

# Issues in Scrum Agile Development Principles and Practices in Software Development

R. Vijay Anand\* and M. Dinakaran

School of Information Technology and Engineering, VIT University, Vellore - 632014, Tamil Nadu, India;  
vijayanand.r@vit.ac.in, dinakaran.m@vit.ac.in

## Abstract

**Background/Objectives:** Software Development assumes an exceptionally significant part in this modern world. In the late couple of years more and more software development organizations are endeavouring to adopt agile software development methods and techniques. Fruitful agile adoption leads to producing higher quality software, enhance developers moral and at a lower cost than the conventional waterfall model approach. This study seeks to evaluate, synthesize and present aspects of research on agile methods, approaches adopted and the criteria utilized for agile practice selection. **Methods/Statistical Analysis:** The method adopted was a Systematic Literature Review on studies published from 2010 to 2015. **Findings:** The agile scrum method has several technical and generic issues. Technical issues include assigning, prioritizing and integrating product backlog items. Generic issues include distributed environment issues, issues with idle team members and Developer-Tester problem. **Application/Improvements:** We identified the impacts of agile methodology on software development processes with respect to quality within the organizational, methodical, and cultural framework.

**Keywords:** Agile Software Development, Agile Software Testing, Product Backlog Item, Scrum

## 1. Introduction

Software Engineering is the platform to develop the software step by step and to produce a high quality product at the end. There are many models to implement Software Engineering. All the models are actually based on the Software Development Life Cycle (SDLC). The SDLC has phases or steps to be followed to achieve software development. These steps are customized to form a process model which will help the software engineers to follow a certain path. Some process models are Waterfall model, Spiral model, Prototyping, iterative and incremental model, rapid application development and agile model. Agile software development approach enables developing software in regular intervals, i.e., iterations, producing the software in increments<sup>1,2</sup>. This research concentrates on the major issues and challenges in Scrum method of agile model. Scrum is a process framework to deliver products with the highest possible value and handle complex problems or situations. Use iterative and incremental

approaches to develop products using cross-functional teams<sup>1,2</sup>. The issues are listed and described below:

## 2. Issues with Prioritizing the user Stories

### 2.1 Assigning Product Backlog Items

In scrum method Product Backlog Item is central. The PBI contains a prioritized list of all items relevant to a specific product. Bugs, customer requested enhancements, competitive product functionality, competitive edge functionality and technology upgrades are available in this list<sup>3</sup>. The first major issue is to convert user stories into Product Backlog Items. There are lots of confusion in assigning PBIs from the Software Requirement Specification document. There are cases in reality wherein either PBI are assigned directly from the functionalities of the software that is requested by the client or the Scrum team assigning the PBIs or in some cases PBIs are

\*Author for correspondence

assigned based on the sprint duration<sup>4,5</sup>. These cases are not the right method to assign PBI, because the functionality of software can be a PBI but not all functionalities are PBIs. A scrum team cannot assign PBI since the team does not have insights on the project, only the Product Owner and Scrum Master are responsible for assigning PBI. The assigning of PBI is the first task of Scrum, if that is not properly assigned then the whole project leads to failure or inappropriate delivery of increments.

### 2.1.1 Prioritization of PBI

In Scrum, the user-stories are converted into Product Backlog Items (PBI). These PBIs have to be prioritized so that it will benefit both the client and the company. Prioritization of user-stories is a difficult task in agile environment because of constant changes in the user-stories. PBIs have to be prioritized with respect to some characteristics. The client and the company have their own priorities for the PBIs<sup>4,5</sup>. The method for prioritization of the PBI by the company for instance may be based on the criticality of the PBI, type of scrum team available to work on a particular PBI or which PBI will have a better software reuse. The client priorities are based on the business value of the PBI and the need for a PBI. These two different priorities might not be the same, so in that case there is a need to adopt a prioritization technique which values and also addresses the characteristics of both the client's and the company's priorities. Such a technique is not available and is a major issue in Scrum.

## 2.2 Traditional Agile Prioritization Methods

The traditional methods used to prioritize user stories are discussed<sup>4,5</sup> in Table 1.

## 3. Distributed Environment

In a distributed scrum, spontaneous flow of communication gets compromised having no feasibility of sitting in the same work-room. Communication factors like Distance, Time Difference, Cultural Difference, Language, etc<sup>6,7</sup>, keeps on generating issues every day. There is no proper mechanism to communicate between Scrum teams placed in different geographical locations during the Sprint planning and analyzing phases. Team cannot be put together under a single roof in the current scenario. Pairing of off-shore member and onsite member can solve this problem but the problem is, it's kind of a mentor-mentee relationship which can be done only between two members. This issue is faced by most of the software companies which are implementing Scrum. "Productivity Theft" is another challenge particularly for remote team which means engagement of members into several "beyond scrum" activities without knowledge of Scrum Master sometimes by offshore management and sometimes by even onsite members. The Scrum teams have to involve in various meetings like Scrum Planning Meeting, Scrum Review Meeting, Scrum Retrospective Meeting, etc. The Scrum team members have to participate in these meetings in



**Table 1.** Prioritization Techniques

S. No	Prioritization Methods	Definition	Problem
1.	Validated Learning	The features having high risk are developed first and are then released in the market. The feedback is then collected and the learning is applied onto the new feature. <sup>5</sup>	The problem is that here is a dependency between the user stories and therefore division only on the basis of risk is a difficult task.
2.	Walking Skeleton	The requirements are selected such that the features are built within a short span of time with minimal care. <sup>5</sup>	The problem in this method is that the client requires the product according to his/her ease, not as selected by the development team.
3.	MoSCoW	The PBIs are prioritized based on M – must have this, S – should have this if at all possible, C – could have this if it does not affect anything else, and W – won't have this time but would like in the future. <sup>5</sup>	The problem is that assigning priority in this manner is a difficult task.
4.	Business value based	Each requirement carries a business value which it can potentially generate to the client. <sup>5</sup>	The problem is that this method doesn't consider dependency factor.

every sprint cycle. Teams in various geographical areas are the ones affected by this issue. There is no proper or practical solution for this issue.

## 4. Regression Testing

Regression testing is testing of software after it has been modified, to check whether the existing functionality gets affected or not. It is one of the most expensive activities that occur as the software is developed and maintained. Studies shows that a significant portion of development and maintenance costs go to this regression testing, which is known as *retesting*. Reports state that regression testing uses 80% of the overall testing cost and can use up to 50% of the cost of software maintenance<sup>8,9</sup>. Rapidly changing software and computing environments present many challenges for effective and efficient regression testing in practice. Regression testing can be performed after the changes have been made to the software, before the new version of the software is released every time the software is saved and compiled. In an agile development environment or before patches, such as security patches are released. The goals of regression testing are to improve the confidence that the changes behave as required and that they have not affected the previous parts of the software. Since Scrum is about adapting changes of the software that is being developed, it is a must to test the software with regression test after the changes have been made. The regression testing is implemented on the every integration of PBI for the final delivery. It is important to develop PBI in such a way that it does not create lots of bugs on integration with other PBI. But in reality it is highly unlikely in cases of complex projects, which subsequently increase the cost and time of regression test, which in turn directly affects the agile delivery of software because of increased time taken to solve bugs to provide high quality software. It is a major issue among other issues as it directly affects the cost and time. There is no proper solution for this problem.

## 5. Integration of PBI

In Scrum method of agile methodology, after the completion of the development of PBI, it will be delivered to the client as an increment. After completion and delivery of all the PBIs, the Scrum team has to integrate each and every PBI. Here the significant issue is that, the analyzers need to perform relapse testing for every last

combination, which builds the testing cost and time. More than 80% of the expense of testing is utilized just for relapse testing<sup>8,9</sup>. No proper model or method is currently available for “how to integrate PBIs with reduced regression testing”.

## 6. Idle Team Members

During the identification and analysis of PBIs, the other team members like developers, testers and maintenance providers will be idle and it is a waste of time for the team. As in agile, there should not be any wastage in time otherwise it will affect the delivery of the PBI.

## 7. The Developer-Tester Problem

In Scrum method, for a particular sprint the starting time and time to complete the PBI will be planned and allocated. The developers and testers have to perform according to that time. Suppose if developer couldn't complete a PBI in a given time then it will also affect the testers because the testers cannot start testing before the completion by developers<sup>9,10</sup>. This results in problem between developers and testers and there is no proper technical or generic solution for this problem yet.

## 8. Definition of Done

In Scrum method, the testers basically have issues when a particular sprint for the PBI is done. They will be doing regression testing again and again and would not get a clue when to stop a particular sprint. Even the developers have this issue sometimes when they do not know when to stop the code and deliver the increment to client<sup>10</sup>. The product owner and Scrum master should monitor the situation and guide the sprint team but there are cases even the Scrum master cannot properly identify when to disclose a sprint and deliver.

## 9. Testing Issues in Scrum

Scrum life cycles are becoming common and every life cycle affects testing. Testing is done in each sprint cycle, when the PBI is developed it is immediately tested and released to client and when all the PBIs are developed and delivered, the final step is to integrate all the PBI and a final testing is implemented. The quantity of blunders<sup>10</sup> is likewise subject to the individual or the group of persons

building up the product. In the event that the designers are experienced there are lesser possibilities of blunders. There may be special cases to this. Most of the issues in Scrum are the issues which are related to testing.

In Table 2 some of the testing Issues in Scrum<sup>11</sup>.

### 9.1 Other Common Issues in Scrum

Other common issues in Scrum are discussed in Table 3.

**Table 2.** Testing Issues in Scrum

S.No	Testing Issue	Statement
1.	Volume and Speed of change	Welcoming the changing requirements even at the late development stage is one of the advantages and also a problem in Scrum. A late change increases the regression test cases and also to complete the project on time the speed of testing has to be increased. Thus it directly affects volume and speed of test.
2.	Inconsistent and Inadequate unit testing	Since Scrum releases products as increments, each increment has to be tested and it is tested by Unit testing. But Unit testing has limited bug finding effectiveness (25% to 30%) as compared to System testing (85%). Thus causing a problem related to the software quality.
3.	Many Meetings	Scrum focuses on working software and not on comprehensive documentation. This involves more meetings and excessive meetings can happen in any life cycle. The major problem is to have balance between documentation and meetings on every project and life cycle.

**Table 3.** Other Common Issues in Scrum

S.No	ISSUE/PROBLEM	STATEMENT
1.	Problem Solving in Daily Scrum	The Daily Scrum should not be used to find solutions to problems. It is used to keep the meeting short and have those problem solving conversations with only the intended team members. The Scrum Master should facilitate the Daily Scrum.
2.	Assigning Tasks	Even though the concept of self-organizing teams has been around for a long time, the tasks for every member in the team should be assigned properly but in reality it is highly unlikely. The Scrum Master should follow certain constraints for better results.
3.	Failed Sprint Restart	Cancelling of sprint is very rare in Scrum. Scrum team has to wait until everything is “perfect” and “ready” before restarting. The team should immediately restart after cancelling a sprint.
4.	Product Owner Doesn't Show	The Product Owner (PO) is the authorized member from the client side. It is very important that the PO is present during the sprints, so that the team can clarify doubts and acceptance testing can be performed easily after completion of each milestone.
5.	Individual Heroics	Individual team member in Scrum should not do over excessive work or over time in Scrum so that to become a great hero in a team. It will directly affect the other team members and also the schedules. A whole team can do excessive work but not an individual. Scrum should build a “great teams of people, not teams of great people”.
6.	Team organizes Product Backlog	The team does not have direct insight to client needs and instead should be focused on the problem solving. The Scrum Master and the Product Owner are the ones who organize and prioritize the Product Backlogs.
7.	Changing Team Membership	If the membership of the team changes, then it will force the team to restart the Forming-Storming-Norming-Performing process. If the team is in Norming or in Performing state, then any reason to change is a waste of investment.

## 10. Conclusion

These are the major issues and problems faced by software companies when they implement Scrum method of Agile. These issues do not have any proper technical or generic solutions to solve. Each software company is trying and implementing some methods which they believe will solve these problems on a temporary basis. But these customized method does not lead in the right path always.

## 11. References

1. Campanelli AS, Parreiras FS. Agile methods tailoring – A systematic literature review 2015. *The Journal of Systems and Software*. 2015 Dec; 110:85–100.
2. Othmane LB, Angin P, Weffers H, Bhargava B. Extending the agile development process to develop acceptably secure software. *Dependable and Secure Computing*. IEEE Transactions. 2014 Nov-Dec; 11(6):497–509. DOI:10.1109/TDSC.2014.2298011.
3. Vlaanderen K, Jansen S, Brinkkemper S, Jaspers E. The agile requirements refinery: Applying SCRUM principles to software product management. *Information and Software Technology*. 2011 Jan; 53(1):58–70.
4. Danevaa M, Veena EVD, Amrita C, Ghaisasb S, Sikkela K, Kumarb R, Ajmerib N, Ramteerthkarb U, Wieringaa R. Agile requirements prioritization in large-scale outsourced system projects: An empirical study. *The Journal of Systems and Software*. 2013 May; 86(5):1333–53.
5. Popli R, Chauhan N, Sharma H. Prioritizing user stories in Agile Environment. *International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*. 2014 Feb 7-8. p. 515–19. DOI:10.1109/ICICT.2014.6781336.
6. Shrivastava SV, Rathod U1. Categorization of risk factors for distributed agile projects. *Information and Software Technology*. 2015 Feb; 58:373–87.
7. Ghosh GK. Challenges in Distributed Scrum. *IEEE Seventh International Conference on Global Software Engineering*. 2012. p. 200. DOI: 10.1109/ICGSE.2012.46.
8. Parsons D, Susnjak T, Lange M. Influences on regression testing strategies in agile software development environments. *Software Qual J*. 2014 Dec; 22(4):717–39. DOI: 10.1007/s11219-013-9225.
9. Anita, Chauhan N. A Regression test selection technique by optimizing user stories in an agile development. *IEEE International Advance Computing Conference (IACC)*. 2014 Feb 21-22; 1454–58. DOI:10.1109/IAdCC.2014.6779540.
10. Ekbal R, Srikanta P, Vandana B. Estimation and Evaluation of Change in Software Quality at a Particular Stage of Software Development. *Indian Journal of Science and Technology*. 2013 Oct; 6(10).
11. Akif R, Majeed H. Issues and challenges in Scrum implementation. *International Journal of Scientific and Engineering Research*. 2012 Aug; 3(8):1–4.1. ISSN 2229-5518.