing. 2008. (September): 1631–1636.

39. Larman, C. and Vodde, B. *Practices for scaling lean & agile development: large, multisite, and offshore product development with large-scale Scrum*. c. 2010. ISBN 978-0321636409. URL http://books.google.com/books?hl=en&lr=&id=fqdTsH36TVYC&oi=fnd&pg=PT3&dq=Practices+for+Scaling+Lean+&+Agile+Development:+Large,+Multisite,+and+Offshore+

Product+Development+with+Large-Scale+Scrum&ots=6Gbuf6WiCM&sig=iWx2VplbpeUwunwmFKtyYa7rk00.

40. Stresnjak, S. and Hocenski, Z. Usage of Robot Framework in Automation of Functional Test Regression. *The Sixth International Conference on Software Engineering Advances*, 2011. (c): 30–34. URL http://www.thinkmind.org/index.php?view=article&articleid=icsea_2011_2_10_10057.

41. Durrani, U. K., Pita, Z. and Richardson, J. Coexistence of agile and SCM practices: An exploratory study of Australian agile software development organizations. *Journal of Systems and Information Technology*, 2014. 16(1): 20–39. ISSN 1328-7265. doi:10.1108/JSIT-09-2013-0045. URL http://www.emeraldinsight.com/10.1108/JSIT-09-2013-0045.

42. Altarawneh, H. and Shiekh, A. E. A Theoretical Agile Process Framework for Web Applications Development in Small Software Firms. *2008 Sixth International Conference on Software Engineering Research, Management and Applications*, 2008: 125–132. doi:10.1109/SERA.2008.14. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4609418.

43. Bass, J. M. Influences on Agile Practice Tailoring in Enterprise Software Development. *2012 Agile India*. IEEE. 2012. ISBN 978-0-7695-4657-5. 1–9. doi:10.1109/AgileIndia.2012.15. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6170020.

44. Chen, J., Lin, M., Yu, K. and Shao, B. When a GUI regression test failed, what should be blamed? *Software Testing, Verification ...*, 2012: 468–471. doi:10.1109/ICST.2012.59. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6200139.

45. Stolberg, S. Enabling Agile Testing through Continuous Integration. *2009 Agile Conference*, 2009: 369–374. doi:10.1109/AGILE.2009.16. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5261055.