

Short communication

A study on Agile Software Testing: Emergence and techniques

B. K. Madhu¹, Megha Jigalur² and V. Lokesh^{3*}

¹Department of CSE, Amruta Institute of Engineering and Management Sciences, Anna University – Coimbatore, India.

²Acharya Institute of Technology, Rayaluseema University, Andhra Pradesh India.

³Department of MCA, Research Center, ACIT, Bangalore, India.

Accepted 5 August, 2010

Customer satisfaction, quick and efficient delivery of finished products is all that is needed to lead the way in addressing issues regarding numerous software life cycles. Starting from waterfall model to any present life cycle, we find various advantages and disadvantages to be implemented in software development. Software testing is the state that takes the much of elapse time of SDLC. Improving the testing time and optimizing the techniques will always result in fast and efficient finish of software product. In this direction, Agile Software Development (ASD) is a new tool addressing most of the issues encountered. As in ASD, Agile Software Testing (AST) carries the manifesto of agile methodology and follows the highest techniques and best practices of software testing life cycle. Though AST is used in many Product-Based Software Company, the optimal procedure and superior techniques yielding better results are yet to be discovered. This paper is aimed at guiding and forming a hub for all agile testing practitioners in addressing these issues. It also lists advantages of agile testing and core techniques of implementation process in AST.

Key words: Software development life cycle (SDLC), agile software development (ASD), agile software testing (AST).

INTRODUCTION

In waterfall development environments, there well-defined phases, each having a thorough review and authorization process that should be completed before moving on to the next stage. On the surface, this process is not inherently bad. Problems arise when the project requires adjustments and modifications that were not anticipated in the early stages. So, for projects that can be specified out with precision ahead of time, waterfall may be the proper choice. Unfortunately, lot of unpredictable stages and issues will be encountered in the process of developing the software (Cripsin and Gregory, 2009). The answer to this problem depends on what type of project we are addressing. We need to have strong life cycle which answers the changing needs of customers and to build software effectively. It also gives quality testing results and review points.

The emergence of Agile testing

In response to the problems highlighted above, development teams began shifting to frequent releases. At first, these iterative methods went with small cycles, but remained with a fixed schedule ahead of time (that is, fixed budget, strict plan, and predetermined feature sets). (Alessandro, 2009) So instead of the one major release, many minor releases would be scheduled. Agile would eventually take this concept one step further: Iterative releases, with no fixed plan. Here, each release adds a bit of value, allowing for better decisions as to what features would be included next. Development now works in small cycles – planning and designing only far enough, ahead to keep the flow working. With this increased flexibility, the burden of excessive formal documentation was greatly alleviated. More importantly, we now incorporate testing into each step of the process, not just at the end of each release. Although agile is gaining the upper-hand in terms of popularity, this piece is not an endorsement of one over the other. Many of the concepts

*Corresponding author. E-mail: lokiv@yahoo.com. Tel: 9448345075.

discussed in this paper are based on agile concepts, but are relevant to both types of development processes. (Juristo, 2002). If traditional waterfall methodologies are used, by adopting agile concepts benefit can be noticed distinctively. Several agile methods exist. Some of the well known include extreme programming (XP), lean software development, crystal, DSDM (dynamic systems development method), scrum, and feature-driven development (FDD). Lean - Moves closer to customer, shorter cycles, eliminates waste, decides as late as possible, empowers the team, and builds integrity. DSDM - Empowers the team to make decisions, emphasizes frequent product delivery, integrate testing throughout, and promotes collaboration and cooperation between all stakeholders. FDD - Centers development on the feature and creates a domain model with domain experts. Crystal - Emphasize people, gather techniques from other methods, improve communications, adapt the process itself (shrink or grow to fit). Scrum - Manages a prioritized list of requirements on a product backlog, collaborate through daily standup meetings, exhibit the product upon iteration completion, use retrospectives to correct the process. XP – Emphasizes the values of communication, simplicity, feedback, and courage; use specific technical and collaborative practices, refactoring, pair programming, continuous integration, open workspace, and automated acceptance tests.

BEST PRACTICES OF AGILE METHODOLOGIES

Software engineering can often be complex. To succeed, it demands the use of IT methodologies. However off-the-shelf methodologies can often be unwieldy or inflexible. There has been a notable shift towards more iterative development methodologies in the past decade, as researchers and technology leaders sought increased adaptability and flexibility that was not deemed possible using traditional 'waterfall' methods (Lisa and Bob, 2009). Agile methodology gives birth to thinking in different directions leading to customer satisfaction and optimized software development. It provides optimal approach for forming small testing groups of highly motivated resources, working with a defined goal. This approach of testing not only complete testing life cycle in quick time but also ensures a proper quality testing will be implemented. Agile software testing states an effective approach in providing effective results in best possible time-scale. To practice and implement any process, principals and methodologies need to be defined and freezed (Dyba, 2008). The following are points which shed light on the said topic:

Principles of Agile software testing

- i) Customer satisfaction by quick and continuous delivery of software product, without compromising the quality of the testing process.
- ii) Working version of the software is released at every regular interval making a way for the customer to check acceptability and correctness of the software product.
- iii) Progress of completed project will be decided by work completion and by having gateways which meets the timeline as planned.
- iv) Any changes in the ongoing process are accepted and suitable adjusted. Also those changes are accommodated immediately.
- v) Close circular interactions between customer and developer which results in avoiding re-designing and re-implementations.
- vi) Projects are built by self-motivated trustable personnel with high technical skill and with standard testing steps.
- vii) Continuous attention to technical excellence and good design.
- viii) Developing and modeling is done in simplicity and care is taken to optimize the process.

RESULTS AND CONCLUSION

If testing process is to succeed in Agile life cycle, the AST must be a central pillar of the development process and proper review process need to be implemented. A collaborative team of developers and testers need to be established. When executed well, agile is not chaotic at all. In fact, it is letting the developers run wild and testers to perform optimally. As a self-motivated group will work on continuous basics, a clear picture of goals can be viewed through agile methodology and timelines can be met.

REFERENCES

- Alessandro C (2009). The Transition from Fragile Testing to Agile Testing, Swiss Testing Day Tutorial, http://www.swisstestingday.ch/tutorial2_1.htm. Agile Conference.
- Crispin L, Gregory J (2009). Agile Testing: A Practical Guide for Testers and Agile Teams, Pearson Education.
- Dyba T, Dingsøyr T (2008). Empirical Studies of Agile Software Development: A Systematic Review, Information and Software Technology, doi: 10.1016/j.infsof.2008.01.006.
- Juristo N, Moreno AM, Vegas S (2002). "A Survey on Testing Technique Empirical Studies: How Limited is our Knowledge", Int. Symp. Empirical Software Eng. (ISESE'02), pp. 161.
- Lisa C, Bob G (2009). Agile Conference "The Testing Stage" Sessions proposal, <http://agile2009.agilealliance.org/testing>.