- 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.

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.

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.

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
  - Not implementing requirements/features that are not desired by customers,
  - Adopting agile/iterative development model of software development that allows requirements to evolve during product development
  - 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

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.

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.