

## An Efficient and Well-organized Lean Software Development Model

Shumaila Iram, Raheela Nasim

Department of Computer Science, University of Agriculture, Faisalabad, Pakistan

### Abstract

*In the growing field of software engineering the new patterns and models were being introduced day by day to increase the effectiveness of software development process. In cost decreasing and quick delivery the Lean Software Development was getting attention of the industry. Several studies and research was conducted in lean software development. Recently agile software development using lean principles had been specified noticeable consideration. However an entire software development cycle which satisfying and specifying all the lean principles had not been proposed until now. It was the art of research to initiate a new "Lean "software development model which contained all the development phases. The goal was to recommend a new Life cycle model; an entire cycle with elevated goals and requirement change control as well as requirement tractability. The definition of a well-organized and efficient model by reducing the time and managing the project cost in maintenance process was the purpose of this research. This model was developed to deliver the project quicker, to enhance the maintainability and effectiveness at some stage in and after the project development.*

**Key words:** LEAN; Efficient model; LSD; Lean Vs Agile

**Corresponding author's email:** [iram\\_shumaila24@yahoo.com](mailto:iram_shumaila24@yahoo.com)

### INTRODUCTION

The term agile stands for 'moving quickly'. Agile methodology has an adaptive team which is able to react to the changing requirements. Customer satisfaction by quick delivery of useful software. Agile software development is a collection of software development methods in which requirements and solutions develop through teamwork between self-organizing, cross-functional teams. It increases the adaptive planning, evolutionary development, before time delivery, uninterrupted improvement, and encourages quick and elastic response to change (Sharma et al., 2011).

Iterative development is a mixture of both iterative design and incremental build model for any software development. Iterative development is a method of breaking down the software development of a huge application into small iterations. In iterative development, features are designed, developed and tested in frequent cycles. In each iteration, additional features can be developed and tested until there is a complete functional software application prepared to be delivered to customers. If requirement changes repeatedly and smaller projects, deploy product in short period time with accomplished resources then we can select iterative method. Iterative method of software development stresses quick iterations, small and repeated releases, and change requirements facilitated (Jorge et al., 2011).

Lean means elimination of waste. Lean develops process speed by eliminating waste. Lean means continuous improvement. Lean Software Development has the goal of giving as much value to your customer as rapidly as possible in the well-organized manner possible. The keywords here are speed and low cost. Lean focus on wastages during the creation and development waste means to combine and describe those things that does not give and incorporated in construction and development of customer value. They don't improve the customer's value. Lean is to decrease waste and increase effectiveness of lean principles and the agile practices are intended over a squat time period to execute projects, with customer interaction constantly and talented for making change at any time during the product development. When lean software development applied in the approved manner then outcomes in high quality software that is developed at the minimum possible cost and rapidly (Poppendieck, 2007).

There is constantly a risk in the development process, on the other hand, if companies 'acquire Lean principles in the development they will obtain rapid wins. Lean software development quickly functioning on spending time for value by paying attention on creating value for the customer and not paying consideration on those activities that do not produce value. By applying the lean principles companies get the higher satisfaction from the customers about the products quality. There are too

much examples in which when the companies had acquired Lean than their investments were high (Harris, 2004).

There are two main techniques of lean development that makes change simple. First to construct the tests throughout the development process for detecting the errors whenever the procedure is broken down it showed errors and blocked there until error detect and removed as well as it necessity “construct tests” at variety of stages of the process development. Unit and regression tests apply throughout any change of the process. Refactoring is another method that Improves the design without altering its functionality, or improving the design of existing software in a restricted and quick way (Kumar, 2005).

The significance of traceability is too much important. If tracing is done rightly than as a result product quality will increase and less time is required to complete the project. The majority of the work with tracing will be completed by the team and one of the issues that are related to this is only one write but read many. If several times there is require to change the requirements then tracing the requirements is always useful (Jakobsson, 2009).

A complete development cycle fulfilling all the lean principles has not been proposed yet. A complete cycle with high goals and requirement tractability as well as requirement change control is not introduced yet. This research is related to build up an efficient model based on quality research and its actual world practice also. The model is quick and well-organized by implementing the lean principles and all the steps those are essential to build up complete software including traceability and control change principle.

**MATERIALS AND METHODS**

For this research qualitative and quantitative both approaches are used. Initially a literature review is given together information how to build up a model and to prove the model. A model contains all the development phases (requirement specification and analysis, design, coding, system testing and maintenance and deployment) of a software development life cycle as well as requirement change principle and traceability.

Lean principles are client centric, such companies those uses the Lean principles with development models frequently work with their clients to find out what clients requires, the improvements and how new features can be further intended, designed and implemented to assemble their desires needs. Six-sigma methods are an also Lean principle that not only decrease waste and inefficiencies, but also concentrate on constant improvements (Raffo et al.,

2010).

The entire functional and non-functional requirements are collected from users. The data meeting approaches include the questionnaire, surveys, and interviews are conducted. Model confirms according to the users outcome and the information gathered by them. At the last statistically analysis is applied on the proposed model to uncover the effectiveness of the proposed model. Iterations and increments are use in the model while moving from chronological to parallel implementations could enhance the speed of development and time saving (Maglyas et al., 2012).

An Efficient and well organized software development model have six sigma of lean joint with software development life cycle. Finally a survey report after doing analysis is given in which real world practices is also reported.

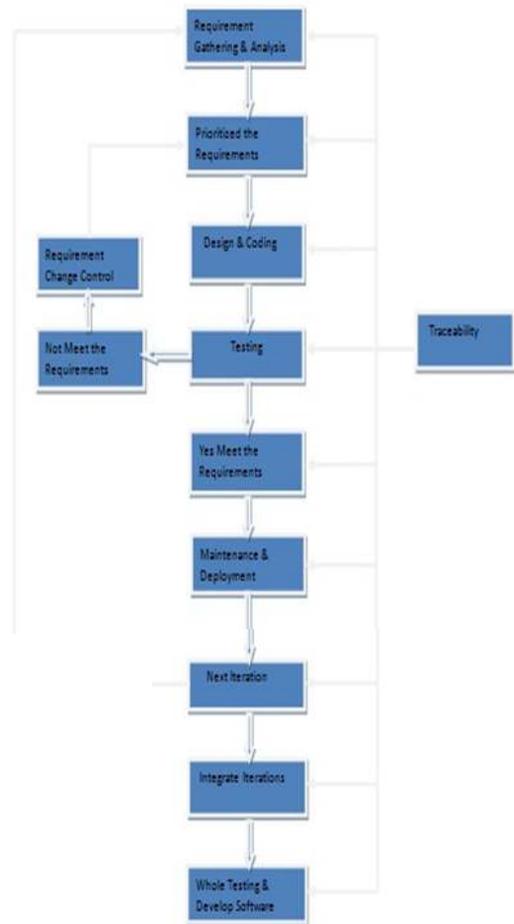
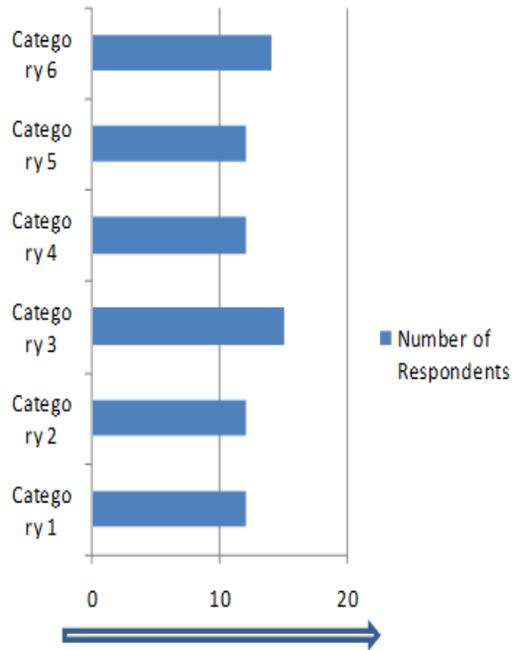


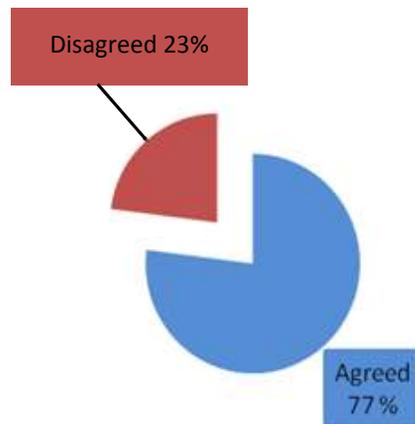
Figure1: LSD-Model.

**RESULTS AND DISCUSSION**

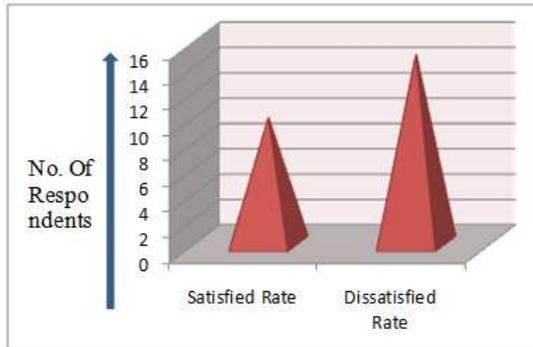
Survey method is a quantitative information gathering procedure about items in any population (Rodriguez et al., 2012). Survey result provides now highly developed survey software for all forms of survey research, as well as: online surveys, paper surveys, and the most modern is mobile surveys (Trimble and Webster, 2013). There are several of the remunerations of survey research like decreased cost, wide-ranging material and flexible. Within the software practitioner’s organization the survey was planned to illustrate the level of usage (Runeson et al., 2009) of lean technique, principles and practices as well as to discover the causes for implementing lean software development, and effects for the reason that of the adoption. The survey questionnaire including ten questions, it was an online survey the consequences are being collected during online from five diverse organizations and 30 people giving response to the questionnaire, it was intended based on the working of lean software development. If to be valid Lean Model for the development of software, The survey questions consisted on the surroundings information of organization, usage of lean techniques and its benefits, advantages of using lean practices and principles and objectives its subsequent to causes in adoption and the respondents who agreed to apply lean and who are not. Surroundings information of the practitioners and their organization consisted of the subsequent questions: Location in the organization, years of experience in software development, quantity of employees in the organization, representative team size. Since taking data from five distinct software organizations and the experience of employees and their locations/positions. Usage of lean software development technique in the respondents organizational division was account by 77% (n=23). The usage of lean software development techniques would be facilitated held by 77% because they agreed on about more than 50% remunerations of lean. And other lifted were not agreed or not sure and those were about 23%.



**Figure 2:** Number of respondents who were usual that quality characteristics about LSD-Model.



**Figure 3:** Agreed and disagreed Ratio



**Figure 4:** Satisfaction rate between number of respondents from traditional development model.



**Figure 5:** Lean according to respondents.

## CONCLUSION

Built a well-organized and an efficient model comprised quality research and also its actual world practice. This model is quick and well-organized by implementing all the phases which are essential to build up entire software and implementing the lean principles as well as control change principle and traceability.

## REFERENCES

- Sharma A and Lehman J. (2011). Software Development as a Service: Agile Experiences. Annual SRII Global Conference, Institute of Electrical and Electronics Engineers (IEEE) 749-758.
- Jorge AO, Chaudron MRV and Heijstek W. (2011). Moving From Waterfall to Iterative Development: An Empirical Evaluation of Advantages, Disadvantages and Risks of RUP. 37th EUROMICRO Conference on Software Engineering and Advanced Applications,

Institute of Electrical and Electronics Engineers (IEEE) 453-460.

- Poppendieck M. (2007). Lean software development. Companion to the proceedings of the 29th International. Journal of Economica. Seria Management 162-170.
- Harris A. (2004). Lessons in lean [lean manufacturing]. Manufacturing Engineer 83 (5):16-19.
- Kumar D. (2005). The Critical Success Factors in applying Lean Software Development in Virtual Teams. Master Thesis, Department of Economics and Business, University of Amsterdam. 1-55.
- Jakobsson M. (2009). Implementing traceability in agile software development. Master Thesis, Department of Computer Science, Lund Institute of Technology Sweden 1-496.
- Hines P, Holweg M and Rich N. (2004). Learning to evolve: a review of contemporary lean thinking. International Journal of Operations & Production Management 24(10):994-1011.
- Li S, Subba SR, Nathan TSR and Nathan BR. (2005). Development and validation of a measurement instrument for studying supply chain management practices. Journal of Operations Management 23(6):618-641.
- Alwardt A, Mikeska. N, Pandorf R and Tarpley P. (2009). A lean approach to designing for software testability: Proceedings of the IEEE AUTOTESTCON 09 Systems Readiness Technology Conference IEEE. 178-183.
- Jailia E, Sujata M, Jailia M and Agarwal M. (2011). Lean software development (As a Survival Tool in Recession). International Journal of Software Engineering & Its Applications 5 (3):63-70.
- Wang X, Convoy K and Cawley O. (2012). Leagile software development: An experience report analysis of the application of lean approaches in agile software development. Journal of Systems and Software 85(6):1287-1299.

- Nord RL, Ozkaya I and Sangwan R. (2012). Making architecture visible to improve flow management in lean software development. Institute of Electrical and Electronics Engineers (IEEE) Software 29 (5):33-39.
- Ebert C, Abrahamsson P and Oza N. (2012). Lean Software Development. Institute of Electrical and Electronics Engineers (IEEE) Software 29(5):22-25.
- Jonsson H, Larsson S and Punnekkat S. (2013). Synthesizing a Comprehensive Framework for Lean Software Development. Software Engineering and Advanced Applications (SEAA). 1-8.
- Pilar R, Pasi K, Markku O. 2014. Combining Lean Thinking and Agile Methods for Software Development. 2014 47th Hawaii International Conference on System Science, Institute of Electrical and Electronics Engineers (IEEE). 4770 - 4779.
- Raffo D, Mehta M, Anderson DJ and Harmon R. 2010. Integrating Lean principles with value based software engineering. Technology Management for Global Economic Growth (PICMET), Proceedings of PICMET'10: IEEE. 1-10.
- Maglyas A, Nikula U and Smolander K. (2012). Lean Solutions to Software Product Management Problems. Institute of Electrical and Electronics Engineers (IEEE) software 29(5):40-46.
- Rodriguez P, Markkula J, Oivo M, Turula K. (2012). Survey on Agile and Lean Usage in Finnish Software Industry. ESEM 4(3):139-148.
- Trimble J and Webster C. (2013). From Traditional, to Lean, to Agile Development: Finding the Optimal Software Engineering Cycle. 46th Hawaii International Conference on System Sciences (HICSS). 4826-4833.
- Runeson P and Host M. (2009). Guidelines for conducting and reporting case study research in software engineering. Emp. Software Engineers 14(2):131-164.