Journal of Advanced Research in Quality Control and Management
Volume 5, Issue 2 - 2020, Pg. No. 15-20
Peer Reviewed Journal

Article

Sustainable Software Development using Lean Methodologies

Sonu Gupta

Assistant Professor, Thakur Institute of Management Studies, Career Development and Research, Mumbai, Maharastra, India.

I N F O

E-mail Id:
sonu.gupta@thakureducation.org

How to cite this article:
Gupta S. Sustainable Software Development using Lean Methodologies. *J Adv Res Qual Control Mgmt* 2020; 5(2): 15-20.

Date of Submission: 2020-12-29
Date of Acceptance: 2021-03-03

A B S T R A C T

The Industrial Revolution in the world of manufacturing started during the early 1800s and took a different course in the early 1900s when Henry Ford built the first mass-produced automobile. Henry's Ford principles of "scientific" management and assembly-line production that dominated the automobile industry for much of the 20th century are still prevalent in manufacturing business. The way auto factories-built cars on assembly lines, tested them at the end and fixed what didn't work, software programmers spend months working on software code, test it at the end and then rework it again, sometimes even after the product has been released. In the 1950s Toyota introduced lean manufacturing into its factories which led to increased and earlier testing, improved efficiency and lowered costs.

This paper discusses applicability of lean principles in software development. Lean IT is not only a set of tools and practices, it is a cultural transformation that educates everyone in the organization to think about the role of quality information in the creation and delivery of value to the customer.

Keywords: Lean Methodology, Value Added Activity, Value Stream Analysis, (TPS) Toyota Production System, (ADM) Application Development Methodology, 5S, Software Waste

Introduction

Quality information and effective information systems are essential for the success of not only IT, but all business organizations. Hence the effect of unstable and inflexible IT systems, failed projects, misalignment of IT activities with business strategy become all the more dire and unsustainable.[1] Moreover, with inherent volatility in the business environment, probability of changes in business requirements also increases. A study by Barry Boehm and Victor Basili found that the cost of correcting a software defect was approximately $139 during the software requirements phase, $455 in the design phase, $977 during the coding phase, $7, 136 during testing and $14,102 in the maintenance phase after the software has been released.[2]

The lean manufacturing and Toyota Production System (TPS) techniques can be successfully applied to the software development process to help developers and their organizations find and fix software defects earlier and continuously during the software development cycle and reduce the overall cost of the project.

Lean programming explores advantages of using smaller modules of flexible programming code to build software for customers based on emerging needs. Windholtz, owner of Cincinnati's ObjectWind Software, said this approach reduces errors and gets products in customers' hands much faster, its business implications are much wider than just for those in the software business.[3] Software development organizations face the daunting task of creating high

quality, reliable, maintainable software while still delivering their systems on time and within budget. Software tools incorporating lean manufacturing help to achieve these seemingly opposing objectives. In 2007, Accenture CTO Don Rippert described "Industrialized Software Development" as one of eight major trends that Accenture has identified as likely to have major impact on IT over the next five years.[4]

David Norton, a research director at Gartner, confirms that adopting lean can require quite a radical organisational change. "I think that within about five years, most CIOs will be aware of lean and will deploy the principles in some fashion. But it is an area that is ripe for consultancy as it is not just a matter of downloading a specification and implementing it. Organisations need to understand the principles and interpret and apply them to their own processes, which means that it is up to them to do most of the hard work," he says.[5]

The aim of this publication is to do a systematic review of the existing literature on lean principles and analyze its applicability in software development.

There are obvious similarities in manufacturing industry processes and Application Development Methodology (ADM) of IT systems. (Table 1).

Table 1.Similarities Between Lean Manufacturing and Effective Product Development

Lean Manufacturing	Lean ADM
Frequent Setup changes	Frequent Software releases
Short manufacturing time	Short development time
Reduced WIP (Work in Progress) between manufacturing steps	Reduced information inventory between development steps
Frequent transfer of small batches of parts between manufacturing steps	Frequent transfer of preliminary information between development steps
Reduced Inventory	Reduced development time
Adaptability to changes in product mix	Adaptability to changes in product design

Literature Review

Although lean principles were originally developed for manufacturing industry, they are increasingly and successfully being applied to service businesses. The best example of is delivering packages overnight by Federal Express, started in 1971. The concept of shipping products the same day they were ordered was a novel concept when LL Bean upgraded its distribution system in the late 1980's. Dell maintains profitability in a cutthroat market by manufacturing to order in less than a week.[6] eBay eliminated all the unnecessary steps in the trading value chain by adopting lean thinking.[7] In each of these examples, the objective of software development was in-time response to any emergent need. The basis behind these and many other industry-rattling success stories is lean thinking. Lean thinking looks at the value chain and focuses on. How can things be structured so that the enterprise adds value, and does that as rapidly as possible? All the intermediate and unnecessary steps, time and people are eliminated.

There is definitely scope of applying lean methods to application development and maintenance not only because it involves a great many processes with the potential to be optimized but also because large differences in productivity among organizations suggest that some are far less efficient than others. As per various research conducted, applying the principles of lean manufacturing to ADM can increase productivity by 20-40 %, improving the quality and speed of execution at the same time.[8] Explored the concept of waste in agile/ lean software development organizations and how it is defined, used, prioritized, reduced or eliminated in practice. The data was collected from 23 practitioners from 14 embedded software development organizations using semi-structured open-interviews.

[9]Discussed methods to ascertain waste within the software development process in Scrum teams, using JIRA tool, that supported software development planning, management and controlling activities.[10] Identified and described different types of waste in software development.

Lean Principles

So how does Toyotas Product development system helps in leaner development environment? TPS creates knowledge through broad exploration of design spaces, hands-on experimentation with multiple prototypes and regular integration meetings at which the emerging design is evaluated and decisions are made based on as much detailed information as possible. The tacit knowledge gained during both development and production is condensed into concise and useful one-page summaries that effectively make the knowledge explicit.[11]

The core Lean principles that it follows are:

- Value: understand what adds value for the customer
- Value Stream: understand how the organization generates customer value
- Flow: maximize speed and minimize cost by achieving continuous flow
- Pull: deliver value on a just-in-time basis based on actual customer demand
- Perfection: continuously improve the performance of your value streams
- Empower those who add value: delegate decision making to people who are actually involved in developing and creating processes

These principles form the basis for a large number of enabling practices, powerful tools and techniques that help enterprises maximize customer value and avoid waste.[12]

Table 2.Comparison of Wastes in Manufacturing and Wastes in ADM Process

Seven Wastes of manufacturing	Seven wastes of software development
Overproduction	Extra Features that nobody ever uses
Inventory	Partially done work - a feature for which development has started, but nobody is currently working on it
Extra Processing Steps	Software written that doesn't contribute to the final product
Motion	Finding Information, unplanned task switching
Defects	Defects Not Caught by Tests
Waiting	Waiting for decisions, instructions, and information. Ex. QA waiting for a feature to be written so they can test it, Developers idling because of incomplete information on the request
Transportation	Handoffs. Time waiting for files to copy

Lean Principle #1: Value for a Customer

The first step in lean thinking is to understand what value is for your organization and what activities and resources are absolutely necessary to create that value. The justification for a product feature should be that it adds value by helping the customer perform better in some quantifiable way, such as improving profits or sales, saving time or money. Anything that does not directly contribute value to the customer is waste. The biggest source of waste in software development is unused functionality. Only 7% of features in enterprise software applications are always used, 13% are often used, 16% are sometimes used, 19% are rarely used and 45% are never used, according to the Standish Group.

If something does not directly add value, it is waste. If there is a way to do without it, it is waste. Taiichi Ohno, the mastermind of the Toyota Production System, identified seven types of manufacturing waste. Each category of waste in manufacturing has a counterpart in ADM, as specified in above Table 2, which is quite similar to a factory that develops new applications according to business requirements. Changes to an application's requirements during development are most common source of ADM waste, causing many of the classic varieties identified in lean: designers have to rework their specifications, coders have to wait for specifications to stabilize, testers have to overproduce as their testing environments have to be set up repeatedly, and ultimately unmet requirements pile up in a large backlog. As in manufacturing, if we systematically eliminate these sources of waste, we can improve the delivery time, quality and efficiency of the ADM end product. (Table 2).

Lean Principle #2: Identify the Value Stream

Once value is identified, activities that contribute value have to be identified. These activities and their sequence is called the value stream. Then it is determined for every activity whether it contributes value to the product or service or necessary due to some other reasons. Necessary operations are defined as being a prerequisite to other value added activities or being an essential part of the business.

A value stream is the set of activities required to develop and deliver a product or service. It usually involves multiple processes, teams and departments. We avoid looking at individual functional departments such as Engineering and Marketing in isolation when considering these activities as the value stream involves everything the organization does to develop products. In contrast, with traditional approaches to performance improvement, Value Streams allow us to see product development as an integrated whole.

Surprisingly, the biggest barrier to adopting lean practices is not technical or operational feasibility but organizational issues. As processes move from one department to another a big gap often develops, especially if each department has its own set of performance measurements that are unrelated to the performance measurements of other departments because of which conceptual integrity is lost.

Conceptual integrity means that all of the parts of a software system work together to achieve a smooth, well functioning whole. Software with conceptual integrity presents the user with a single image of how a task is to be done. Conceptual integrity is achieved through continuous, detailed information flow between various technical people working on a system.[13]

To achieve this kind of integrity, people at all levels in the organization have to talk to each other, early and often. There can be no communication gap between supplier, development team, support team and customer. Everyone has to be involved in detailed discussions of the design as it develops, from the earliest days of the program. For example, Boeing gives credit for its rapid and successful development of the 777 to its 'Working Together' program, where customers, designers, suppliers, manufacturers, and support teams all meet from the beginning to design the plane. Thus, it was discovered in the initial stages that the fuel tank was beyond the reach of all existing refueling trucks, a mistake that took less time, efforts and cost to remedy at that stage. In a normal development process

this expensive mistake would not have been discovered until someone tried to fuel the first plane.[14]

Lean Principle #3: Flow

Continuous flow means that individual units of customer value move through the value stream with no time spent on non-value-added activity. If we do nothing but add value, then we should add the value in as rapid a flow as possible. If this is not done, then waste builds up in the form of inventory or transportation or extra steps or wasted motion.

Once we have discovered what our value streams actually look like, we can begin to look at how efficiently they generate customer value. The concept of flow requires us to rethink how we utilize elapsed time. How much of the time elapsed is actually spent on activities that add value to the customer?

All lean approaches focus on eliminating waste by looking at the flow of value from request to delivery. So if a customer wants something, what steps does that customer request have to go thorough to get delivered to the customer? How fast does that process flow? If a customer request waits in a queue for approval, a queue for design, a queue for development, a queue for testing and a queue for deployment, work does not flow very fast.

Flow implies a significant change of perspective. We are no longer so interested in how busy we are keeping our people, or how "efficiently" they perform individual tasks. Instead, our primary focus is on making efficient use of elapsed time. Flow is what occurs when we utilize elapsed time while adding value to the customer. Continuous flow means that individual units of customer value (a product function, for example) move through the value stream with no time spent on non-value-added activity.

In a typical organization, only 1/ 3 of the 24 hour day is spent on working, the rest is downtime. During downtime, there is no flow, because no one is working. Many companies still organize their product development processes as a sequence of distinct stages, such as concept development, requirements definition, architecture design, etc. Each of these stages results in a big batch of partially completed work. Partially complete work constitutes in-process inventory, which is a form of waste. Inventory clogs up the development pipeline, ties up resources and increases the time it takes to respond to customer needs. The answer is to distribute work into smaller units or batches. Smaller batches of work lead to less inventory, which means that less time is lost due to waiting. This is why iterative development is faster than the traditional waterfall approach.

Lean Principle #4: Pull

The idea that flow should be 'pulled' from demand is also fundamental to lean production. 'Pull' means that nothing is done unless and until a downstream process requires it. The effect of 'pull' is that production is not based on forecast, commitment is delayed until demand is present to indicate what the customer really wants. The company must make the process responsive to providing the product or service only when the customer needs it not before, not after. As in manufacturing, in software development, the key to rapid delivery is to divide the problem into small batches (increments) pulled by a customer demand and customer test.

The emphasis is to pair a skilled development team with a skilled customer team and give them the responsibility and authority to develop the system driven by customer priority and feedback. In manufacturing, the key to achieving rapid delivery is to manufacture in small batches pulled by a customer order. Similarly in software development, the key to rapid delivery is to divide the problem into small batches (increments) pulled by a customer demand and customer test. The bigger the increment of functionality you try to deliver, the longer it takes to decide what is needed and then get it developed, tested and deployed.

Lean Principle #5: Continuous Improvement

Repeated and constant attempts are made to remove non-value activity, improve flow and satisfy customer delivery needs. Though lean focuses on removing waste and improving flow, it too has some secondary effects like improved quality. The product spends less time in process, reducing the chances of obsolescence, in the world of ever-changing requirements. Simplification of processes results in reduction of variation. As the company looks at all the activities in the value stream, the system constraint is removed, and performance is improved.

The "Do It Right the First Time" rule has been widely used to develop a detailed system design before code is written. The problem with this approach is that works on the assumption that customer requirements are static and can be defined by a predetermined system. As requirements keep changing frequently throughout the life of most systems, they cannot be adequately fulfilled by a rigid design. The learning that comes from short feedback loops is critical to any process with inherent variation. The idea is not to eliminate variation but rather adapt to variation through regular feedbacks.

Lean Principle #6: Empower those who add value

This principle of Lean Production aims to drive decisions down to the lowest possible level, delegate decision-making authority to the people "on the floor", involved in process of creation. Lean methodology aims to develop engaged, thinking people at every level of the organization and most particularly at the front line. The truly lean plant has two key organizational features: "It transfers the maximum

number of tasks and responsibilities to those workers actually adding value to the car on the line, it has in place a system for detecting defects that quickly traces every problem, once discovered, to its ultimate cause."[15]

A team is not empowered unless it has the training, expertise and leadership necessary to do the job at hand. But once those elements are in place, a working team, who is involved in the day to day activities of developing software is far better equipped to make decisions than those sitting at managerial positions.

In software, the development team is in the best position to know how to respond to difficult problems and urgent requests. Requirements can be analyzed early on and their feasibility can be determined. The best way to be sure that we get things right is to work directly with customers to understand their needs, collaborate with colleagues to figure out how to meet those needs and frequently present the results to customers to be sure we are on the right track. Management's job is to supply the organization, training, expertise, leadership and information so that we generally make the right decisions, make rapid course corrections as we learn and end up with a successful outcome.[16]

Tools for Implementing Lean: 5S and Visual Management

5S and Visual Controls are workplace organizational tools that provide the necessary groundwork for workplace improvement. 5S and Visual Controls ensure that there is a place for everything and that everything is in its place clean and ready to use.

The 5S pillars, Sort (Seiri), Set in Order (Seiton), Shine (Seiso), Standardize (Seiketsu) and Sustain (Shitsuke), provide a methodology for organizing, cleaning, developing and sustaining a productive work environment. In the daily work of a company, routines that maintain organization and orderliness are essential to a smooth and efficient flow of activities. This lean method encourages workers to improve their working conditions and helps them to learn to reduce waste, unplanned downtime and in-process inventory.

A typical 5S implementation would result in significant reductions in the square footage of space needed for existing operations. It also would result in the organization of tools and materials into labeled and color coded storage locations, as well as "kits" that contain just what is needed to perform a task.

The intent of a visual management is that the whole workplace is set-up with signs, labels, color-coded markings, etc. such that anyone unfamiliar with the process can, in a matter of minutes, know what is going on, understand the process and know what is being done correctly and what is out of place.

Conclusion

Lean Software Development provides a management philosophy together with a set of practical tools for designing and delivering software-intensive products and services. These tools enable us to select design solutions, methods, design tools and organizational structures based on fitness for purpose. That purpose is to produce value for the customer with minimum waste for us.

There is a wonderful supermarket of tools, methods, and techniques from decades of progress in software engineering management. Lean does not invalidate or validate any of these. Instead, it gives us the wisdom to shop wisely and employ just the right combination of remedies needed to maximize customer value, minimize waste and produce real top-line and bottom-line results.

The lean production metaphor is a good one for software development, if it is applied in keeping with the underlying spirit of lean thinking. In the past, the application of some manufacturing concepts to software development ('Do It Right the First Time' comes to mind) may have lacked a deep understanding of what makes lean principles work. The underlying principles of eliminating waste, empowering front line workers, responding immediately to customer requests, optimizing across the value chain are fundamental to lean thinking. When applied to software development, these concepts provide a broad framework for improving software development.

The simple tenets of Lean Production and Lean thinking have brought about dramatic improvements in a myriad of industries. If applied to software development process as "Lean Software development", these practices can make highest quality, lowest cost, shortest lead-time software development possible. Sustainable information systems improvement and innovation cannot be achieved by focusing on technology alone. Successful adoption of Lean principles depends on continuously improving people, process and technology.

References

1. Bell SC, Orzen MA. Lean IT - Enabling and Sustaining Your Lean Transformation. ISBN: 978-1-4398-1756-8.
2. Boehm B. (University of Southern California) and Victor R. Basili (University of Maryland). Software Defect Reduction Top 10 List, IEEE Computer, 2001.
3. 'Lean' could be newest trend in software design Business Courier, 2003.
4. Keynote presentation, CIO Forum and Executive IT Summit, 2007 in Philadelphia as appearing in an article on searchsoftwarequality.com
5. 'Boost productivity with lean software development' by Cath Jennings, Computer weekly, 2007.
6. Direct from Dell, by Michael Dell with Catherine Fredman, Harper Business, 1999; 159.

7. Q and A with eBay's Pierre Omidyar, Business Week Online, 2001.
8. Alahyari H, Gorschek T, Berntsson R. An exploratory study of waste in software development organizations using agile or lean approaches: A multiple case study at 14 organizations. *Information and Software Technology* 2019; 105: 78-94.
9. Bufon MT, Leal AG. Method for Identification of Waste in the Process of Software Development in Agile Teams Using Lean and Scrum.
10. Uden L, Ting IH, Corchado J. Knowledge Management in Organizations. KMO. *Communications in Computer and Information Science* 2019; 1027.
11. Sedano T, Paul R. Software Development Waste. ICSE '17: Proceedings of the 39th International Conference on Software Engineering Software development waste. 130-140.
12. The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer, McGraw Hill, Jeffrey Liker, 2004.
13. Yadav R, Mittal M, Jain R. Adoption of lean principles in software development projects. *International Journal of Lean Six Sigma* 2018. https://doi.org/10.1108/IJLSS-03-2018-0031
14. Lean Software Development: An Agile Toolkit by Mary Poppendieck and Tom Poppendieck.
15. Deadline! How Premier Organizations Win the Race Against Time. Dan Carrison, AMACOM (American Management Association), 2003; 5.
16. James P, Womack, Daniel T et al. The Machine That Changed the World: The Story of Lean Production. 1990; 99.
17. Parsons M, Waugh D, Foley A et al. An Analysis of Lean Software Development. *Computer Science and Information Technology Journal* 2019; 5(1). Retrieved https://csitpub.org/index.php/csitpub/article/view/345