

Agile Software Testing Technologies in a Large Scale Project

Bojana Koteska

Faculty of Computer Science and Engineering
"Rugjer Boshkovikj" 16 P.O. Box 393
1000 Skopje, FYR Macedonia
+38970453393

koteska.bojana@students.finki.ukim.mk

Anastas Mishev

Faculty of Computer Science and Engineering
"Rugjer Boshkovikj" 16 P.O. Box 393
1000 Skopje, FYR Macedonia
+38970330386

anastas.mishev@finki.ukim.mk

ABSTRACT

Software testing in agile development is very complex issue in the IT world. Since agile methods do not focus much on testing activities and agile software development does not include testing practices required for a quality project, the process of finding the suitable method for testing, especially for large scale projects is challenging. In this paper, we propose a concept of integrated agile software testing in a large scale project. This research explains the software testing process technologies and principles for agile software testing and identifies the best practices for testing software products in agile development process.

Categories and Subject Descriptors

D.2.4 [Software Engineering]: Software/Program Verification – Validation, Reliability

General Terms

Reliability, Verification

Keywords

Agile software testing, large-scale software project, agile software development

1. INTRODUCTION

Software testing should not be defined as a process of finding errors in the execution of a program, but it should be understood as a mental discipline where test engineers can express their creativity, find the most appropriate testing method or suggest new testing ideas. Mature organizations have master software testing skills and techniques and they see testing as a mental discipline that can help the all processes in the software lifecycle [1].

Many mature organizations adopt agile software development techniques to increase the productivity. The main reason for adopting the agile development methods during the project lifecycle is to produce higher quality software in less time while reducing development costs. However, the most important issue that comes along with the agile software development is testing.

Agile software development and testing could be easily integrated in a small project, but the real challenge for organization is to find a suitable testing technology for testing a large scale project. The

process of testing during the agile software development is not unified and it must be adjusted to the project requirements, organization's policies and team ability in order to satisfy the customer's needs.

Table 1: Agile Development Manifesto

Individuals and interactions	processes and tools
Working software	comprehensive documentation
Customer collaboration	contract negotiation
Responding to change	following a plan

According to agile manifesto shown in table 1, agile software development adds high value to the left side items. Agile development approach allows requirements' changes and it means that changes can be made also in test cases. That process requires better communication between developers, testers and end users in order to overcome the problems and make more flexible and optimized solutions. The individuals and their interactions are the most important part in the project, i.e. the collaboration between the team members is responsible for better learning environment where new members could learn things from the senior and most experienced team members [7].

Since agile development requires multifunctional teams that follow the principles of iterative and incremental developing practices, the testing process should be efficient and it requires making tests early and often. There must be clear definition what the results of testing should be at the end of each sprint. It means that tests have to be done before the implementation of the project functionalities in each sprint.

The key factor of successful testing is close collaboration between end users, developers and testers. Testers should be part of the development team and all activities must be parallelized as much as possible. For example, while testers are working on test cases, developers are coding the user stories. Because the agile development paradigm says that all the processes must be done fast, there is no so much time for testing, so the test data preparation should be done in the planning phase in the development process [7]. Better approach is these test data to be prepared together with the stakeholders in order to satisfy the functional and performance requirements.

Since iterative development is focused on providing new software at the end of each of the iterations, there must be new software part with indented functionality. Time required for each of the

BCI'12, September 16–20, 2012, Novi Sad, Serbia.

Copyright © 2012 by the paper's authors. Copying permitted only for private and academic purposes. This volume is published and copyrighted by its editors.

Local Proceedings also appeared in ISBN 978-86-7031-200-5, Faculty of Sciences, University of Novi Sad.

iterations depends of the project size and used agile methods, but usually it takes from two to eight weeks. Agile software testing enables testers to express their creativity and to think in different directions leading to effective development process and end user's satisfaction [4].

The open question here is how to find the most suitable method for testing in agile developing environment. Like we mentioned before, the process of agile testing is not unified and it requires additional experience and analysis to be performed successfully. Since agile software development is a high-level framework, every team has to define its own strategies and plans about software testing [8].

The hardest part is to make testing method ensuring that the solution of the "problem" in the software development at the current iteration is done properly. That is the art of the software testing where not only formal methods and theory have to be included, but the new strategies and ideas also.

2. AGILE SOFTWARE TESTING VS TRADITIONAL SOFTWARE TESTING

When a company uses traditional waterfall model for developing software, that means that testing is performed at the end of development process. That is the real problem when we talk about quality software. Testing at the end means increased possibility of errors at the beginning stages. It means that if error is found at the beginning stage, the stages after it have to be re-evaluated. That will increase the costs and reduce the time intended for developing.

Agile software testing can help reducing the errors in the first stages. The good thing is that agile software testing methods can be included in traditional waterfall developing methods. For example, if company has started developing project using waterfall model, agile software testing methods can be easily adapted to the process [3].

The main benefit of using agile software testing is reducing the errors in the beginning stages. If we want to provide assurance for quality, we should test often. Agile software testing methods enable testing during all stages and the most important thing is that testing is performed on small units.

Agile software testing is new testing technology that enables thinking in different directions in order to provide optimized testing and satisfied customers. Agile testing methods increase the efficiency in providing results for a less time. Agile approach is great approach in many situations but it does not mean that it should be used anywhere. Development and testing team must always decide about method they will use in the development process. They must find the most appropriate method that will guarantee success.

3. AGILE SOFTWARE TESTING PRINCIPLES IN LARGE SCALE PROJECT

Agile software developing methods are becoming important part of the software development evolution process. Leading companies in the software industry adopt these methods fast in order to make their products to be completed early and efficient.

Testing methods in agile development processes play an important role when company wants to offer a quality product.

If team project members want to use agile software testing methods and principles, first of all they must have a vision about what should be achieved and how it can be done in a less time and more efficiently.

The list below shows the most common principles of agile software testing [7, 6]:

1. Agile software testing requires testing in the early stage and testing parallel with software development process.
2. Testers in the agile software testing team must adapt to fast changes.
3. Customer must be included in testing process in order to accept the new changes that had been made.
4. Testing must be performed at each of the iterations.
5. At the end of each of the iterations there must be provided a working version of the software.
6. The process of successful testing requires close interactions between customer and developer in order to save time by avoiding unnecessary re-designs and re-implementations.
7. Progress of the work can be calculated with the help of planned activities for current period of time.
8. Continuous integration is one of the most important steps in agile software testing and development because the development team should try to provide not only functionality of the individual tasks, but the business logic.
9. Testing process should be automated because it will provide faster testing results and efficient work progress.
10. Any changes in the process must be accepted by customers.

However, using these principles does not guarantee success. Everything depends on the team structure, project size and customer requirements. The key factor for success is the proper combination of these principles that will be aligned properly with the company development policies.

4. BEST PRACTICES IN AGILE SOFTWARE TESTING METHODOLOGIES

Like we mentioned before, software testing is the core for successful project and satisfied customers. The main question here is how to make the process of testing efficient, fast and correct. There is not unified process of selecting the proper practices when we are talking about best practices, but every team can choose the right ones for the project progress.

The first thing associated with professional testing is professional testers. In agile software development process everyone writes tests, so professional testers are required to write more precise and inventive tests, not only to provide more tests. Professional testers are included to improve the coordination between testers and developers also [9].

Agile testing should be automated. It means that test will be performed faster and more efficiently. The automation requires selection of the right tool or making own tool to satisfy the projects' testing needs, to save time and money. If the tool with the requirements for testing the current project could be found in the market, it is wiser to use that tool, instead of making new one. Testers should know that not always all testing processes should be automated and that some of them cannot be done without automation. The challenge is to find a balance between manual and automated testing. Agile development requires more automation of the processes, but it is useful when there is a repetitive task or regression testing task. Automation of test task that will be used one time is not necessary because it requires additional time that can be used for making new tests [2].

The next thing is planning of the quality activities. It means planning for time required for feature and regression testing and also time required for fixing the bugs. In agile software development time required for testing and coding should be equal. If team knows that one feature is more difficult than the others, then the time estimated for it should be prolonged.

On the other hand, regression testing is allocated as global time period because these tests are done in the end of each of the iterations. Difference in time between the net hours spending on regression testing and iterations is small and that is the reason why time for the regression testing is considered as global.

Time required for fixing the defects is also considered as global. That is because the time required for fixing the defect and complexity of the defect could never be predicted. Planning the defects is not good idea because we cannot predict a complex defect that will require much time to be fixed. The unrelated defects could require that much time for fixing also [9]. The main paradigm in agile software testing about fixing defects is to fix the defects as soon as possible.

4.1 Automation of Agile Software Testing

Automation of the agile software testing methods is primary intended to minimize the quantity of manual work involved in test execution and to gain higher coverage with greater number of test cases. Automated tests in agile testing developing environment are usually unit tests that are responsible for quality of the smallest possible modules [5].

This way of testing has a big influence on the tools sets used for test execution and how the tests are designed. It needs to take advantage of what is already known and develop the practices for fields where it is not entirely performed.

Writing of the tests could be almost a third of testing task. Since the technology for automation has not advanced as expected, the automated test generation tools sometimes generate too large test set which reduces the gains of automation. Anyway there are some good techniques and tools that are recognized as fine methods for the task. It is necessary to understand which of them are appropriate in what environments.

Agile software testing requires automated tools, but testers should keep in mind that tools must be simple and easy to learn. That means that tool must be user-friendly and it must provide a simple user interface. Automation of testing requires writing test scripts. All scripts must be prepared well to ensure providing the right result. Important thing here is to split the long test cases into smaller because the smaller ones are easily maintained. The

developers should also help the process of automation by adding an extra view point in solving problems [2, 5].

One of the most important questions here is how to choose the right tools for automation or how to make an own tool. There are lots of tools in the market, but the team knowledge and experience should help to choose the right one. Decision depends on project size, time allocated for the project and projects' costs. If developers and testers cannot find the appropriate tool in the market, they can create a new tool which suits their needs for testing. If there is already a tool that can satisfy the testing needs then it is wide decision to use that tool, instead of making a new one [2, 5].

The secret for successful testing is focusing on one story at a time and building functionality step-by-step. Continuous integration will help to integrate the unit tests and that will improve the process of automation of the tests for next units [5].

Automation of testing is not the best solution when making performance tests. These tests require time and enormous amount of data. If change is made to data model, then it must be propagated to test conditions. This process is difficult, especially when the size of the test conditions is equal to several GB [5].

The real application of automation is in regression testing. These tests are repeatable and they are usually made to simulate user actions and check the results. If results are positive, then it means that all functionalities are implemented correct [5].

5. OPTIMIZATION IN AGILE SOFTWARE TESTING METHOD

Here are some steps that can help agile software testing in order to optimize the process of agile testing:

- Client provides the functional requirements.
- They are reviewed by the Engineering team (Quality Assurance team).
- In the design and implementation phases, user stories are written and they are reviewed by customer.
- The requirement specification is updated with the new changes.
- When the implementation starts, making the test plans, cases and strategies are being made and all of them should be documented properly and reviewed by customer team.
- During the implementation, testing team determine if the testing code could be applied to software code and that will decrease the potential errors at the start of each developing iteration.
- The process iterates until the software is fully accepted.

However, we cannot be sure that the software is 100% error free when handed to the customer. This method is best approach for software with requirements that are changed rapidly.

There is no need to follow any specific process or method. For different project there should be defined appropriate strategies to ensure the system quality. Sometimes, better solution is to use heuristic approach that is based on previous test work experience.

6. CONCLUSION

Software testing is a mental activity that can be experimented, so testing can be expressed as a scientific field that can be continuously improved. For example, automate tools with an artificial intelligence, i.e. robots can help the software testing.

Agile testing is promising direction, so the future testers must be more technically professional to develop test stubs and drivers that interact with more programming languages that will work with more complex data sets and to be integrated with agile teams. Tools must be highly complex to support the testing of these data sets.

Testers in agile teams must be professionally educated and accredited with computer science degree and maybe some courses additionally. They must be concentrated on adding value to the business software development lifecycle. They should operate in position where the combination of their knowledge, creativity and ability for decisions for testing criteria will be main reason for success of the company.

The role of automation in agile testing environment will become more essential and testing and quality assurance will take part of modern technologies to satisfy the goals new products to the market rapidly at minimal risk. This kind of system interaction will lead to minimizing the risks and increasing the benefits of business processes in the companies, that is also the main reason for their existence.

7. REFERENCES

- [1] Ammann, P. and Offutt, J. 2008. *Introduction to software testing*. Cambridge University Press, UK.
- [2] Arora, A. 2008. *Agile Automation Testing*. Adobe Systems India Pvt. Ltd, Noida.
- [3] Deshpande, S. A., Deshpande, N. A., Marathe, V. M. and Garje G. V. 2010. Improving Software Quality with Agile Testing. *International Journal of Computer Applications*, 1, 66-71.
- [4] Harish, R., Madhu, B. K. and Lokesha, V. 2012. A Sophisticated Study on Best Practices of Agile Software Testing. *International Journal of Electronics Communication and Computer Engineering*, 3, 26-30.
- [5] Jureczko, M. The Level of Agility in Testing Process in a Large Scale Financial Software Project. In *Software Engineering Techniques in Progress*, Hruska T., Madeyski L. and Ochodek M., University of Technology, Wroclaw, Poland.
- [6] Madhu B. K., Jigalur, M. and Lokesha, V. 2010. A study on Agile Software Testing: Emergence and techniques. *African Journal of Mathematics and Computer Science Research*, 3, 288-289.
- [7] Nawaz, A. and Masood, M. K. 2008. Software Testing Process in agile development. Master thesis. Blekinge Institute of Technology, Sweden.
- [8] Puolitaival, O.P. 2008. *Adapting model-based testing to agile context*. Edita Prima Oy, Helsinki.
- [9] Talby, D., Keren A., Hazzan, O. and Dubinsky, Y. 2006. Agile Software Testing in a Large-Scale Project. *Software, IEEE*, 23, 30-37.