

A review paper agile testing technique for software testing

Avinash Bhagat

**Assistant Professor, School of Computer Science & Engineering,
Lovely Professional University, Phagwara, Punjab, India.**

Abstract

The software testing is the technique of software engineering to test software faults. The various type of testing is there which are a black box, white box, automatic, etc. the agile testing contains multiple testing which is applied on the software to detect a maximum number of faults. The major issue of agile testing is to define the exact sequence of the testing. In the base paper, the genetic algorithm is applied to define the sequence of testing. In the genetic algorithm, the objective function is static. Due to which, its efficiency is less as compared to an algorithm in which objective function is dynamic. In this research work, the technique will be proposed which has a dynamic objective function to define the exact sequence of testing in agile testing.

1. Introduction

The testing phase of software development sometimes faces various issues concerning the developers and IT managers involved. However, the quality of software can be determined only with the help of testing which also tells whether the application can perform properly or not. Within the Software Development Lifecycle (SDLC), the testing plays a very important role. There are various risks taken within the companies due to which the software might perform poorly if the testing strategy applied is not good. The software testing practice in which the principles of agile software development are followed is known as agile testing. The testing is integrated into the development process with the help of agile development and not kept aside from the SDLC phase. A cross-functional agile team is included within the agile testing which depends on the special expertise that the testers provide. A “whole-team” mechanism is utilized to enhance the quality of the software product by the agile team. The team collaborates actively with the development team as the testing occurs within the real-time. This results in identifying and transferring the issues into the specifications which are executable and protect the coding. The approval of the detailed design documents of the development team is a must before initializing the testing [1]. This can help in initiating the generation of test cases. Within the Sprints or Iterations, the coding, as well as testing, is performed incrementally and iteratively. This helps in generating each feature until there is enough stability delivered. This results in improving the product’s quality.

The specifications of a business are analyzed by the combined team that involves the development and testing phases. The Sprint goal is defined through the combination of these phases. The testing scope is defined by the QA team where the whole team as well as the client validates and approves it. The work is initiated on the test case design by the QA team while the implementation of modules is carried out by the development team. Either within the testing tool or in the Excel spreadsheet, the generated modules are documented accurately. To review this, the document is forwarded to the development team and project sponsor. It is ensured here that the test coverage is completed to the utmost level. The testing is initiated on the QA environment by the QA team once the test case review and modification are covered. Within the defect tracking tool, the defects identified during testing are logged in a proper manner[2].

There are certainly important elements present within the effective agile systems which are mentioned below:

- The testers and QA managers are included as the members within the agile development team.
- Within the planning and requirements analysis, the testers are included as active contributors.
- The importance of testers is promoted and the continuous sharing of feedback is encouraged with the help of programmers and the customer.
- The main business flows are defined with the help of testers who actively participate within the meeting.
- Along with the developers, the testers work on the short iteration activities.
- The traceability amongst the requirements, test cases, and bugs are encouraged here.
- The story improvements of the user are encouraged by the testers [3].

- The specialized skills of test-driven development are leveraged which include unit testing, continuous integration as well as unit level.
- Regression testing is done with automation testing.

1.1. Agile Testing Disadvantages and Mitigations

The best testing methodology is proved by agile testing only when the requirements of the user are very clear to the sponsors of the project. The details are not clear if the overall requirements of the user are not clear. A path is taken based on initial requirements for the new products to design the software architecture. The following conditions occur if the requirements change with frequent time duration:

- As the significant effort has already been mentioned within the initial requirements development and testing process, the team struggles to adapt to the next changes.
- Within the estimations and sizing requirements, various challenges occur. As the QA is logically the last task for completing the story, it gets short shrift. Thus, the QA timelines get affected due to the delay in the prior development task.
- A test case for the complete iteration is prevented from being executed by the QA which leaves the team to struggle to complete the tasks assigned.
- The right questions are not asked which is very harmful to the QA. There should be regular team meetings to avoid such kind of issue to arise in the future.

1.2. Agile Testing Strategies

There are four different stages involved within the life cycle span of agile testing. These stages are shown in fig1 and explained below:

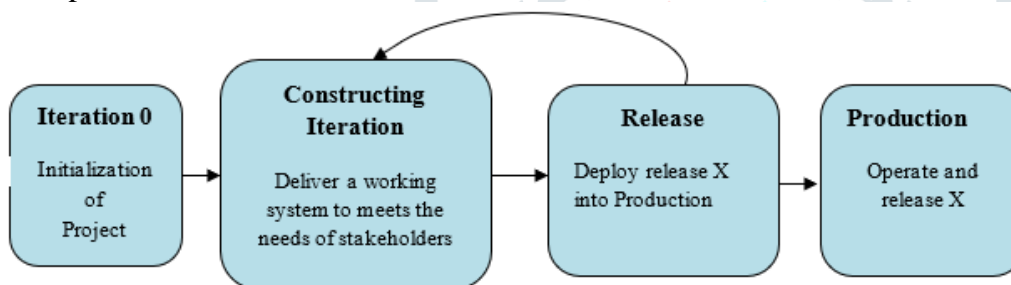


Fig. 1 Agile Testing Strategy

1.2.1. Iteration 0

The initial setup tasks are performed within the initial stage or iteration 0. The people are recognized here to perform testing, to install the testing tools, schedule the resources and so on. To achieve Iteration 0 [4], the following steps are followed :

- Establish a business case for the project.
- Establish boundary conditions.
- Candidate architectures are outlined.
- Identification of risk.
- The estimated cost is accessed and the initial project is prepared.

1.2.2. Construction Iterations

The construction iteration is the secondary phase of testing in which the majority of testing is performed. To build an increment of the solution, a set of iterations is performed within this phase. A hybrid of practices is implemented by the team to do so. The prioritized requirement practice is followed by the agile team during construction iteration. The most important requirements which are remaining from the work item stack are gathered from each iteration here and then implemented. There are two classifications of the construction iteration which are confirmatory testing and investigative testing. A verification related to the fulfillment of intent of the stakeholders

as described to the team to date is done within the confirmatory testing. The team performs this type of testing[5]. The issues that are ignored by the confirmatory team are detected by investigative testing. The potential problems are identified by the testers in the form of defect stories by the investigative testing. Common issues such as integration testing, load testing, and security testing are handled within the investigative testing only.

1.2.3. Release End Game or Transition Phase

The system is deployed successfully into production with the help of an objective which is “Release, End Game”. The training of end users, support people and orientation people are the three phases that are included within the activities. The marketing of product release, back-up and restoration, finalization of system and user documentation are also involved within this method. Complete system testing and acceptance testing is involved within final testing stage. The product needs to be tested more frequently within its construction iterations for finishing the final testing stage without many problems. The testers will work on the defect stories identified at the end stage [6].

1.2.4. Production

The product is forwarded to the production stage once it is dispatched from the release stage in this process.

1.3. Effective management of a Test project

Even before the testing is performed, any type of risk or error is recognized when effective planning techniques are implemented. A different mention is necessary even when the tasks are interrelated to each other.

1.4. Allocation and distribution of tasks

Within the planning stage, the effort estimation and sizing are included from which the need for these tasks is identified. There are some pointers used to allocate the tasks while keeping in check the size, schedule, properties, and skills of the team. They include:

- i. Making the team members gather around and communicating the objectives to the members is very beneficial. The particular task is specified in proper rationales to the members. A meaningful method for accomplishing the tasks is required when the fitting of work within the overall scheme of things is known. This will help in deriving a meaningful method to accomplish these tasks [7].
- ii. A skill cart of all the team members is prepared and the tasks are assigned to them accordingly. Even the junior team members are given a fair chance. For instance, in any case, there is a huge task given, the task can be divided amongst the senior and junior members of the team. The leadership qualities within the senior member can be enabled through this and will also improve the knowledge of a junior team member. More than one senior member can provide help if any task is highly complex and cannot be handled by one member.
- iii. The design after being complete is forwarded to the complete team to receive a review. The motive and logic which is behind the assignment are explained here. The individual can be checked before finalizing the assignment whether they have any issues with it or they need any suggestions to make. The schedule is to be discussed in case there are any objections and the type of status report that is required from their end is also presented here. If there is any feedback given it also shows that the allocations are not satisfactory for the user. The opinions of individuals are to be discussed and their opinions are to be respected. The options for re-balancing the assignment are also checked if the concerns expressed are valid. This helps in ensuring that the task will be completed on time and there will be no risk.
- iv. The mode of reporting which is expected is to be agreed upon here. Any kind of mode which the user is comfortable with is to be selected here. The user must feel free to share any kind of issues arising and the team must help them in solving those issues so that the goal can be achieved.
- v. The required tasks are tracked by identifying the required tools. The tasks that are not yet worked upon can be reviewed after a few time durations here along with the tasks that are in progress and the ones that have been achieved successfully.

1.5. The Agile Testing Quadrants

The complete process is divided into four quadrants which are known as the agile testing quadrants as shown in figure 2. This helps in understanding how agile testing can be performed [8].

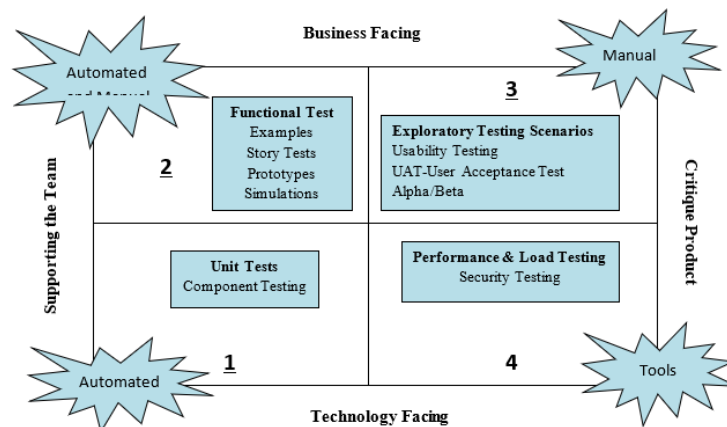


Fig. 2 Quadrants of Agile Testing

1.5.1. Agile Quadrant I: The major focus of this quadrant is on the internal code quality. There are various technology-driven test cases involved here that are implemented to support the team. They are:

- Unit Tests
- Component Tests

1.5.2. Agile Quadrant II: The test cases which are business-driven are involved within this quadrant. They support the team and help in focusing on the requirements involved [9]. The various types of tests performed within this phase are:

- The examples of possible scenarios and workflows are tested.
- The user experience such as prototypes is tested.
- Pair testing.

1.5.3. Agile Quadrant III: The feedback of quadrants one and two are provided by this quadrant. To perform automation testing, test cases can be utilized here as the base. Various rounds of iteration reviews are carried out in this quadrant which helps in building confidence within the product being designed. The different types of testing performed in this quadrant are:

- Usability Testing
- Exploratory Testing
- Pair testing with customers
- Collaborative testing
- User acceptance testing

1.5.4. Agile Quadrant IV: The non-functional requirements such as performance, security, and so on are handled within this quadrant. The non-functional qualities and the expected value is delivered by the application through this quadrant.

- Non-functional tests like stress and performance testing.
- Security testing based on authentication and hacking.
- Infrastructure testing
- Data migration testing
- Scalability testing
- Load testing

1.6. Challenges with agile software development

Various challenges are being faced during the development of agile software [10]. Some of these challenges are:

- In agile, the possibilities of occurrence of an error are more as there is less priority given to the documentation and high pressure is imposed on the QA team.
- There is an introduction of novel features which help in minimizing the available time for the test teams which helps in recognizing whether the recently added features match the requirement or not and do they address the business suits or not.
- A semi-developer role is played with the help of testers.
- There is a huge compression of test execution cycles.
- The test plan is prepared in very little time.
- Minimal timing is provided for regression testing.

1.7. Risk of Automation in Agile Process

Various risks arise within the automation process of agile testing. The list below mentions some of these risks:

- A high level of confidence is provided by the automated UI which is however slow to be executed, fragile in order of maintenance and very costly. Unless the testers know the complete procedure of testing, the automation might not be able to enhance the test productivity.
- Within automated testing, unreliable tests are very important to be considered. It should be on a higher priority to fix the failing tests and resolve the problems relevant to the brittle tests. This avoids false positives within the system.
- There are many simple features provided by various commercially available automation tools. The manual test cases are also required to be replayed here.
- While performing manual testing, there are various test set up and tear down procedures that are missed out during test automation which make is seamless to test the setup and tear down procedures .
- The testing solutions which need huge maintenance about the value provided are proposed and delivered by automation.

2. Literature Review

Swadha Gupta et.al (2017) presented in this paper that there are still various applications in which the method of agile is not appropriately applied to compute complex software projects involved within them. The Agile manifesto is applied along with Agile Software Development to provide quick modifications. This will step towards the success of agile adaptation on large products to develop the software[11].

Manuel Acosta, et.al (2017) proposed in this paper a robust virtual sensor that uses a type-model-less approach for estimating the chassis planar motion stated and tire forces during agile maneuvers. All such enhancements help in providing improvement in the traditional approaches and provide a better system that can perform better testing in terms of various aspects [12].

James B. Dabney, et.al (2017) investigated in this paper the previous techniques of agile testing involved. Further, the various enhancements required within particular techniques are also identified along with the utilization of agile frameworks for accommodating the development of huge systems. All such investigations and modifications proposed in this paper help in making enhanced testing systems and provide better results [13].

Khush Bakhat Awar, et.al (2017) presented in this paper that numerous issues are being faced by the distributed teams when they utilize agile systems within their environment. The agile practices which are utilized by the distributed systems are studied in detail and are presented within the latest studies as well. An empirically-based investigation is proposed in this paper which provides a detailed study related to the important factors present within the agile distribution method. Further studies relied on the results achieved from the proposed work. The Scrim and XP methods were tailored to apply the agile practices within the distributed scenarios in a better way and to avoid various issues [14].

Narjes T. Jahromi, et.al (2017) presented in this paper that there is a need to provide value-added video services as there is a huge growth in demand for these services from the user's end. Within the content delivery networks (CDNs) a cost-efficient and agile technique are to be presented which might help in providing an enhancement in the previous existing approaches. The performance is evaluated here in terms of various aspects and it is seen that there is a huge improvement within these systems to provide enhanced mechanisms for testing [15].

Olivier Liechti, et.al (2017) presented in this paper that the important aspects of agile methods involve continuous improvement, feedback mechanisms, and automated testing. There are three practices involved within the test analytics which are presented here. The concept is explained here with the industrial case study and the experiments are conducted by the team. A simple and effective manner is presented in this paper which helps in visualizing the progress achieved by the team. To initiate the test analytics platform, there are numerous experiments conducted [16].

3. Research Methodology

The technique of job scheduler will be applied in which the white and black box testing will be applied according to schedule as shown in figure 3. As agile testing consists of all types of testing and these testing methods are classified as white-box and black-box. The testing which you applying will be scheduled and errors are detected from the software. To implement the project, a case study of an online shopping website will be considered. In this case study, various modules of the website are considered and their test cases. Agile testing will be applied to each module of the website with its developed modules. This will help out to detect a maximum number of errors or bugs from the project.

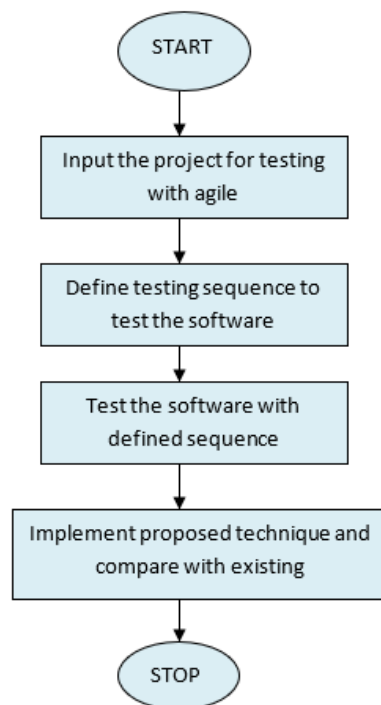


Fig 3: - Proposed Flowchart

4. Expected Outcomes

Following are the various expected outcomes of this research:

1. The proposed improvement can generate proper sequencing of testing in agile testing. This will increase the fault detection rate.
2. The correct sequence of testing in agile testing can reduce the execution time of fault prediction.

Conclusion

In this work, it has been concluded that agile testing is an efficient technique to detect a maximum number of faults from the software. Agile testing consists of various other types of testing like a black box, white box, automation and so on. The major challenge of agile testing is to define the exact testing sequence. In this work, the algorithm which has the dynamic objective function will be proposed to define the testing sequence. The proposed improvement leads to an increase in the fault detection rate and reduces execution time.

References

- [1] “R. Phalnikar, V. S. Deshpande, and S. D. Joshi, ‘Applying agile principles for distributed software development,’ Proc. - Int. Conf. Adv. Comput. Control. ICACC 2009, pp. 535–539, 2009.”
- [2] “J. S. Park, P. E. McMahon, and B. Myburgh, ‘Scrum Powered by Essence,’ vol. 41, no. 1, pp. 1–19, 2016.”
- [3] “M. Korkala and F. Maurer, ‘Waste identification as the means for improving communication in globally distributed agile software development,’ J. Syst. Softw., Apr. 2014.”
- [4] “R. Giuffrida and Y. Dittrich, ‘A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams,’ Inf. Softw. Technol., 2015.”
- [5] “M. Niazi, ‘Teaching global software engineering: experiences and lessons learned,’ IET Softw., pp. 1–8, 2015.”
- [6] “R. Giuffrida and Y. Dittrich, ‘A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams,’ Inf. Softw. Technol., vol. 63, no. December, pp. 11–30, 2015.”
- [7] “E. Bjarnason, K. Smolander, E. Engström, and P. Runeson, ‘A theory of distances in software engineering,’ Inf. Softw. Technol., no. May, 2015.”
- [8] “S. Farid, R. Ahmad, I. A. Niaz, M. Arif, S. Shamshirband, and M. D. Khattak, ‘Identification and prioritization of critical issues for the promotion of e-learning in Pakistan,’ Comput. Human Behav., vol. 51, pp. 161–171, 2015.”
- [9] “V. T. Heikkilä, M. Paasivaara, K. Rautiainen, C. Lassenius, and T. Toivola, ‘Operational release planning in large-scale Scrum with multiple stakeholders – A longitudinal case study at F-Secure Corporation,’ vol. 57, pp. 116–140, 2015.”
- [10] “E. Tüzün, B. Tekinerdogan, M. E. Kalender, and S. Bilgen, ‘Empirical Evaluation of a Decision Support Model for Adopting Software Product Line Engineering,’ Empir. Softw. Eng., vol. 60, pp. 77–101, 2014.”
- [11] “Swadha Gupta and Deepali Gouttam, ‘Towards Changing the Paradigm of Software Development in Software Industries: An Emergence of Agile Software Development’, 2017 IEEE International Conference on Smart Technologies and Management for Computing, Communicat.”
- [12] “Manuel Acosta, Stratis Kanarachos, and Michael E. Fitzpatrick, ‘Robust Virtual Sensing for Vehicle Agile Manoeuvring: A Tyre-model-less Approach’, 2017, IEEE.”
- [13] “James B. Dabney, James D. Arthur, ‘Applying Standard Independent Verification and Validation (IV&V) Techniques within an Agile Framework: Is there a Compatibility Issue?’, 2017, IEEE.”
- [14] “Khush Bakhat Awar, M. Shujah Islam Sameem, Yasir Hafeez, ‘A Model for Applying Agile Practices in Distributed Environment: A Case of Local Software Industry’, 2017 International Conference on Communication, Computing and Digital Systems (C-CODE).”
- [15] “Narjes T. Jahromi, Sami Yangui, Adel Larabi, Daniel Smith, Mohammad A. Salahuddin, Roch H., Glitho, Richard Brunner, Halima Elbiaze, ‘NFV and SDN-based Cost-efficient and Agile Value-added Video Services Provisioning in Content Delivery Networks’, 2017, I.”
- [16] “Olivier Liechti, Jacques Pasquier, Rodney Reis, ‘Supporting Agile Teams with a Test Analytics Platform: a Case Study’, 2017 IEEE/ACM 12th International Workshop on Automation of Software Testing.”