

Combining Agile Model with Lean Principles to Improve Scrum Model Methodology in Software Companies

Gul Naz and Nayyar Iqbal

Department of Computer Science, University of Agriculture, Faisalabad, Pakistan

Copyright © 2016 ISSR Journals. This is an open access article distributed under the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT: In this research a process model is proposed for software companies to overcome scrum process issues with the combination of agile model and appropriate lean principles without compromising its benefits and evaluate its success in five different software development companies. Scrum is widely used agile development method that put prominence on the cross functional and self-organized teams working in small items called sprints to regularly construct entire increment of product. It heavily depends upon the people and project due to flexibility in requirement change. Moreover, Software development managers, project managers and team lead engineers unsure how to apply agile methods with lean principles, how to minimized developer and team management overload and how to promote interest in team members. Many organization fail in getting success and proper benefits from scrum as it is difficult to handle large and complex project that involve weak value flow, team level and multiple team level issues. In software development to much developer overload due to extra features makes the project heavy weight. No proper model is yet introduced which implement lean principles in Agile model, If proper eliminate waste principles are used in scrum process it will provide better and light weight improved process.

KEYWORDS: ASD, Lean Software development, Traceability, Agile methods, Project Management, Scrum.

1 INTRODUCTION

Software development process is a controllable approach in software engineering that facilitates software team to deliver quality software in timely manner and satisfy the stakeholders and customer by selecting the appropriate task, actions and set of work activities. Each of these activates collection of work and actions exist in a model or framework that identifies their association between them and with the process. Process model organized the project into activities help the team and project manager to work in a proper sequence [1].

Agile methods have been proven to be valuable in small software teams. However, it has been found to be a challenge to achieve same value in large scale software development teams. In mission critical projects Lean thinking has been considered as enterprise perspective way to surpasses limitation in scaling ASD. Lean ideas appeared in the manufacturing and started in industry and considered key success driver of Japanese industry [2].

Quality of software depends upon the process of software development and process considered to be successful if it is being completed according to schedule and fulfill the customer requirements. Software development is always associated with uncertainty due to dynamic requirements. Agile methodologies is flexible approach and suitable to handle dynamic requirements. No significant empirical evidences are reported on the successful adaptation of agile methods. Scrum is mostly used in software companies it provide quality and cost but there are some issues and challenges while companies are trying to adopt and transformation from plan driven approach to agile. Lean works with value streams and provide management with high progress visibility. An inclusive development cycle accomplishing all the lean principle with agile combination, requirement tractability has not been proposed yet. This research will formulate a model with a hybrid approach of lean and scrum process. The proposed model consists on the phases of agile model with scrum process.

2 BACKGROUND

Wang conducted secondary data analysis to examine how agile and lean methodologies have been combined in context of software development process and categorized six approaches of combine use of both methodologies that are non-purposeful, using Lean to collaborate with other business areas while maintaining Agile, combination of Agile practices and Lean principles and tool, applying lean and agile development principles together, agile practice with lean techniques to improve software development. It has been not stated proper model for combination of Agile. However, It has stated that the combination types and success of software development is depends upon the needs and situations of the organizations [3].

Balachander *et al.* described the implementation of lean concept and flow on agile software development. In an industrial web based project that was executed by agile development experienced team. It has been conducted a case study to find how lean concept implemented on agile software development methodology. It described the important factors that are limited to the project the boundaries within which these can be used, doing valuable work with the high degree of collaboration of team member and customers. In a production-ready state test and staging production environments has been providing the handiness to the team to always retain the system. It significantly reducing the risk typically associated with progress to production. Highly skilled team, Coding standards and engineering approaches also contributed greatly towards this [4].

Ali [5] conducted an online survey in Pakistan to find out the challenges and benefits of agile software development. The work identified that lack of experience of scrum master team leader and the project manager are the biggest issue. There is no consideration on the training as most of the companies are recently shifted from traditional to agile development. Moreover it is difficult to handle complexity of project with small teams. It has been examined that productivity, customer satisfaction, time to market and quality are advantages of the agile methodology.

Ahmad *et al.*, (2013) observed the results from a literature survey and identified the obtained benefits and related challenges to adopt Kanban in agile software development. It has explored new research opportunities for the researchers. This paper has identified a greater number of studies on Kanban practice in software development. The recent publication trend indicates that Kanban is gaining trust in the software engineering domain. A high percentage of the primary studies appear to be experience reports (47%); and most of the studies involved small-scale projects, to the Kanban approach. Based on this, software engineering has been demanded for more broad and precise scientific research on Kanban [6].

Jonsson *et al.*, (2013) observed from a literature survey that lean thinking could be transformed to a framework. This framework could be providing difference between lean concept and agile method and provide guidelines how to apply lean concept in software development process. Before this study lean and agile often use as interchangeable terms but it has been claimed that both terms are different. This proposed framework has a potential to complement the view of other researchers and show to the practitioners what lean software development in reality is. Although the framework is developed in a traceable and extensible manner but validation is necessary for properly use [7].

Rodríguez *et al.*, (2014) analyzed the transformation to Lean in the background of software development at Ericsson R&D Finland. Authors have explored how Lean principles in software development are put into practice and the challenges that commence when implemented Lean. Two major contributions of the research is to analyzed main factors that helped Ericsson R&D's achievements ,identified elements such as continues integration, communities of practice, network of product owners etc and categorized challenges in implementation of Lean Software Development. This study has been provided evidences of compatibilities between Lean and Agile [8].

2.1 AGILE MODEL

Agile practices are a subset of evolutionary and iterative processes and are based on iterative improvement. In all iterative artifacts, iteration is a self-contained, small project with mini activities that cover requirements analysis, design, implementation, and test. Agile software development methods were developed as remedy for the heavy weight methods' weaknesses Plan-driven methods or heavy weight process are begun with the supplication and documentation of a set of requirements that is as complete as achievable. [9].

A group of proficient's in a collective nature introduced ASD process as a solution to overcome issues of traditional methodologies but Agile manifesto provides the proper and detailed introduction of Agile process to the software industry. The agile manifesto is based on four basic norms: "Interactions and Individuals over tools and processes", "Working software with complete documentation", "Collaboration of customer over agreement negotiation", and "Responding to change according to plan" [10].

The customers were becoming less able to state their needs properly due to rapid change in software technologies, software industry and customers' expectations. As a result, agile methodologies emerged as an endeavor to more properly embrace higher rates of requirements change. The Agile Manifesto contains twelve underlying principles, namely:

- i. Rapid delivery for satisfaction of Customer
- ii. Always ready for changing requirements
- iii. Frequently delivered a working software
- iv. Always considered as principal measure of success to working software
- v. To maintain a constant rate, considered sustainable development
- vi. Meetings are require between stakeholder and developers for daily cooperation
- vii. Face to face conversation
- viii. Motivated individuals and trusted build the projects
- ix. Technical excellence by good design and Continuous attention
- x. Simplicity
- xi. Self organizing teams
- xii. Regular adjustment to changing situations

2.2 SCRUM

Software development procedure implements scrum technique for managing the software development projects. The main purpose of Scrum is to develop a flexible product; therefore it is competent to early respond to changes required by customers. Figure 1 represents the Scrum life cycle. The product in scrum is described in form of user stories, which depicts functionality from a customer viewpoint. The development is being carried along time boxes or incremental tasks, called sprints (iterations). Each sprint defined a sprint objective during planning meeting, including the user stories that being assumed during distinct sprint. The sprint backlog that has been selected from the product backlog is being fixed during the sprint. After every sprint, two defined meetings are conducted to review project current state, and exposition meeting to reveals about the method of working. Three main roles in Scrum are Product Owner, Scrum Team and SM (Scrum Master) [11].

The scrum lifecycle consist on the following simple phases. First phase is planning in this phase expectation and vision is established and funding for the project is secured,

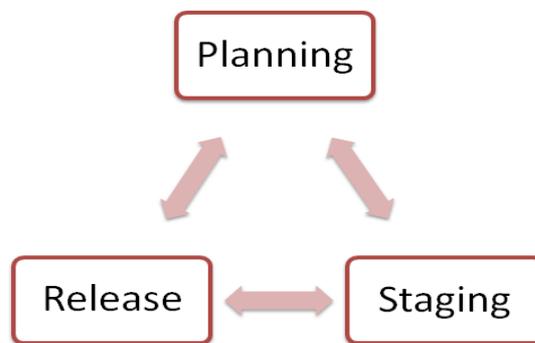


Figure 1: Scrum life Cycle

second phase is Staging this stage requirement are prioritized for first sprint(iteration) development and implement a workable product, last phase is release in which product is deliver as shown in fig.1.

2.3 APPLICATION OF LEAN THINKING IN SOFTWARE DEVELOPMENT PROCESS

Mary and Tom Poppendieck saw the application of the lean thinking in 2007 at work in a Toyota production system. After they saw the advantages of the Lean Principles, they published a book named Lean Software Development: An Agile Toolkit which direction to the application of Lean Principles in Software Development process. Recently Poppendieck and Cusumano (2012) have described seven lean principles for software development which is based on succession experiences. Authors proposed thinking lean as a set of principles comparatively thinking a set of practices [12].

They applying lean principles in soft-ware development process to make more sense and increase quality of product. The lean principles for software development are:

- Elimination of waste
- Enhancement of Quality
- Creation of the knowledge
- Commitment Deferment
- To deliver the product fast
- People should be respected
- Optimization the whole

Those features that have no value for customer elimination of these feature is called Elimination of waste. Waste is the enemy in a system and Value is the basis of any important product, and is specified by the customer. Value means what the customer wants the product to do.

Enhancement of quality means built quality along the development phases instead after the completion of development stages. Creation of the knowledge means over all software development method is like to be a knowledge providing process, the deferment commitment means there should be no hurry and decisions should not be made as late as possible [13]. To deliver the product fast principle is important for customers and averts to change their mind instantly. The optimization of the whole means work should continue constantly as long no explodes in development directs to success. Different authors were suggested different practices for software development to attain the lean principles are listed as following Just-in-time (JIT), Kanban, Kaizen, Make everything visible, Cost of delay, Value stream mapping and Queuing Theory[14]. (Womack and Jones, 2003) shown in Fig.2.

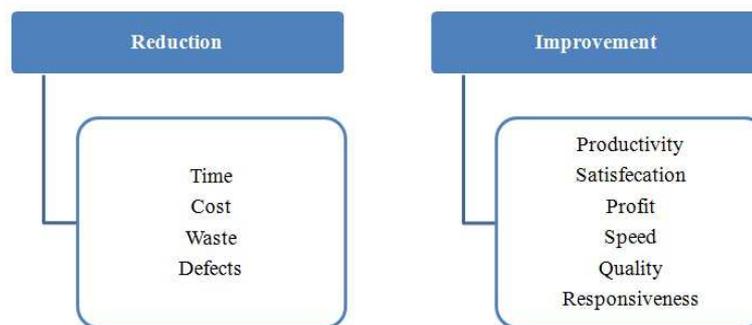


Figure 2: Lean principles benefits in Lean Scrum Software development Model

3 LEAN SCRUM SOFTWARE DEVELOPMENT MODEL

The framework is consisted on the quality assurance steps including lean practices. The steps Included: i) Current Project Initiation, ii) Set Quantifiable Goals and Mission, iii) Prioritized the Goals, iv) Choosing process model, v) Execute Process, vi) Analyze Process and perform development is the main task for lean development process. Metrics those are included in LSSD Model are: firstly Requirement specifications, Change Requests, remove burdens from developer and reduce defects, wastes control through traceability all have been included. Fig.3 shows proposed model.

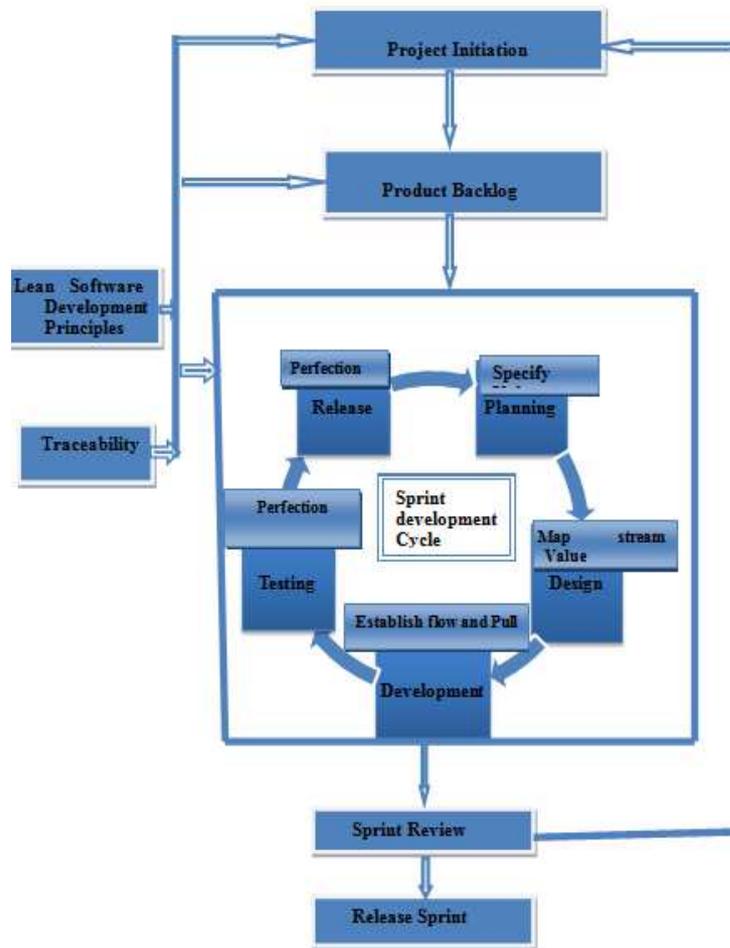


Figure 3: Lean Scrum Software development Model

3.1 PROJECT INITIATION

Project initiation activity consists of all the activities which are necessary to be carried out prior to the start of development, such as:

- Create Project Vision
- Initial Infrastructure
- Initial requirement
- Identify Scrum Master and Stakeholder
- Form Scrum Team
- Develop Epic Release Planning (Release Planning Schedule)
- Create Prioritized Product Backlog

3.2 PRODUCT BACKLOG

The product backlog is most important artifact and structured list of everything that may be required in the product development and is the main source of requirements for any changes to be made. The things those have been identified in this phase of development cycle are:

- Detailed analysis document, which summarizes requirement for product.
- It will be breaks the big-picture story into Product Backlog Items that are manageable increments of work
- Product Owner Prioritized the sprint that which sprint is crucial to implement first
- Requirement Traceability

3.3 SPRINT PLANNING

The lean principle specifies value is used to select a sprint on the basis of value given by the customer. Every sprint will be started with planning meeting which reviews stories from the Product which will be moved to next. Team will be independent to decide how and what the work is performed. All lean principles are used in sprint planning. The Product Owner can terminate a Sprint, which shouldn't occur often, and would usually happen due to a rapid change in business requirements. The Product Owner supports developers to focus on instant goals, and gives customers a tangible based view of progress.

3.4 SPRINT DESIGN

A Design Sprint phase provides flexible product design structure that serves to exploit the chances of doing something people want. It is concentrated effort conducted by a tiny team, A Sprint Master identifies the design challenge, brings the team together and gets a holds through all sprint stages. A Sprint design phase decreases the risk of downstream errors and generates vision-led goals.

The Sprint Design Phase consists of five discrete activities:

- Understand background and user insights
- Brainstorm what's possible
- Rank solutions, pick one
- Create a minimum viable concept
- RCC rule has been working to control the change, there are three steps have been included in requirement change control principle. Those consist of: Planning, Trying and checking; either the change is applicable or not.

3.5 SPRINT CODING

A sprint code phase is getting developers for a specific amount of time for just writing code. Eliminate waste principle is used here to minimized developer overload. Developer will learn from each other's and not focus on instruction. During coding they follow the eliminate principle to reduce time.

3.6 SPRINT TESTING

In this step of development cycle the important points to remember are the following: Design of test Cases (There will be one test case for each Requirement, every requirement will have its individual test case design for particularly to check that requirement and with its implementation.). The following are the test cases have to develop for the requirements.

- Unit Test
- Acceptance Test
- Integration Test
- System Test
- Function Test

Where unit testing result only like binary, either success or failure means yes or no. Like unit testing the acceptance test suggests also what is actually the reason of failure and applicable things can be also tell. Integration is also called definition of action, the testing in which both parts (software and hardware) are combined and comfort either that they both can relate and cooperate with each other according to customer expectations and requirements speciation's to give the user the required output. Software that must be workable frequently is one of the principles of the lean development that it is for delivering. Which means that the small pieces having functionality combined together and to make a set of them for delivery of system. Function testing is performed to check the functionality either it was the desired output or not.

- To handle change either request to change are approved or rejected.
- Removing of errors and failures Through customer involvement

3.7 SPRINT RELEASE

In this step of development cycle the product being developed fully after iterations and provided to customer after proper verification and validation to the customer. Final activities that are performed in this stage are lesson learned and checkpoint.

3.8 DEPLOYMENT

After all sprints, the software will be deploying for the customer in the last step. The proposed model has much capacity to agree to implement new requirements or change the existing requirements based on the customer demand integrated and the entire software will be deployed to the customer.

3.9 TRACEABILITY

Requirements traceability refers to flow from origins of change requirement through its development and specification, consequent deployment and use, and throughout all periods of current refinement and sprint in any of these phases. Not all projects needs to have traceability but tracing is always useful for some project like large scale and mission critical. Traceability provides the following advantages in proposed software development model.

- More transparency, it will be easier for anyone outside of the team to track the progress.
- Reduce risk of losing information.
- Could result in minimum time searching for bugs in big systems, the team could easily locate where to start looking for errors and what tests to fix if they know what constraint that is broken.
- Possible to observe what requirements are tested, designed and validated.

4 THE TOP PRACTICES THOSE HAVE BEEN COMMENCED IN LEAN ARE:

- Designing at the very begin of a lean project to identify a feasible technical strategy for your solution there will be desire to do some preliminary, high-level architectural modeling.
- Continuous Documentation. Write deliverable documentation in small chunks of development during the lifecycle of lean-scrum development.
- Active Stakeholder Participation. Stakeholders and customers should provide information in a timely manner; decision should make on time, and in the development process throughout the use of systems be as actively involved.
- Iteration Modeling. At the start of each sprint there will do a chunk of modeling as part of iteration planning tasks.
- Document Late. As late as possible sprint should be write documentation, avoiding unreal ideas that are in errand of constant information likely to future adjust.
- Prioritized Requirements. Team implement requirements in desire priority order, as characterized by their stakeholders.
- Test-Driven Design (TDD). To accomplish a test write a single test case, either at the level of requirements or at design level.
- Traceability, to save the record of change in requirements.

5 DISCUSSION AND ANALYSIS

This section shows the results of the survey. Primary data for this study was collected through survey questionnaires from different software companies. The surveys were conducted to evaluate the proposed Model. The survey asked almost 13 questions to verify the usage lean principles and lean scrum models in these companies. The questionnaires were not open ended so that the resulted could be analyzed in better statistical form. The questioner based on the working of lean Scrum software development, if to be valid process for the development of software. The survey questions included the information of organization, practitioners, usage of lean principles and its benefits, improvements of using lean practices and principles and intentions its subsequent to reasons in adoption and the respondents who agreed to apply this hybrid model and who are not. Moreover it consisted on the information of the practitioners and their organization. The responses to surveys by the employee of chosen companies driving force towards the hybrid approach (lean, Scrum, agile) and advantages collected. The results are organized and compared with the results of previous methods on agile and lean with combined approach.

The total number of practitioners who were answering the survey questions was 30. The practitioners were working in different position in their organizations. The main organizational positions of the respondents were project managers,

developers and Software Engineer. Usage of lean scrum software development approach in the respondents was reported by 67% (respondents=20) organizational unit. The usage of lean software development methods would be help said by 67% because they agreed on about more than 50% benefits of lean. And left were not agreed or not sure and those were about 33%. Number of respondents who were accepted quality attributes about Lean Scrum Model shown in Table 1 where A is number of respondents who were accepted that quality attributes about lean. The graph also displayed which shows their level of acceptance.

Table 1: No of respondents who were accepted quality attributes about Lean Scrum Model

Lean benefits	Description	A
B1	Lean Principles in Scrum development reduce the developer overload	17
B2	Build quality into the code from the start take less time of software development	13
B3	Lean principles have acted as stabilizing mediator in scrum practice	14
B4	Decision will be made late lead to less taken of wrong decision	16
B5	Combining lean and agile with scrum model methodology improve scrum methodology	14

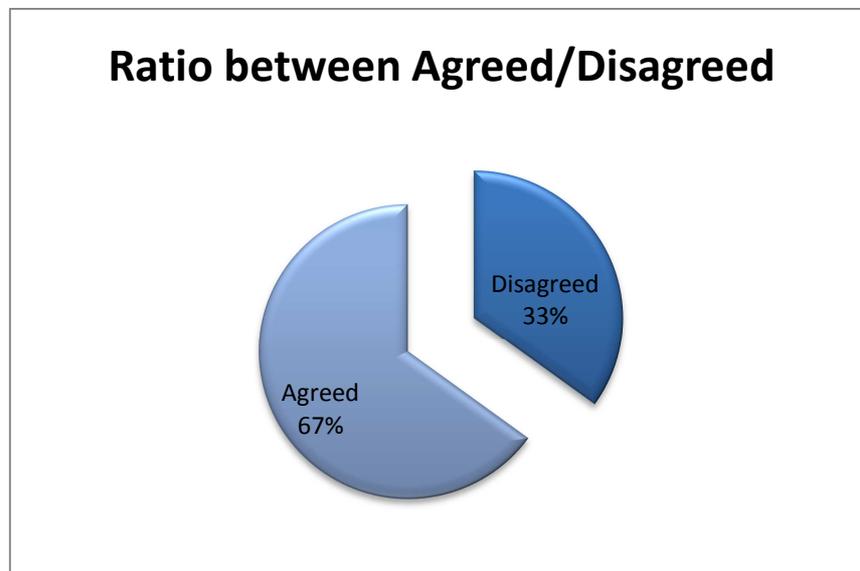


Figure 4: Ratio between Agreed/Disagreed

Fig.4 shows ratio between agree and disagree about Lean Scrum benefits.

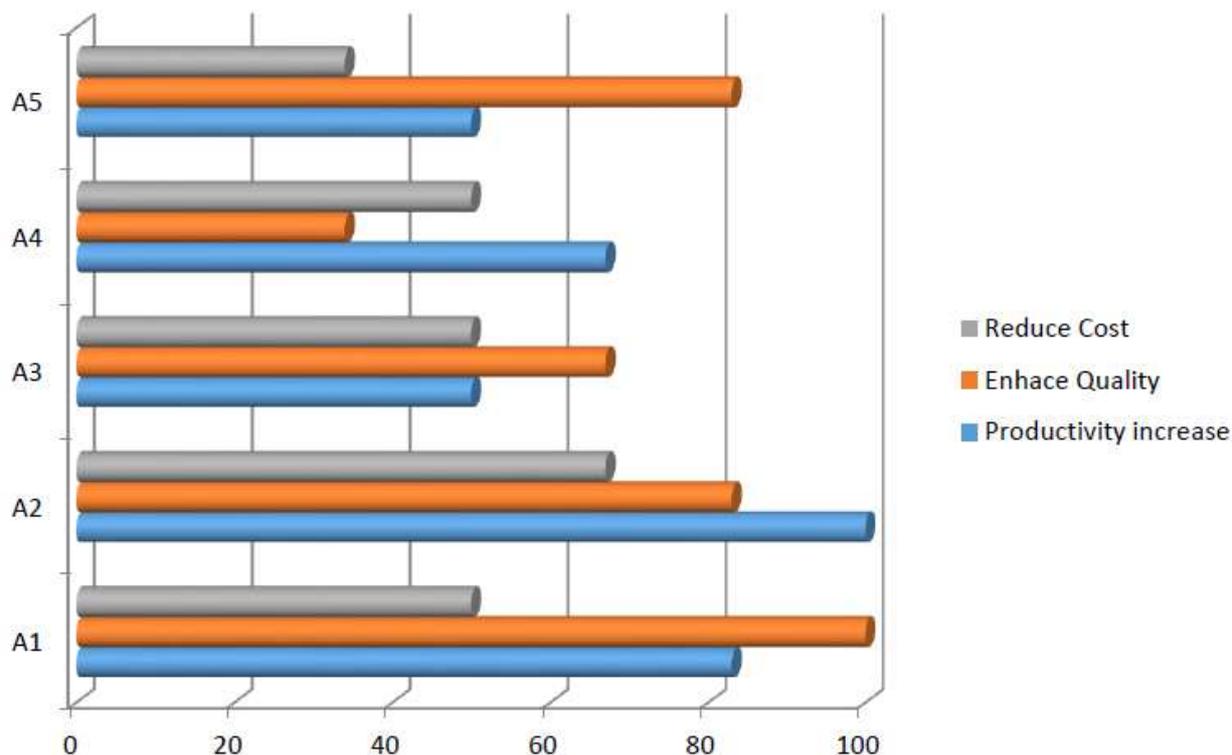


Figure 5: Benefits by using lean principles and agile Model with scrum practice

Fig.5 shows the benefits by using lean principles and Agile practice. Different organization shows productivity and quality enhanced with the proposed model.

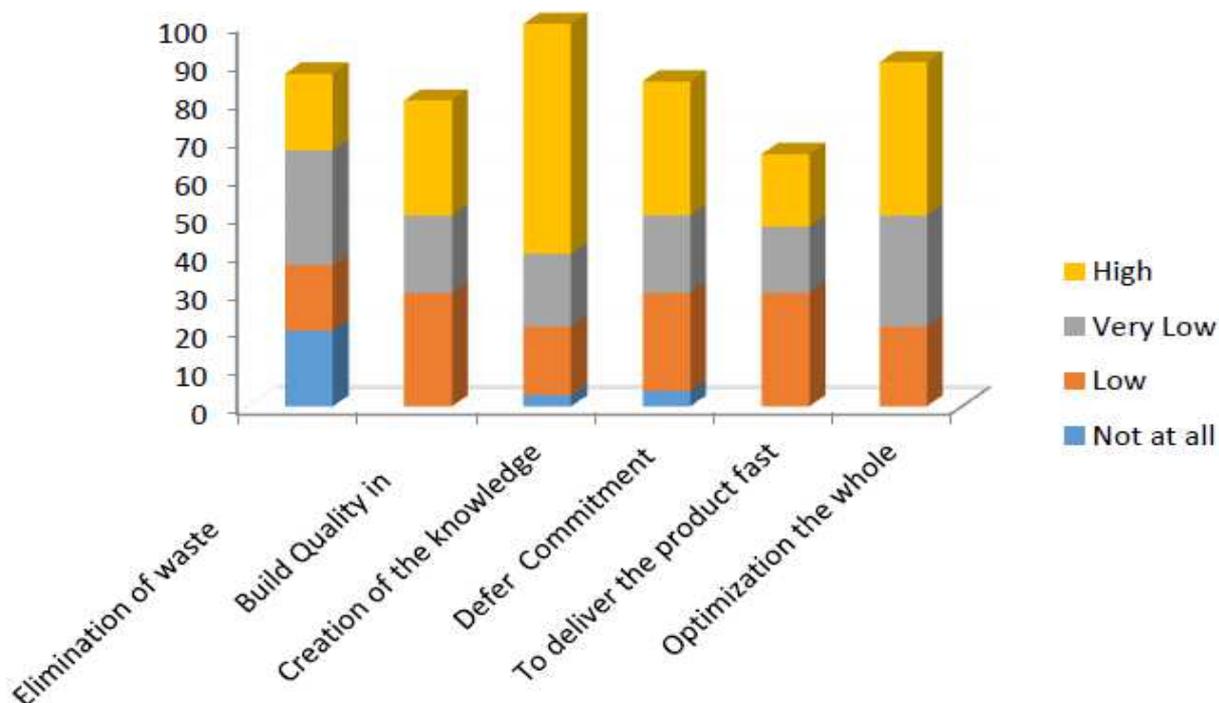


Figure 6: Lean principles value according to respondent

Fig.6 represents lean principles value according to respondents in software development process.

6 CONCLUSION AND FUTURE WORK

The current trend in software development points towards the combination of Lean thinking and ASD, after success in manufacturing process lean becomes most important part of software development. However, the body of knowledge on this topic is limited. This research formulated a process model by combining appropriate Lean principles with the agile process model to minimize the scrum issues and improved the scrum process certainty, productivity and stability. Lean thinking opens up a spacious variety of opportunities for advance research. The challenges recognized when moving towards a combination of Lean and ASD particularly advantage required further study.

REFERENCES

- [1] Pressman, *Software Engineering a Practioner's Approach*. New York, NY: McGraw-Hill education. pp. 38-94, 2012.
- [2] Middleton, P. and D. Joyce, Lean Software Management: BBC Worldwide Case Study. *IEEE Transactions on Engineering Management*, vol.59, no. 1, pp. 20-32. 2010.
- [3] Wang, X. Conboy, K. And O. Cawley, Leagile software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems Software* vol.85, no. 6, pp.1287–1299.2011.
- [4] Balachander, A and A. Harris, A. Lessons in lean [lean manufacturing]. *Manufacturing Engineer*.vol.83, no. 5, pp. 16-19. 2012.
- [5] Ali, A. M. Survey on the State of Agile Practices Implementation in Pakistan. *International Journal of Information and Communication Technology Research*, vol. 2, no. 5, pp. 459–464, 2012.
- [6] Ahmad, M., J. Markkula and M. Oivo. Kanban in software development: A systematic literature review. Proceedings of the 39th Euromicro Conference series on *Software Engineering and Advanced Applications (SEAA) Santander*, Spain, pp.9-16. 2013.
- [7] Jonsson, H., S. Larsson and S. Punnekkat, Synthesizing a Comprehensive Framework for Lean Software Development. *Software Engineering and Advanced Applications (SEAA)*. pp. 1-8.2013
- [8] Rodríguez P, Partanen J, Kuvaja P &Oivo M (2014) Combining lean thinking and agile methods for software development. A case study of Finnish provider of wireless embedded systems. Proceedings of the 47th Hawaii International Conference on Systems Sciences (HICSS 2014). In press.
- [9] Leffingwell, D. *Scaling software agility: Best practices for large enterprises*.1st Ed. 2007.
- [10] Agile Manifesto (2001) Beck K, Beedle M, van Bennekum A, Cockburn A, Cunningha, W.,Fowler M, Grenning J, Highsmith J, Hunt A, Jeffries R, Kern J, Marick B, Martin RC, Mellor S, Schwaber K, Sutherland J & Thomas D Manifesto for Agile Software Development, <http://www.agilemanifesto.org>.
- [11] Schwaber, K., and M. Beedle. *Agile software development with Scrum*, Upper Saddle River, NJ: Prentice Hall.pp.53-62. 2001.
- [12] Poppendieck, M. & Cusumano, M.A., *Lean Software Development: A Tutorial*. *IEEE Computer Society*. 2012.
- [13] Y. Matsui, "An empirical analysis of just-in-time production in Japanese manufacturing companies", *International Journal of Production economics*, vol. 108, no. 1-2, pp. 153–164, 2007.
- [14] Womack, J.P., Jones, D.T, *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*, Simon & Schuster, New York. 2003.