

# A Review on Lean Thinking for Software Process Improvement

Dr. Poornima Nataraja<sup>1</sup>, Bharathi Ramesh<sup>2</sup>

Professor and Head, Dept. of MCA (VTU), Dayananda Sagar College for Engineering, Bangalore<sup>1</sup>

Assistant Professor, Dept. of MCA, Surana College, Bangalore<sup>2</sup>

**Abstract:** Software engineering is rapidly evolving field with new methodologies, technologies, design tools, and paradigms emerging every year. Software industry is a relatively young business, actively searching for innovative ways of developing products. The latest trends towards a more customer-centric, responsive, iterative software development are bringing up new paradigms, frameworks in the field of software development processes. Lean software development is a translation of Lean manufacturing and Lean IT principles and practices to the software development domain which introduces a Continuous Improvement Model that provides a stable development model for organizations. Lean has become a standard for efficiency in production systems. This paper mainly focused on lean, principles and practices of Lean with Agile, importance of lean in reducing wastes during software development and maintenance.

**Keywords:** Lean, Lean Thinking, Software Development life cycle, Lean Software Development[LSD], Process Improvement, defects, Software wastes, Software maintenance.

## I. INTRODUCTION

Increasing role of technology has a major impact on our day-to-day life. The software industry is afflicted with problems, such as high cost, low quality, unexpected project failures, and missed deadlines. To overcome these limitations, Lean is one of the ways to improve software development and continuous process improvement.

In the late 1940s, as a small company, Toyota changed the way of producing cars and the way managers believed production should work. Since then, many companies have changed their production management by the Lean Thinking. One of the domains affected by the Lean Thinking was the software development, which generated the term Lean Software Development, according to the works of Mary and Tom Poppendieck [1].

### A. LEAN SOFTWARE DEVELOPMENT

Lean software development is a product development paradigm with an end-to-end focus on creating value for the project owner and for the end user, eliminating waste, optimizing value streams, empowering people and continuously improving [2]. The Lean software development methodology does not impose a certain process in terms of conducting the project. The project owner's feedback is central to the Lean software development methodology [3].

Lean Software Development does not prescribe particular practice or model. It is more important to demonstrate that actual development process definitions are aligned with the lean principles and values.

### B. APPLICATIONS OF LEAN DEVELOPMENT

- Standardized processes to ensure everyone uses best practices

- More successful projects (satisfied customers and project team, full-scope, on-time, under-budget).
- Faster project completion
- Reduced pressure on project team members.
- Reduction of the waste that causes project delays: Multi-tasking, queuing,

### C. LEAN PRINCIPLES

Womack and Jones [4] bring forth five basic principles

1. Specify value from the standpoint of the end customer by product family.
2. Identify all the steps in the value stream, eliminating whenever possible those steps that do not create value.
3. Make the value-creating steps occur in tight sequence so the product will flow smoothly toward the customer.

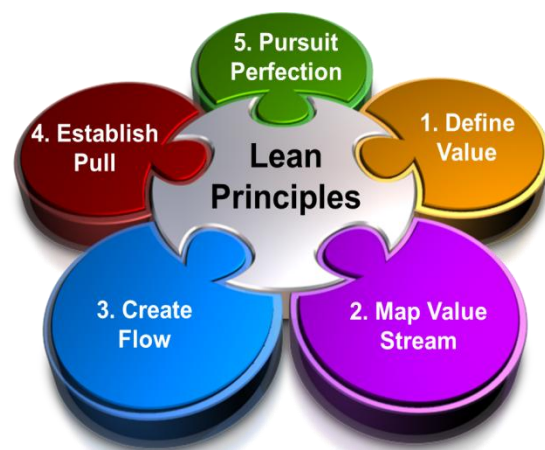


Fig 1: Lean Principles

4. As flow is introduced, let customers pull value from the next upstream activity.
5. As value is specified, value streams are identified, wasted steps are removed, and flow and pull are introduced, begin the process again and continue it until a state of perfection is reached in which perfect value is created with no waste.

#### D. LEAN SOFTWARE DEVELOPMENT PRACTICES

Lean practices, or what the Poppendiecks call "tools" are expressed slightly different from their equivalents in Agile software development, but there are parallels. Examples of such practices include [5]:

- Seeing waste
- Value stream mapping
- Set-based development
- Pull systems
- Queuing theory
- Motivation
- Measurements

Lean thinking represents a culture where all employees continuously look for ways to improve the process with the philosophy of eliminating all non-value added activities, encompassing wasted time, activities, inventory, and space – and create processes that flow and are initiated by customer demand.

#### II. LEAN AND AGILE

Agile and Lean are becoming widely accepted approaches in industrial, civil and software engineering. Their association is considered as a new competitive strategy and is claimed to be “the next wave of life-cycle process”.

Lean and Agile methods have been proposed to facilitate the effective project deployment and organization for minimizing resources, costs, durations and risks. Agile shares the same principles with Lean. Basically, agile approaches are instances of Lean thinking. Also it is practically proved that success of Agile Software Development practices can be explained by understanding the principles of Lean Software development [6].

Agile practices enables companies to respond quickly as needed when market conditions change, new technologies arise, or new ideas are developed.

A retrospective studies show that a part of Agile community has started to look toward Lean approaches capability, in addition to Agile methods such as XP and Scrum. Others even consider Lean as just another Agile method.

Some common principles shared between Lean and Agile software development.

- People centric – First priority is people. Processes, tools, technologies are here to support them.
- Increased value and reduced waste – providing what people really needs (value) and are willing to pay for.

- Seamless flow – process roles are just the decomposition of the process activities.
- Faster lead time – ability to deliver a product or service faster is quite important. The same applies also to responding to change and learning
- Built-in quality – quality is not just polishing at the end of the process, it is the aspect of every step in the value chain.
- Learning – continuous reflection and step-by-step improvement is necessary towards learning organization.

#### III. REVIEW OF LITERATURE

According to several studies from Standish Group, Gartner group, Cutter consortium, Centre of project management and others, on average:

- Approximately 23% of all software projects were successful
- Approximately 53% were completed but exceeded time and/or budget
- Approximately 24% were aborted

Also there seems to be a direct correlation between team size and project duration and high failure rate [4].

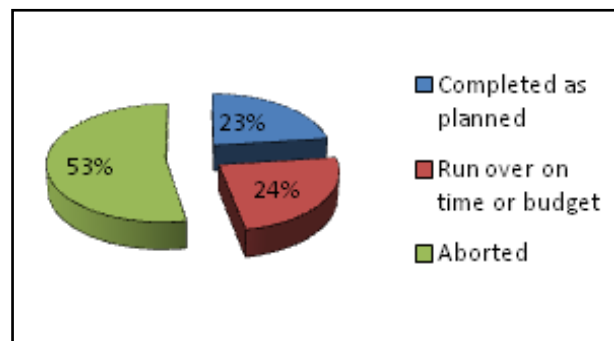


Fig 2: Project Success Rates

The general root causes for project failure are:

- Lack of Project management culture and maturity
- Project management regarded as heavy overhead bringing no benefits, therefore not implemented
- Organization not adapted to the project
- Missing method and process
- Unclear roles and responsibilities
- Information does not flow properly
- Proper tools not adapted
- Unclear scope and requirements
- Missing prioritization of requirements by stakeholders.

So there is always a room for improvement, especially in a fast changing world that needs to adjust quickly to ever-changing environments. Software companies could collect the project management’s best practices in the company from the domain experts and combine those in a single, documented process that everybody could use when managing the project [7].

In recent years, Lean and Agile Software Development practices play a very important role with their principles and practices. 70% of all traditional development methodologies are replaced by these two approaches [8]. Lean Management is a system of organization based on continuous improvement with a strong participation of all personnel involved in the process.

A summary of the lean thinking principles discussed in relation to software development processes and the potential benefits that can be derived from these processes is presented in Table 1.

Table 1: Lean Thinking Principles, Software Development Processes and Potential Benefits.

Lean Thinking	Principle Software Development Processes	Potential Benefits
<b>Eliminate Waste</b>	Identify Waste	non-value adding steps
	Value Stream Mapping	Eliminates non-value adding steps
<b>Amplify Learning</b>	Software Development Feedback Loops	Allows feedback to developers early in the development cycle
	Set-Based Development	Allows for the development of multiple options in parallel
<b>Decide As Late As Possible</b>	Options Thinking Enables set-based development	Enables set-based development by examining all options
	The Last Responsible Moment	Enables the decision maker to have as much information as possible when making decisions, resulting in better decisions
<b>Build Integrity In</b>	Perceived Integrity	Better customer satisfaction and a product that meets the customer need (more sales).
	Conceptual Integrity	Better product that requires less and easier maintenance.

**A. SEVEN SOFTWARE WASTES**

As in manufacturing, lean thinking in software begins with the ability to identify these wastes. The technique for accomplishing this in software development is the same as in manufacturing: create a value chain and then look for steps in the value chain that add no customer perceived value and eliminate them [9].

The iterative process of learning and continual improvement is an important part of identifying waste and eliminating it.

The best approach to eliminate waste is to follow some basic principles.

- As soon as waste is detected, the priority should be to eliminate the waste and its cause
- The search for waste should be continuous; it should be performed by all the team members, all the time.
- Prioritize the waste to eliminate.

Seven important wastes identified in software development life cycle are shown in fig 2.

As waste elimination is one of the Lean principles and one of the most effective ways to increase quality and reduce cost. Lean Management involves all stakeholders to eliminate waste which reduce the efficiency and performance of a project.



Fig 3: Seven wastes in software development

**B. LEAN AND AGILE SUPPORT FOR SOFTWARE MAINTENANCE**

Several researchers stated that it is appropriate to adopt principles and practices of lean and agile for major disciplines involved in the software life-cycle in order to increase the success rates of projects.

- Requirement engineering
- Architecture

- iii) Programming/Development
- iv) Testing and verification
- v) Integration
- vi) Project Management/Maintenance

Middleton et al [10] studied an organization that had several problems with project resource allocation and tracking, with too high costs for development and the role of lean and agile principles in improving project management.

Software maintenance/project management is one of the major concerns of software development. Good maintenance process is very essential to maintain the quality of software [11].

This stage is very important as it:

- ✓ Ensures that the system remains running at Peak performance levels.
- ✓ Solves any software bugs/problems that arise.
- ✓ Customize the software to users' needs as user's needs may change with time.
- ✓ Adds increased functionality to the system.

### C. ISSUES AND CHALLENGES

The maintenance phase of the SDLC lasts the longest of all the phases. Various research studies proposed that software maintenance consumes 60% to 80% of cost in whole development life cycle; these surveys also report that maintenance costs are mainly due to enhancements, rather than corrections [12].

There are several technical and managerial problems encountered while maintaining software [13].

- a) Cost
- b) Impact Analysis
- c) Corrective Changes:
- d) Adaptive Changes
- e) Program Comprehension

IT organizations can improve their quality and productivity during application development is by eliminating its largest sources of waste – defects and the rework they cause. In many organizations, 30% to 50% of development effort is devoted to rework [14] [15]. That means defects become 10 times more expensive to fix for each major phase of the software life cycle. Under these circumstances, productivity is largely determined by quality.

### D. LEAN PRINCIPLES IN MAINTENANCE/PROJECT MANAGEMENT

Lean Thinking is the manufacturing concept borrowed into software development and management of IT products and services. There are few practices used in lean mainly to reduce the issues in post development phase, mainly focused on defects identification which reduces complexities in maintenance phase. There are many existing tools in which two of them described below:

Kanban is a lean and Just-in-Time process for regulating the flow of software development based on demand. It is a

framework for optimizing workflow that maximizes efficiency, product quality and customer satisfaction.

The entire process of software development can be considered as a pipeline in which requested software features and development issues go through predefined steps to produce the deliverable and deployable software.

Kanban is defined to be executed in value stream with focus on delivery of value. Kanban in software development can be visualized as the features flowing across the value stream.

Six general practices of Kanban are:

1. Visualize
2. Limit work in progress
3. Manage flow
4. Make policies explicit
5. Implement feedback loops
6. Improve collaboratively, evolve experimentally.

### Disadvantages of Kanban

- ❖ David Anderson said “It is actually not possible to develop with only Kanban. The Kanban Method by itself does not contain practices sufficient to do product development”.
- ❖ Keeping the Kanbans resized as the demand frequently changes can be slow and difficult to manage.

Just visualizing the workflow and the other Kanban principles is not enough for software development. Software development has things like business value, technical complexity, and user experience/acceptance/adoption — all of which are not addressed directly by Kanban.

Several existing tools and practices in lean and agile such as Scrum, XP, Kanban, CMMI and etc. which are used for efficient project management process, with many pros and cons.

Present Lean Software development [LSD] mainly focused on three areas as the primary way to increase efficiency, assure quality, and improve responsiveness in software-intensive systems [16]:

- Understanding the customer journey,
- Architecture and automation to expose and reduce dependencies
- Team structures and responsibilities.

But in Lean the workability of the team decides success of development process. Also unsuitable business analyst or team members, excessive flexibility leads to more problematic situation. Projects and programs are basically flexible in nature that creates change. To successfully implement organizational strategy, companies need project and program managers with the appropriate skills to drive and navigate change.

Organizations that manage change effectively will stay ahead of the competition. The reality is only 18 percent of companies are effective change enablers [17].

Change enablers incorporate these practices:

- Having well-defined milestones and metrics

- Having senior management committed to change
- Establishing and communicating concrete ownership and accountability
- Using standardized project management practices
- Having engaged executive sponsors

#### IV. OBSERVATIONS

Lean software Development practices are a good way to pinpoint bottlenecks in the existing processes. Since the lean principles focus on eliminating wasteful (non-value added) activities and optimizing the whole process using existing resources and functions.

The main aim of this paper is to summarise the concepts of lean applied for different SDLC activities. Literature review mainly focused on the seven software wastes in process activities and also issues and challenges in project management/maintenance phase.

Many IT organizations are currently implementing or leaning towards the modern development methodologies. Exponential growth is essential to the industry, and IT companies should know their way around processes that enable and support it.

The following are few observations from the literature survey:

- One of the key challenges in lean is identification of value generation stream and eliminating the non-value adding activities termed as ‘Waste’, in order to improve quality, speed, efficiency and productivity. Basic problem in this activity is identification of valid waste. For Example: the work done by an agile team towards a deadline, but when it is completed it ends up standing in a queue and from lean point of view this is waste. This kind of situations occurs in all the SDLC activities. Setting the clear standards, practices and principles of lean to identify and reduce these risks is the primary issue.
- As mentioned earlier more than 60% of development cost is consumed by maintenance phase. The two primary sources of waste in maintenance are rework due to defects or badly constructed software. By identifying structural defects that have a high probability of causing operational problems and result in maintenance rework, software analysis and measurement provides strong support for Software Maintenance.
- There may be so many barriers in all the organizations for implementing, aligning the lean principles according to the requirements and also achieving successful lean projects. It is proved that Lean Software Development is much more than just methods or tools. Even though there are many existing tools in agile and lean all are based on the underlying philosophical thinking and principles. Aligning these principles for the company’s competitive situations also a major challenge.

In addition here are few downsides in the existing practices and methodologies:

- ❖ Rigidity in process design frameworks and development
- ❖ Process selection and integration are cumbersome
- ❖ Requires specialized skills, training and competency
- ❖ Requires organization cultural change
- ❖ Problems in architecture alignment and
- ❖ Continuous measure and improve the processes and its standards.

A few key areas/main activities is proposed in fig 4 which requires specific work standards and alignment of lean principles:

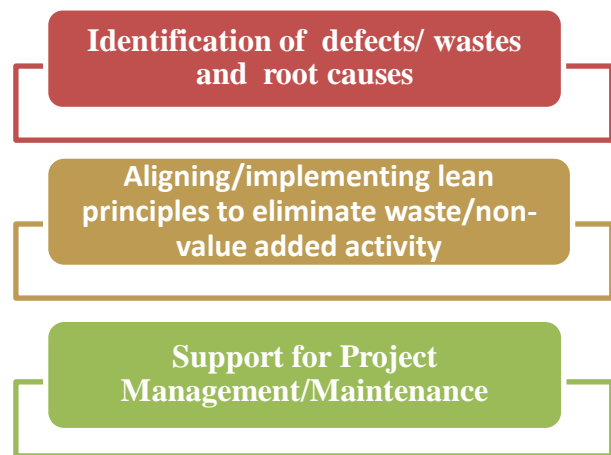


Fig 4: Key areas for lean implementation

Lean however does come with its own set of challenges and complications. Companies that find it difficult to imbibe and transform with Lean because of the following reasons:

1. The main problem of all Lean and Agile projects is their high dependence on the qualification and discipline of team members.
  2. Lean isn’t just a set of tools or a change program that can run its course over a short period; it takes considerable time for organizations to change and evolve
  3. Lean is a change in thinking which doesn’t apply only to specific departments or functions, but even the upper levels of management and the organization as a whole.
- By focusing on the fundamentals of developing a project management culture, cultivating talent, and defining processes either using lean or agile, organizations will capture the value of project management and gain competitive advantage.

#### V. CONCLUSION

Increasing demand to find suitable projects management approaches for maintenance and support of applications has become a major task nowadays. Software development

using lean management system is changing the way that software is developed. With shrinking project outcomes, shifting global trends and an uncertain economy, organizations must shift their thinking and embrace project management as a strategic competency that can reduce risk, cut costs and improve success rates. In this regard lean and agile are aiming towards achieving business goals and satisfying customers with a competitive product of the best quality.

[18] Wang, Xiaofeng, Kieran Conboy, and Oisín Cawley. 2012. "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems & Software* 85, no. 6: 1287-1299.

### ACKNOWLEDGMENT

One of the authors **Ms. Bharathi Ramesh** acknowledges Surana College PG Centre, Bangalore, Karnataka for providing the facilities for carrying out the research work.

### REFERENCES

- [1] S. M. Mary Poppendieck; Tom Poppendieck (2003). *Lean Software Development: An Agile Toolkit*. Addison-Wesley Professional. pp. 13–15. ISBN 97 8-0-321-15078-3.
- [2] C. Ebert, P. Abrahamsson, N. Oza, "Lean Software Development", *IEEE Software*, vol. 29, no. 5, pg. 22-25, 2012.
- [3] Mihai Liviu Despa "Comparative study on software development methodologies" *Database Systems Journal* vol. V, no. 3/2014
- [4] Womack, James P., Daniel T. Jones and Daniel Roos, *The Machine That Changed the World: The Story of Lean Production*, 2007 updated edition, Free Press, 2007
- [5] Mohamed-Larbi Rebaiaia, Darli Rodrigues Vieira "Integrating PMBOK Standards, Lean and Agile Methods in Project Management Activities" in *International Journal of Computer Applications* (0975 – 8887) Volume 88 – No.4, February 2014
- [6] M. Shabeena Begam, R. Swamynathan, J. Sekkizhar "Current Trends on Lean Management – A review" *International Journal of Lean Thinking / Volume 4, Issue 2* (December 2013)
- [7] Piyush Kumar Pareek, Dr. A. N. Nandakumar "Agile and Lean Usage in Small and medium Software Firms" *International Journal of Advanced Research in Computer Science and Software Engineering* 4(12), December - 2014, pp. 813-818
- [8] Lv Yan-mei, Wang Gefang, Wang Haitai, Wang Jialing, Wang Weihue "Design and development of the software system for equipment FMEA based on .NET .. *Mechatronics and Automation (ICMA)*, 2012 International Conference on Digital Object Identifier: 10.1109/ICMA.2012.6285745, Publication Year: 2012, Page(s): 2531 – 2535, IEEE CONFERENCE PUBLICATIONS.
- [9] Reinertsen, Donald G., *The Principles of Product Development Flow: Second Generation Lean Product Development*, Celeritas Publishing, 2009
- [10] Middleton, P., A. Flaxel, and A. Cookson. 2005. *Lean software management case study: Timberline Inc.*
- [11] Software Maintenance: Challenges and Issues Aakriti Gupta St. Xavier's College, JAIPUR - 302001, INDIA Shreta Sharma St. Xavier's College, JAIPUR - 302001, INDIA shretasharma@hotmail.com. Aakriti Gupta et al. / *International Journal of Computer Science Engineering (IJCSE)*.
- [12] Abran A., Moore J.W. *Guide to the software body of knowledge (SWEBOK)*. Ironman version. IEEE Computer Society Press: Los Alamitos CA, pp. 6-1-6-15.
- [13] Alkhatib, G. The maintenance problem of application software: an empirical analysis. *Journal of Software Maintenance: Research and practice* 1,2 s.pp.83-104.
- [14] R. Dion (1993). Process improvement and the corporate balance sheet. *IEEE Software*, 10 (4), 28-35.
- [15] B. Boehm (1987). Increasing software productivity. *IEEE Computer*, 20 (9), 43-57
- [16] Poppendieck, Mary and Tom. "Lean Mindset", February 2015
- [17] "Enabling Organizational Change Through Strategic Initiatives" by Part of Pulse of the Profession®, *The High Cost of Low Performance* 2014 series