![telliant — Intelligent Software. Delivered.]

Applying Lean Principles to Software Product Development

Intelligent Software. Delivered

Applying Lean Principles to Software Product Development

INTRODUCTION

"Lean Principles best address frequent changes in software development with early testing and a continuous improvement cycle."

To address frequent changes in business requirements during software product development, organizations resort to having more rigid processes in place to ensure adherence to cost and time estimates. These actions result in centralization of decision making, leading to further increase rigidity and delay in decision making. Changes to requirements may not be accommodated due to the loss of flexibility and the end result is delivery of a product that lacks some of the features that are now demanded by customers. Thus, a linear software development model that is so widely adopted (like waterfall methodology) does not permit iterative development and early testing. By the time requirements changes are evaluated for implementation, changes may need to be done in design. This can be expensive and time consuming. Applying principles of lean management to software product development can help overcome some of these challenges. Lean principles also help build a continuous improvement cycle within the process of new software product development.

Lean principles focus on eliminating waste, delivering value to customers, reducing cycle times, and implementing continuous process improvement initiatives. Once Lean principles are implemented, several benefits can be realized including:

- Reduction in costs through waste elimination and compressing/eliminating waiting time
- Reduced testing cycles with early testing during the development life cycle
- Continuous process improvement leading to reduction in the number of defects identified during testing
- More flexibility in dealing with changing requirements

This paper begins with an overview of the principles of lean software development. Subsequently we will highlight how the application of Lean tools improves the product development process by examining their impact on various phases of the software product development life cycle.

telliant
Intelligent Software. Delivered.

Relating Lean Principles to Software Development

"Iterative development model in software development eliminates waste in many phases of the development life cycle. "

Mary and Tom Poppendieck enunciated the seven principles listed below for software development in their book 'Implementing Lean Software Development: From Concept to Cash':

Eliminate waste: Anything that does not add value to the customer is deemed wasteful. Lean principle advocates that organizations should learn to identify and eliminate waste. In software development, waste can include unwanted features in products, repeated cycles of testing, slack time between phases of the project (waiting for approvals or appropriate resources) and creating excess documentation.

Build in quality: Allow for integrity and testing to be carried out to provide feedback from end users to the team. If there are repeated cycles of testing, it indicates that the development process is not in order and requires to be fixed before proceeding further. Lean advocates an iterative development model that introduces testing early in the software development life cycle thereby eliminating discovery of defects late in the development life cycle and thus helps avoid costly fixes in the form of changes to design.

Create knowledge: Knowledge generation is done through feedback loops, iterations, synchronization and set based development. Feedback can be obtained from testing, getting customer response to the output and increasing the feedback loop to the development team. Iterations help obtain frequent feedback, allow the ability to "decide as late as possible," prioritize high value items early and require team planning, short complete cycles and help achieve convergence. Synchronization permits development of multiple components using architecture matrix strategy. Set based development help solutions to emerge by communicating constraints.

Lean suggests that there should be interaction of the software development team with customers to get feedback early in the life cycle of product development. Set based development implies developing multiple options, communicating constraints and discovering a solution.

Defer commitment: Allowing decisions to be made as late as possible ensures that possible solutions are not eliminated early during the product development life cycle. This includes identifying the last possible moment for making decisions and making decisions after evaluating all alternatives/options. In product

telliant
Intelligent Software. Delivered.

development life cycle, requirements are allowed to evolve over a period of time instead of insistence on "finalizing the requirements" in the beginning of the development cycle, as advocated by linear models of software development. In the idea generation phase of product development, not ruling out some of the ideas generated early can be valuable further down the process.

Deliver fast: This is required to satisfy customers who like rapid delivery. This can be accomplished by allowing people to be self-directed, reducing the cycle time, managing slack by eliminating or reducing the wait time between activities and highlighting the cost implications of delay in delivery.

Respect people: Team members need to be empowered by allowing them to determine for them the work processes and have in place a feedback loop between managers and the team members. They need to be motivated by providing them with clear, objective, and measurable goals with access to customers and management providing them support, guidance, resources and protection as required. The team also needs to have a leader to set direction, align people and enable motivation. Expertise must be fostered within teams and opportunities for continuous improvement should be provided. This can be accomplished by having a governance model that empowers the software product development team members to make decisions within the boundaries of the scope of work.

Optimize the whole: To be able to visualize the system as a whole and understand the nature of interactions between various components of the system and identify appropriate measures and metrics for performance measurements. This principle also promotes the system way of thinking during solution development. Comparing Lean Principles and Agile Methodology

On the surface, some of the Lean principles appear quite similar to what is suggested by Agile methodology of software development in that they advocate faster delivery, flexibility of accommodating changing requirements, defer commitment (keeping options open) and build quality in (advocating early testing). Lean, however has a broader view and takes the entire context of software development into account viewing Agile development methodology as supporting the practices advocated by Lean software development. Lean ideas provided a context for developing the tools that are used by Agile. Agile is often confused to mean "no documentation" whereas Agile recommends that "minimum documentation" be created keeping in mind the needs of the project. This is in line with the Lean concept of avoiding waste by doing away with excessive documentation and instead focusing on delivering software that can be tested for correctness and utility value. As described above, Lean can addresses all organizational functions and has an impact on the entire system.

telliant
Intelligent Software. Delivered.

Incorporating Lean Principles in Software Product Development

"Each Phase of Lean Development requires many sources to review, this assures the product is exactly what the client wants. "

Software product development goes through various phases from idea generation to product launch. At the end of each phase is a structured review process that results in a "Go/No Go" decision. Ideas can be sourced from various sources including employees, customers, or vendors and are screened for feasibility. Upon approval by management after a business review, a project may be created and funded for developing the product. Software product development goes through the following phases before the product launch:
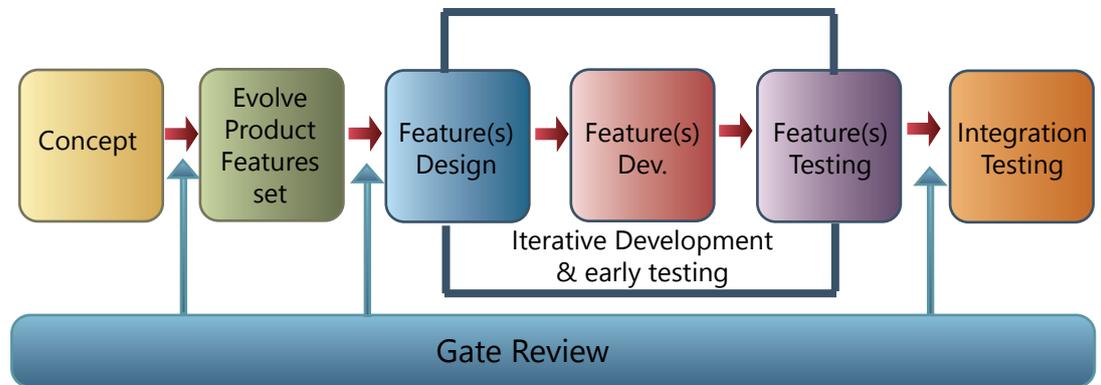
The table below shows the various phases of product development and a representative list of Lean principles that can be applied:

Phase	Application of Lean Principles
Concept	• "Wish list" of product features provided by potential customers
Product Features & Requirements identification	• Requirements evolve during the course of product development and cannot be finalized in the beginning of product development • Developing only those requirements that are required by customers (eliminating waste) • Adoption of an Iterative process that can accommodate requirements changes during product development life cycle (defer commitment)
Product Design & Architecture	• Architecture should be flexible to accommodate changing business requirements

telliant

Intelligent Software. Delivered.

Phase	Application of Lean Principles
Product Development	• Software development is done iteratively – a set of requirements is taken up for development - this follows a cycle of design, develop and test • Scenario driven development based on user stories comprising user experience, technical specifications and interaction standards • Focus on continuous improvement
Product Testing	• Build in quality – introduction of testing early in the development life cycle • Integration testing done in the end to ensure components work well together after development
Product Launch	• Customer feedback is used to enhance product features in a steady state

The table below shows the various phases of product development and a representative list of Lean principles that can be applied:

Dimension	Lean Principle Applied
Measures	• Identify easy to capture and limited number of metrics for continuous monitoring • Continuously monitor the program to identify areas of improvement
Governance	• Empower people • Promote extensive interaction between end users and development teams • Respect people by promoting self-organizing teams • Adherence to standards

telliant
Intelligent Software. Delivered.

Outcomes Realized When Applying Lean Practices Software Product Development

With the close involvement of end users and client partners the requirements are identified early reducing time and cost of development

Requirements Development

Lean advocates close interaction between the product development team and the client partners or prospective clients. With the close involvement of end users or client partners during the early stages of requirements development, only those requirements by customers that are seen as 'being valuable' are identified for elaboration, validation, and development. This results in elimination of wasted effort in developing features in the product that are not seen as valuable by customers thereby reducing time and cost of product development. Customers also need not pay for features that they do not see value in. In the early stages of product development, value is identified and hence waste also gets identified and eliminated. According to a study done by Standish Group in 2009, over 64% of product features are not used by organizations post implementation. Having well defined requirements would prevent unnecessary expenditure on implementing features that do not serve any purpose.
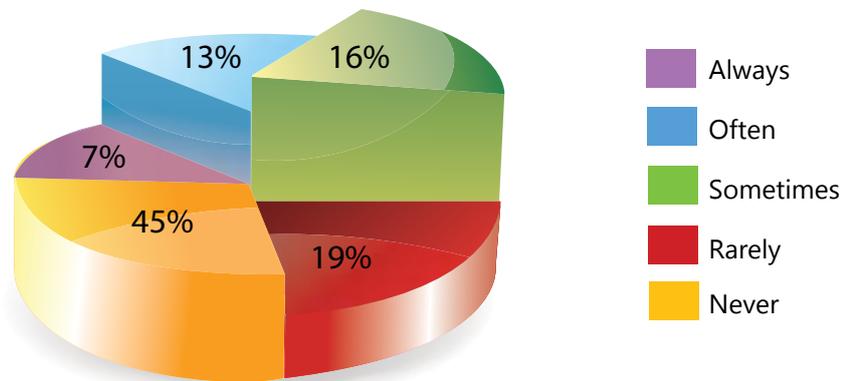
Software Features Usage, Source: The Standish Group, 2009

If requirements stability is too low, it implies that specifications are being finalized too early. Lean advocates making delayed commitments. This allows for better articulation of requirements leading to finalization at a later time, but developing a product that is more closely aligned to customer requirements.

Since potential customers or client partners are involved early on during the product development, validation of requirements happens early in the life cycle of product development. Close interactions between the business team and the technology team can result in improved collaboration and hence a better understanding of the implications of decisions.

In summary, applying Lean principles during requirements development results in the following benefits:

- Elimination of waste
- Prioritization of requirements driven by client partners/potential customers resulting in better alignment of product features with market demand
- Allowing for requirements to develop iteratively instead of 'finalizing requirements' early in the development cycle ensures that the final product being delivered is aligned to customers' expectations.

Product design and architecture

Software products need to be flexible and configurable to allow customers to respond quickly to the challenges in business. Lean principles articulate building a flexible architecture that would help development teams to be nimble footed in responding to changing business requirements. Open computing and Service Oriented Architecture are popular approaches that help implement a flexible architecture.

Adopting this approach would lower maintenance costs for customers on an ongoing basis, besides increasing the ability to support emerging business needs and lowering technology related risks. This would help organizations quickly launch new products in the market and maintain parity or competitive advantage vis-à-vis competition.

Development

Lean stresses on having short development cycles and delivering value at the end of each of these cycles. Agile methodology closely aligns itself to Lean principles. The stress is on iterative development that focuses on:

- Delivering functionality and business value in each sprint/iteration through a cycle of build, test and validation
- Addressing high risk items early to reduce risk as the development progresses
- Evolving the understanding of product requirements through each iteration

- Faster decision making (Every iteration is time bound which forces faster decision making without actually freezing scope. The unaddressed functionality in an iteration is carried over to the next iteration)
- Increasing the number of feedback loops through regular delivery of working software
- Eliminating slack between iterations by including all activities from planning through delivery as a part of the iteration instead of having additional external processes that can create wait time and cause waste to be created

In summary, applying Lean principles during requirements development results in the following benefits:

- Delivery of a software product that is in alignment with customer's requirements.
- Allows requirements to evolve during the course of product development instead of forcing finalization of product features/requirements early on in the development life cycle. The end result is a software product that is more aligned to meet customer's requirements even when there are changes to requirements from the time the product was conceptualized.
- A simultaneous focus on early risk reduction and value delivery forces identification of solution constraints early on and allows a solution to evolve taking into account all alternatives.
- Implementing functionality of highest priority first by delivering working code early in the cycle also validates the architecture, reduces technical and project risk and increases the likelihood of successful product development.
- Knowledge is created through short feedback loops achieved by early end user testing during each iteration.
- Where product development companies have teams that are not co-located, the teams are synchronized through continuous integration.
- Early testing during each iteration ensures that quality is built in early on Too many cycles of testing if run, would be a pointer to fix earlier processes. This can result in continuous improvement in the software engineering processes.

telliant
Intelligent Software. Delivered.

Product testing

Lean emphasizes early testing and continuous process improvements. The benefits of adopting this approach are:

- Reduced testing cycles and hence a reduction in overall software development costs
- Early detection of bugs/defects and resolution of the same resulting in avoiding costly fixes of defects later in the development life cycle

Applying Lean to other dimensions of software product development

Besides the different phases of software product development, Lean principles can be applied to other dimensions of software product development:

Measures: Lean advocates that metrics that are captured as a part of software development should be:

- Easily understood, collected, analyzed and be limited
- Used to monitor the program health and identify areas for improvements

Effective metrics provide the required information to management to take timely and appropriate action. Metrics collection should be automated and be used to analyze critical trends that can facilitate good program governance. While the standard Earned Value Management related metrics help determine the current status, they can also set expectations on how the budget is spent. The focus should be on collecting those metrics that are easily understood, easily collected, and acted upon.

Continuous collection, monitoring and analysis of metrics combined with iterative development helps measure the variance between quality of working code delivered and the specifications. This helps take early corrective action. Collection and dissemination of metrics can happen through project dashboard software. While corrective action can result in course correction, analysis of metrics can result in process improvements that can be implemented within the organization that is engaged in product development. This can result in process improvement initiatives.

Governance: Lean principles focus on quick decision making, empowering people, permitting project team members to have a say in the processes and adherence to standards. These are examined below in detail:

Lean principles advocate empowering people rather than try and control and directly manage them. This requires a high degree of maturity of team members who are capable of being self-directed. Team members would have the authority to allocate work (to them) that would be performed within the overall governance structure. Such a participatory approach ensures that all team members have a say in the decision making process. Team members would select their activities and tasks, commit to the work and co-ordinate regularly to ensure that progress is smooth. In combination with the iterative model of development, empowering people through self-direction works well because of the following reasons:

- In each iteration, the deliverables are agreed upon by all stakeholders and the development team.
- Iterations ensure that there is overall direction and serve as a control mechanism for the Product Development Manager to keep the team members focused on the task on hand
- Increased motivation of team members that can positively impact productivity
- Promotes greater collaboration between team members
- Decision making is decentralized and quicker because team members are empowered
- Better opportunities for team members to up skill themselves

Lean principles suggest adoption of policies and standards that would promote collaborative team work. The development team may be spread across locations, geographies, and time zones. To ensure seamless collaboration and integration, an integrated environment should be in place to ensure that there is support for interaction between project team members. Having a standardized integrated life cycle environment during the product development process helps lower costs, capture metrics, and assist easier onboarding of project resources.

To be effective, the governance model should factor in the capabilities of the project team members and the organization's environment. Successful governance models focus on enabling collaborative work between project team members and motivating them to perform at their peak levels instead of being directed.

telliant
Intelligent Software. Delivered.

Review meetings/Retrospectives:

At the end of an iteration project teams should hold review meetings to reflect on their performance. These meetings should be used to analyze 'what went off well' and 'where the team went wrong'. What went off well should be prioritized for implementation as 'best practices' and 'where we went wrong' should be used to identify 'continuous improvement initiatives'.

Outsourcing:

To take advantage of lower costs, compressing time to market and access to a scalable pool of resources, software product development firms may choose to outsource some phases of product development. Most often, this is the iterative development phase. Some of the Lean principles that can be adapted when having vendor partners include:

- Treating the project team as one team instead of dividing them as onshore and offshore partners
- Providing access to communication tools such as instant messaging, video conferencing
- Facilitate participation in team meetings by scheduling telephonic meetings scheduled with time zone overlaps
- Ensure flow of customer feedback to the offshore team
- Facilitate interaction with the onshore team by having some of the offshore team members travel onshore for a short duration during the initial iterations

While Agile focuses on continuous feedback loops from customers/end users, Lean principles advocate continuous integration, people empowerment, and continuous customer feedback. Applying Lean principles can help organizations that have outsourced some of the product development phases.

telliant
Intelligent Software. Delivered.

Summary of Benefits Derived From Adopting Lean Principles

Adopting Lean Principles enables product development requirements to evolve and members to be self-directed to upgrade their skills

Various benefits that accrue by adopting lean principles are summarized below:

- Reduced cost of product development by eliminating waste by

  o Not implementing requirements/features that are not desired by customers,

  o Adopting agile/iterative development model of software development that allows requirements to evolve during product development

  o Testing early and avoiding expensive design changes due to late testing

- More productive software development members who are self-directed and have an orientation to continuously upgrade their skills

- Developing products with a flexible architecture that can support the evolving business needs of customers and reduce the total cost of ownership to customers

- Allows for the development of multiple options in parallel

- Enables set-based development by examining all options

- Continuous process improvement achieved by introducing testing early on in the development life cycle

- Capturing metrics that are easily understood and analyzed

- Continuously delivers value/functionality addressing the high risk elements first, thereby reducing technical risk as the project progresses over time

telliant
Intelligent Software. Delivered.

Challenges of Using Lean Principles

Whereas the advantages of adopting Lean principles have been listed earlier, there are some challenges as well that are provided below:

- The project team members need to be self-directed (given the emphasis placed on empowerment) and capable of working as a cohesive unit. Having team members that are not committed can result in derailment of the project.
- Lean advocates making late decisions (compared to the insistence on 'timely sign off' in linear methodologies of software development) to ensure that all possible solutions are considered before making a choice.
- When requirements keep evolving without an end in sight, there is a large possibility of scope creep. This has to be managed well; else there would be accompanying schedule and budget slippage as well resulting in failure.
- Lean is focused on continuous improvement and the timelines for implementation can be long. The payback period for reaping the benefits would also be long. This can encounter internal resistance within the organization and the commitment of senior management would be required to overcome this.
- Some organizations have tried to adopt practices that have worked successfully in other organizations in their own organizations. The focus should instead be to understand the principles that contributed to the success rather than the practices.

CONCLUSION

The concept of Lean evolved from the manufacturing industry and has relevance to software product development as well. In fact, Lean has helped the Agile development methodology and its variants to evolve as well. Agile methodology scores over the traditional linear model of software development by not insisting on "finalizing requirements," but allows for midway course correction. In Agile method, instead of building the whole product and commencing testing, the product is developed in small incremental pieces, tested for acceptance by customers, and adjusted where required. Quality also gets built into the system early as testing commences earlier in the software development life cycle instead of being done in the end and in a time boxed manner. Integration challenges arise when there are multiple iterations being run and different teams dispersed in different locations work on a product. While Agile does not prescribe a solution to overcome integration challenges, Lean concepts help. In Lean, the work is broken down into different components and the output of one stream can lead to another and ultimately combined into the product. While multiple

iterations may be the norm in product development initiatives, Lean principles like set-based and concurrent engineering approaches help achieve synchronization.

This paper has highlighted the benefits that accrue to software product development firms during each phase of development by implementing Lean principles. Besides suggesting some principles that can be applied to the different phases of software development, Lean also suggests principles for other dimensions such as metrics and governance. Collecting limited actionable metrics, team empowerment, early feedback during product development from customers and continuous improvement processes are some of the benefits that are derived from adopting Lean processes in software development.

The Lean Software and Systems community agrees on a few principles listed below (The Lean principles are provided in brackets):

- Following a system thinking and design approach (optimize the whole)
- Respect people and encourage leadership (people empowerment)
- Using scientific methods (continuous improvements)
- Reduce flow time and waste to improve efficiency (eliminate waste)

Before embarking on the path to implementing Lean, organizations need to evaluate their internal processes and the suitability of Lean in their environment. Lean software development processes need to be tailored to the organizations where they evolve. Lean principles in software development are still evolving and given the outcomes in product development firms, the adoption is likely to increase going forward. Forays made to implement Lean principles have produced mixed results in the past and this has added to the reluctance of moving away from the traditional linear model of software development. Some of the failure can also be attributed to incorrect understanding of the principles and abandoning the effort midway without going the full distance.

Implementing Lean would entail a long term commitment. Most organizations would be following some of the principles advocated by Lean already. However, it may take organizations a long time to adopt every practice. There is no particular order of priority in which to implement them. How¬ever, the practices are synergistic and can be implemented incrementally to realize the benefits. While the ultimate aim of Lean is to 'eliminate waste altogether', practically it is impossible to achieve this.

telliant

Intelligent Software. Delivered.

In an article titled "Lean programming" in 2003, Mary Poppendieck summarizes the benefits of applying Lean principles to software development as follows: "Lean thinking applied to software development as Lean programming will lead to the highest quality, low cost, shortest lead time software development possible". These benefits that are supported by empirical evidence from different studies should spur software development organizations to adopt Lean principles, moving away from the traditional linear development models.

ABOUT TELLIANT SYSTEMS, LLC

Telliant Systems is an Information Technology services, solutions and staffing company serving businesses globally. Headquartered in Johns Creek, GA, USA, with an offshore development center in India, Telliant is committed to providing a truly different approach to outsourced software development services. Telliant provides a range of IT services including application development, software testing and performance analysis and tuning. Our services cover several software platforms including Java/J2EE, Microsoft .NET, Mobile and a variety of open source technologies including Linux, PHP, MySQL etc. Telliant works with our clients using several engagement models including dedicated offshore model, turn-key projects and onsite services and several hybrid models based on what suits the customer best. We also serve several industry segments such as banks and financial services companies, insurance, social media and healthcare industries among others.

Contact Details

Corporate Office

3180 North Point Pkwy
Suite 108
Alpharetta, GA 30005 USA

Tel: 678.892.2800
Fax: 678.892.2809
Email: info@telliant.com
website: www.telliant.com

telliant
Intelligent Software. Delivered.