Agile Testing Practices in Academia

Premalatha Sampath
Whitireia New Zealand
Premalatha.Sampath@whitireia.ac.nz

ABSTRACT

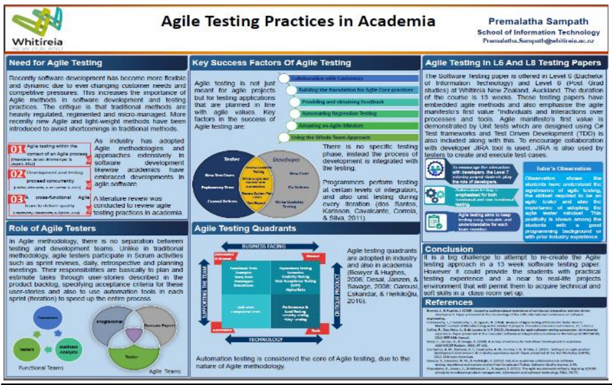
Recently software development has become flexible and dynamic due to ever-changing customer needs and competitive pressures. This competitive pressure increases the importance of Agile methods in software development and testing practices. The critique is that traditional methods are heavily regulated, regimented and micro-managed. More recently new Agile and light-weight methods were introduced to avoid shortcomings in traditional methods. The emphasis of agile methodologies is to deliver software in iterations and with user priority in mind. As industry has adopted the Agile methodologies and approach extensively in software development likewise academics have embraced developments in agile software. However, Agile Testing (AT) is still in its infancy in Academia. The purpose of this poster is to examine the adoption of Agile testing practices in Level 6 and Level 8 Software Quality and testing papers. AT means testing within the context of an Agile process such as Scrum or XP. Agile suggests that development and testing, are two essential functions of software development which proceed concurrently. The poster illustrates how the key Agile testing practices such as Test-Driven Development (TDD), Automated tests, writing unit tests, supporting team work and constant collaboration have been adopted in Software Quality and Testing papers. This literature review has been conducted to investigate the adoption of Agile Testing practices in Academia.

Keywords: Agile Testing, Test-Driven Development

1. INTRODUCTION

Testing is one of the cornerstones of agile software development and has received attention ever since its initial development. Many agile practices are reliant on effective software testing principles. This implies that the quality of the code is good and that a substantial effort has been made to fix bugs post-release (Chakravorty, Chakraborty, & Jigeesh, 2014). Agile Testing involves a cross-functional Agile team actively relying on the special expertise contributed by Testers. This allows the combined team to meet the project's timeline objectives, ensuring quality and usability. Several Agile software development methods in programming courses and Project management courses in academia were extensively reported in the literature. However a lack of research in Agile testing methods in academia is the key motivation for this poster paper.

Curriculum guidelines for the undergraduate degree programme in the information system (SFIA, 2010) and Skills Framework for the Information Age (SFIA) identified (Institute of IT Professionals New Zealand, 2016) that teamwork and collaboration skills are some of the main exit characteristics of IS graduates. Collaborative learning involves the joint intellectual effort by groups of students working together. Agile testing can help to encourage team collaboration.



Agile Testing Practices in Academia

Premalatha Sampath
School of Information Technology
Premalatha.Sampath@whitireia.ac.nz

Need for Agile Testing

Recently software development has become more flexible and dynamic due to ever-changing customer needs and competitive pressures. This increases the importance of Agile methods in software development and testing practices. The critique is that traditional methods are heavily regulated, regimented and micro-managed. More recently new Agile and light-weight methods have been introduced to avoid shortcomings in traditional methods.

01 Agile testing within the context of an Agile process (Vlaanderen, Jansen, Brinkkemper, & Jaspers, 2011)

02 Development and testing proceed concurrently

03 A cross-functional Agile team to deliver quality

As industry has adopted Agile methodologies and approaches extensively in software development likewise academics have embraced developments in agile software.

A literature review was conducted to review agile testing practices in academia.

Key Success Factors Of Agile Testing

Agile testing is not just meant for agile projects but for testing applications that are planned in line with agile values. Key factors in the success of Agile testing are:

Collaboration with Customers
Building the foundation for Agile Core practices
Providing and obtaining Feedback
Automating Regression Testing
Adopting an Agile Mindset
Using the Whole Team Approach

There is no specific testing phase, instead the process of development is integrated with the testing.

Programmers perform testing at certain levels of integration, and also unit testing during every iteration (dos Santos, Karlsson, Cavalcante, Correia, & Silva, 2011).

Agile Testing In L6 And L8 Testing Papers

The Software Testing paper is offered in Level 6 (Bachelor of Information Technology) and Level 8 (Post Grad studies) at Whitireia New Zealand, Auckland. The duration of the course is 13 weeks. These testing papers have embedded agile methods and also emphasise the agile manifesto's first value "Individuals and Interactions over processes and tools. Agile manifesto's first value is demonstrated by Unit tests which are designed using C# Test frameworks and Test Driven Development (TDD) is also included along with this. To encourage collaboration with developer JIRA tool is used. JIRA is also used by testers to create and execute test-cases.

To encourage the interaction with developers, the Level 7 industry project students play the role of developers.

Automation testing is emphasised for both functional and non-functional testing.

Agile testing aims to keep testing easy, reusable and understandable for each team member.

Tutor's Observation

Observation shows the students have understood the significance of agile testing, the skillset required to be an agile tester and also the importance of adopting the agile tester mindset. This positivity is shown among the students with a good programming background or with prior industry experience.

Role of Agile Testers

In Agile methodology, there is no separation between testing and development teams. Unlike in traditional methodology, agile testers participate in Scrum activities such as sprint reviews, daily, retrospective and planning meetings. Their responsibilities are basically to plan and estimate tasks through user-stories described in the product backlog, specifying acceptance criteria for these user-stories and also to use automation tools in each sprint (iteration) to speed up the entire process.

Agile Testing Quadrants

Agile testing quadrants are adopted in industry and also in academia (Bowyer & Hughes, 2006; Desai, Janzen, & Savage, 2008; Garousi, Eskandar, & Herkiloğlu, 2016).

Automation testing is considered the core of Agile testing, due to the nature of Agile methodology.

Conclusion

It is a big challenge to attempt to re-create the Agile testing approach in a 13 week software testing paper. However it could provide the students with practical testing experience and a near to real-life projects environment that will permit them to acquire technical and soft skills in a class-room set-up.

References

[illegible]

This poster appeared at the 8th annual conference of Computing and Information Technology Research and Education New Zealand (CITRENZ2017) and the 30h Annual Conference of the National Advisory Committee on Computing Qualifications, Napier, New Zealand, October 2-4, 2017.

2. ROLE OF AGILE TESTERS

In Agile methodology, there is no separation between testing and development teams. Unlike in traditional methodology, agile testers participate in Scrum activities such as sprint reviews, sprint retrospective and planning meetings (Vlaanderen, Jansen, Brinkkemper, & Jaspers, 2011). Their responsibilities are basically to plan and estimate tasks through user-stories described in the product backlog, specifying acceptance criteria for these user-stories and also to use automation tools in each sprint (iteration) to speed up the entire process. The testing process for agile methodology used in sprints is incremental and iterative. Agile testers should also plan and update test cases for user-stories, test scripts are automated as required to execute all tests, report defects and run regression tests.

In Agile methodology, there is no specific testing phase, instead the testing is integrated into the development process. Programmers perform testing at certain-levels of integration, and also unit testing during every iteration (Santos et al., 2011). Acceptance tests are performed by the customer, thus minimising the overall risk and allowing the project to adapt to changes. Quality Assurance activity is practiced at a high level in the Agile method relying on customer feedback.

3. KEY SUCCESS FACTORS OF AGILE TESTING

Agile testing is not just meant for agile projects but for testing applications that are planned in line with agile values. Agile testing approaches are successful due to the following key factors (Crispin & Gregory, 2009):

- Collaborating with Customers
- Building the foundation for Agile Core Practices
- Providing and Obtaining Feedback.
- Automating Regression Testing
- Adopting an Agile Testing Mindset
- Using the Whole Team Approach.

4. DIFFERENT AGILE METHODS

Different Agile methods follow different strategies in practice to ensure software quality assurance activities. These practices are broadly summarised as follows (Rajasekhar & Shafi, 2014):

a) Unit testing: One of the advantages of Agile testing is that testing is performed by developers. Developers test their code as they write unit tests called unit testing. Unit testing is one of the best approaches used to communicate over issues in the code during development without waiting for the other units to be available. It provides fault detection at a lower cost compared to performing testing at a later stage. Unit tests are foundations of any Agile projects and effective unit tests can be done with automation (Gil et al., 2016).

b) Automation: Manual test cases can be tedious and rigorous to document and also tedious to maintain. The execution of the manual tests and the examination of the results can be error-prone and time consuming. One of the disadvantages noted in the industry is when the schedule pressures rise, manual testing often gets neglected. It is noticed that to deliver projects efficiently unit testing developers could employ automated tools for unit testing. Agile projects emphasise automation testing since they have multiple releases which require repeated testing of units under development during integration. Apart from automating unit tests, some or part of the integration testing and acceptance testing is also automated. Automating tests is beneficial as it reduces the work-load on testers, reduces human-error over repeated tasks and is effective over cost and time (Collins, Dias-Neto, & V. F. d. Lucena, 2012; Gil et al., 2016).

c) Test Driven Development (TDD)
TDD is another important quality assurance activity in some Agile methods (Bissi, Neto, & Emer, 2016). TDD is a type of unit test in which programmers write test cases before they start programming. Therefore, the actual tests start before the programming. This also contains short iterations and test cases are developed before the code is written, in order for the test cases to be passed. TDD is a more cohesive and less coupled code compared to the traditional approach (Janzen & Saiedian, 2008). The significance of TDD is that it uses automated tests that can be used as regression tests when a new feature is included. However, the difference between TDD and traditional unit testing is that the test programs are written before the application code is written.

5. AGILE TESTING APPROACH IN L6 AND L8 TESTING PAPERS

The Software Testing paper is offered in Level 6 (Bachelor of Information Technology) and Level 8 (Post Grad studies). The duration of the course is 13 weeks. These software testing papers have embedded agile methods and also emphasise the agile manifesto's first value "Individuals and Interactions over processes and tools. Agile manifesto's first value is demonstrated as Unit tests which are designed using C# Test frameworks and Test Driven Development (TDD) is also included along with this. To encourage collaboration with developers the JIRA tool is used. JIRA is also used by testers to create and execute test-cases.

To encourage the interaction with developers and understand agile methods in the software testing paper, the Level 7 industry project students play the role of developers. Although, these students may not experience the entire working environment of an agile project, it allows them to develop a good understanding of the agile methods applied in these projects. Automation testing is emphasised for both functional and non-functional testing. Selenium is used for functional testing test-cases for web testing and other tools such as JMeter, Blazemeter, WebAim, Web Link Validator, Netsparker are used to perform non-functional testing.

Observation shows the students have understood the significance of agile testing, the skillset required to be an agile tester and also the importance of adopting the agile tester mindset. This positivity is shown among the students with a good programming background or with prior industry experience.

6. CONCLUSION

The agile testing quadrants provide a model for thinking about testing in an agile world. The Quadrants help to emphasize the whole-team responsibility for testing. They also provide a visible mechanism about testing requirements. Agile testing uses continuous feedback, which allows users to redirect the process during software development. Agile testing aims to keep testing easy, reusable and understandable for each team member.

The attempt to re-create the whole-team approach in a 13 week software testing paper is a challenge, although it may not provide 100% experience of an Agile project work environment, however it provides a good understanding of the Agile methods applied in these software testing projects. It could provide the students with practical testing experience and a near to real-life projects environment that will permit them to acquire technical and soft skills in a class-room set-up.

7. REFERENCES

Bissi, W., Serra Seca Neto, A. G., & Emer, M. C. F. P. (2016). The effects of test driven development on internal quality, external quality and productivity: A systematic review. *Information and Software Technology, 74*(Supplement C), 45-54. doi:https://doi.org/10.1016/j.infsof.2016.02.004

Chakravorty, T., Chakraborty, S., & Jigeesh, N. (2014). Analysis of agile testing attributes for faster time to market: Context of manufacturing sector related IT projects. *Procedia Economics and Finance, 11*(Supplement C), 536-552. doi:https://doi.org/10.1016/S2212-5671(14)00219-6

Collins, E., Dias-Neto, A., & V. F. d. Lucena, J. (2012, 16-20 July 2012). Strategies for agile software testing automation: An industrial experience. In *IEEE 36th Annual Computer Software and Applications Conference Workshops.* Symposium conducted at the meeting of 2012 COMPSACW, Izmir, Turkey. doi:10.1109/COMPSACW.2012.84

Crispin, L., & Gregory, J. (2009). *Agile testing: A practical guide for testers and agile teams*. Potsdam. Pearson Education.
Addison-Wesley Professional.

Gil, C., Diaz, J., Orozco, M., De La Hoz, A., De La Hoz, E., & Morales, R. (2016). Agile testing practices in software quality: State of the art review. *Journal of Theoretical and Applied Information Technology, 92*(1), 28.

Institute of IT Professionals New Zealand. (2016). IITP degree accreditation. Retrieved Aug 27, 2017, from https://itp.nz/Activities/Degree-Accreditation

Janzen, D., & Saiedian, H. (2008). Does test-driven development really improve software design quality? *IEEE Software, 25*(2), 77-84. doi:10.1109/MS.2008.34

Rajasekhar, P., & Shafi, R. M. (2014). Agile software development and testing: Approach and challenges in advanced distributed systems. *Global Journal of Computer Science and Technology, 14*(1), 7-14.

Santos, A. M. d., Karlsson, B. F., Cavalcante, A. M., Correia, I. B., & Silva, E. (2011, 27-30 March 2011). Testing in an agile product development environment: An industry experience report Symposium conducted at the meeting of the 2011 12th Latin American Test Workshop (LATW). doi:10.1109/LATW.2011.5985897

SFIA. (2010). SFIA skills and levels: Curriculum guidelines for undergraduate degree programs in information systems. Retrieved Aug 28, 2017, from https://itp.nz/Members/Certification/Other-information/SFIA skills and-levels

Vlaanderen, K., Brinkkemper, S., Jansen, S., & Jaspers, E. (2009, 1-1 Sept. 2009). The Agile Requirements Refinery: Applying SCRUM Principles to Software Product Management Symposium conducted at the meeting of the 2009 Third International Workshop on Software Product Management doi:10.1109/IWSPM.2009.7