

# Applying Lean on Agile Scrum Development Methodology

SurendRaj Dharmapal, Dr. K. Thirunadana Sikamani

Department of Computer Science, St. Peter University  
St. Peter's College of Engineering & Technology, Avadi, Chennai 600 05, India

**Abstract:** This journal introduces the reader to Agile and Lean concepts and provides a basic level of understanding of each process. This journal will also provide a brief background about applying Lean concepts on each phase of agile scrum methodology and summarize their primary business advantages for delivering value to customer.

**Keywords:** Agile, Lean, Development Methodology, Project Management, Software Engineering, Eliminating Waste

## I. INTRODUCTION

Lean software development is a translation of lean manufacturing and lean IT principles and practices followed in manufacturing domain to the software development domain.

Now a day, Software Companies adopt both lean and agile software development methodologies with the intent to reduce development cost and project durations. What Agile and Lean have in common is their value-driven philosophy. Agile is about delivering value for the organization, based on a clear understanding of its values and objectives. The goal of Lean is to eliminate as much of the waste from the process and to balance and optimize the flow of work through the process for maximum efficiency

## II. BACKGROUND

Whether we want to admit it or not, projects fail and often even when completed, there is only a small portion of code which is utilized. The Standish Group reports that 45% of features and functions are never used, 16% are sometimes used and then 19% are rarely used. Thus why would you spend time on planning to develop features which in the end will not even be used? Lean and Agile addresses working on the highest priority items, assumes things will change, negotiates scope vs. having scope creep, and delivers working software at the end of each iteration. Considering risk is brought to light sooner, risk can be minimized. In the case of Lean and Agile, if a project is ended, there is business value which can be delivered, even if there is only 3 iterations worth of code vs. no code and no value. Thus there is early ROI, increased control and reduced risk, improved communication, and accelerated

time to market. Generally customer satisfaction is improved as a result and is better able to manage business through metrics.

## III. AGILE SCRUM DEVELOPMENT PHASES AND PROCESS.

Scrum is an agile process that allows us to focus on delivering customer request in the shortest time. The Agile Scrum is one of the processes widely followed across companies. In Agile Scrum, the business team prioritize the requirement needed in an Iterative model. Product progresses in a series of month-long "sprints" Requirements are captured as items in a list of "product backlog"

Scrum development methodology has the following three phase

### A. Planning Phase

Planning is the first phase in Agile process. This phase involves activities related to Requirement analysis & elicitation, preparation of stories for Product Backlog. Product Backlog will always evolve during enhancement or change of requirement.

#### *Steps involved in Planning Phase*

- Contract between the customer and vendor (if required)
- Stakeholder identification
- Kick off meeting with all stakeholders

- Story preparation for product backlog
- Each Story will be planned to deliver individual functionality that can be tested and delivered independently.
- Appropriate tools identification for development and process, hardware and software identification, technical feasibility, infrastructure, communication plan preparation, Resource identification, training plan preparation.
- Plan for each Iteration and Release
- Risk Analysis & Identification
- Appropriate metrics identification

- Eliminate waste
- Amplify learning
- Decide as late as possible
- Deliver as fast as possible
- Empower the team
- Build integrity
- See the whole

#### B. Development Phase

The key phase is the Development phase, also known as the Sprint. Sprint is an iterative cycle – that stops when requirements are completed. The scrum is Iterative and Incremental software development methodology. The Scrum will range from 1 to 6 weeks in duration. In each sprint, the product owner prioritizes the stories which are in backlog and bring them into Iteration.

#### A. Eliminate Waste

Agile Methodology highest priority is to satisfy the customer through early and continuous delivery of valuable software.

In order to be able to eliminate waste, one should be able to recognize it. If some activity could be bypassed or the result could be achieved without it, it is waste. Partially done coding eventually abandoned during the development process is waste. Extra processes and features not often used by customers are waste. Waiting for other activities, teams, processes is waste. A defect is also a waste. We do skip the traditional formal documentation such as requirement and detailed design in Agile methodology. The customer is involved throughout the software development and gets what is needed at the early stage by providing constant feedback to the developers. Eliminating Waste reduces cost and time..

#### Steps involved in Development Phase

- Daily Stand-up meeting
- Planning Prioritized Stories for an Iteration and setting up acceptance criteria
- Coding and Unit Testing
- Reviews
- Implementation in appropriate environment
- System Testing and User Acceptance Testing
- Demonstration of Software developed
- Retrospective meeting

#### B. Amplify Learning

Learning is a continuous process in Software Development. Most agile methods accumulate learning and knowledge by prototyping screens to solicit feedback from end-users. The accumulation of defects should be prevented by running tests as soon as the code is written. Defect analysis and learning should be discussed in retrospective meetings. Instead of adding more documentation or detailed planning, different ideas could be tried by writing code and building usable product. It's also important to have a body of reusable standards and guidelines that people can easily modify to meet specific project needs. Identifying the areas that need the most improvement (knowledge-wise) and taking steps to improve is the key to Amplify learning.

#### C. Closure Phase

Last phase of this process is called Closure. Closure occurs when all requirements are met. This is the last phase before releasing the product with all documentation. Final tests are done at this point and at the end - the release itself is final

- Integration/ Release Testing
- Defect Fixes if any
- User Acceptance Testing
- Implementation Plan Preparation
- Implementation
- Post implementation support

#### C. Decide Late

Every decision has a cost. It may not be immediate and material, but the decision has a cost associated with it. Sometimes, deferring decisions helps to fully prepare for the problem that needs to be managed. Many big projects developed in traditional waterfall model failed or increased cost because most of the functional and technical design assumptions made at the beginning of a project changed at the time of completion. Customers do change requirements or features when they see the developed product during UAT Phase. Since Agile methods rely solely on short iterations and constant code re-factoring, adapting to these design changes is possible. In Complex projects delaying certain crucial decisions until customers have realized their

#### IV. LEAN BACKGROUND

Lean development can be summarized by seven principles, very close in concept to lean manufacturing principles:

needs better is one of the feasible approach in Agile Methodology.

**D. Deliver Fast**

Delivering a product sooner will definitely adds value to the customer. History has proved that an incomplete product with acceptable or known defects is better than nothing. In Agile, the sooner the product is delivered, the sooner you get feedback to incorporate into the next iteration. The shorter the iterations, the better the learning and communication within the team.

**E. Empower the team**

The lean approach favours the aphorism "find good people and let them do their own job," encouraging progress, catching errors, and removing impediments, but not micro-managing. Given the environment and support they need, and trust them to get the job done. The best architectures, requirements, and designs emerge from self-organizing teams.

The development team have the exclusive right to estimate the level of effort required to deliver a particular feature in the product backlog. The business cannot force a change to an estimate nor question the estimate.

**F. Build Integrity**

Model-based solutions are better aligned to the business need because they are focused on the business and not the technology. Models are used to establish a shared understanding of what the solution is expected to do. UI models are used to give the end-user a sense of the user experience very early in the life cycle. Requirements are expressed in terms of model elements, so they are captured using a common "language" and can be tested early and often to ensure the solution will work as needed (as opposed to "as designed"). The architectural quality of software design is ensured by using design and code automation. Blueprints codify design guidance, standards, patterns, and best practices and are used to compile models into high quality, consistent code. The blueprints can be modified to incorporate new learning's or non-functional requirements and then re-applied to propagate across the code base. This ensures that the solutions remain flexible, maintainable, and extensible.

**G. See the whole**

Lean thinking has to be understood well by all members of a project, before implementing in a concrete, real-life situation. "Think big, act small, fail fast; learn rapidly" – these slogans summarize the importance of understanding the field and the suitability of implementing lean principles along the whole software development process.

**V. CHALLENGES AND LEAN SOLUTIONS**

Let us discuss some common challenges in Agile Scrum methodology and overcoming those with lean solutions

Below are some challenges in Agile Planning Phase

**A. Story Backlog**

Agile Scrum Challenge	Lean Solution
Unwanted Stories written by user in story backlog	Identify and remove unwanted stories at early stage – Eliminate waste
There may be missing stories in backlog since the user does not have clear visibility of requirement at early stage.	Decide the story late immediately getting complete visibility of requirement
Independent module could be developed but Integration of stories missing in backlog are difficult to implement.	See the whole before beginning of project so that the integration of stories when developed does not fail.
Excessive Stories in backlog resulting in overlapping of stories or huge amount of time spent on creating stories	Decide the story late when required.

**B. Tools**

Agile Scrum Challenge	Lean Solution
No common tool for the development	The common tool helps in reducing effort in manual process.
The tools not adequate to capture all required information's	The tools used should be flexible to capture all assumptions, discussion threads, conclusions, effort, status, acceptance criteria, defect log and more. This eliminates the effort in tracking manually.

**C. Hardware and Software**

Agile Scrum Challenge	Lean Solution
Hardware and Software not available on time	Eliminate the waste of waiting time and impediment in development by planning necessary

	hardware and software required for the development and make it available on time.
--	---

*D. Access to Resource*

Agile Scrum Challenge	Lean Solution
Stake holders not having proper permissions to tools or required resources based on their respective role.	With proper permissions to the required resources based on respective role helps to deliver the project as fast as possible.

*E. Infrastructure*

Agile Scrum Challenge	Lean Solution
Inadequate infrastructure for the development such as communication facility etc	Effort will go waste without having proper infrastructure for the development. The team has to ensure that it has all required infrastructure

*F. Product Owner*

Agile Scrum Challenge	Lean Solution
Product Owner having inadequate knowledge of requirement or end product or goal	Defer the development until the requirement is clear and unambiguous. See the whole picture before start of development.
Involvement of product owner is inadequate	Active involvement of product owner throughout the development phase is needed to make the product right and eliminate any rework effort.

*G. Resources*

Agile Scrum Challenge	Lean Solution
unavailability of skilled resource	Empower the team by having right skilled resource. Emphasize the concept of "Right Resource Right Work"

*H. Daily Standup*

Agile Scrum Challenge	Lean Solution
Long meetings	Lean emphasizes on having short meeting to the point of eliminating waste
Team member participation	Only needed members can participate in the daily stand-up.

*I. Stories*

Agile Scrum Challenge	Lean Solution
Incorrect prioritization of stories	Correct prioritization helps to eliminate the waste of time developing stories that are not in priority.
Improper estimate	The team should be empowered to provide the better estimate for the work.
Inadequate impact analysis	The team should see the big picture of the product and the proper impact due to changes or enhancement to be determined at early stage itself.
No acceptance criteria	The acceptance criteria will help the developers to go in right track and to deliver right product to the customers and eliminate the waste of developing unwanted features or functionalities
No time set for reviews	Reviews are the most important quality process that should be followed during the development phase. Empower the team to set time for reviews to reduce any defects at later stage. Built the integrity with the team to ensure reviews are done at right time.
No demo to customer	Demo to customer helps to build the integrity and trust.
Assumptions and discussion points on stories not updated in a common tool	The process of updating the assumptions and discussion points on common tool helps the learning to develop the product correct.
No scope freezing	Scope freezing is essential to deliver the product faster
Flexibility in deadline date	The deadline date again helps to deliver the product

	faster.
Integration stories not available	Separate stories needed to be built for integration of all stories developed.
Defect analysis and learning's in retrospective meeting	Lean emphasize on amplify learning's. Therefore this defect analysis and learning's in retrospective meeting helps deliver more quality product.

• CONCLUSIONS

The Agile can be considered a methodology that has its roots in Lean manufacturing. Lean encourages continuous improvement; Agile utilizes Iteration Planning sessions and retrospectives, thus encouraging constant inspecting and adapting. Lean looks to minimize risk; Agile brings risk to light faster through iterative development and prioritization. Lean looks to minimize inventory and storage, while Agile minimizes work in process and thus avoids surprises which often come at the end of a project. Lean-Agile is a combination of Lean Thinking and Agile disciplines.

Lean Agile helps to align business needs to the project as well as aligning to the vision and mission of your organization, thus there is an improvement to the service provided to your customers.

REFERENCES

[1] Practices For Scaling Lean And Agile - by Craig Larman

[2] Lean-Agile Software Development: Achieving Enterprise Agility - Alan Shalloway (Author), Guy Beaver (Author), James R. Trott (Author)

[3] The Software Project Manager's Bridge to Agility - Michele Sliger (Author), Stacia Broderick (Author)

[4] Lean Software Development: An Agile Toolkit Mary Poppendieck (Author), Tom Poppendieck (Author)

[5] Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum - Craig Larman (Author), Bas Vodde (Author)

[6] Mary Poppendieck and Tom Poppendieck. Lean software development: an agile toolkit. Addison-Wesley, Boston, 2003.

[7] James M Morgan and Jeffrey K. Liker. The Toyota product development system: integrating people, process, and technology. Productivity Press, New York, 2006.

[8] Mary Poppendieck and Tom Poppendieck. Leading lean software development: results are not the point. Addison-Wesley, Upper Saddle River, NJ, 2010. Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[9] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

[10] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. .