

# AN APPROACH FOR USING LEAN PROCESS IN AGILE SOFTWARE DEVELOPMENT

A. Pathanjali Sastri<sup>1</sup> and K. Nageswara Rao<sup>2</sup>

**Abstract-** The paper focuses on the critical components which can be considered and also the instinctive use of minimal process with few artifacts while maintaining the required discipline for project success. Additional practices or tailoring may be needed to complement the agile methods and merged into some of the existing software engineering methods without reducing the focus on quality. This paper addresses the key issues and further discusses the factors behind agile methods and proposes various methods that can develop agile techniques successfully fit into thereby improving the quality of the developed products with lean process.  
**Keywords –** *Agility, Agile Methodology, Project Scope, Lean Process, Reusability*

## I. INTRODUCTION

Recent studies [1]-[5], though are informative, providing insight as to what might be done to accomplish expected practices have dismissed the risks factors that are associated with the agility. The risks evaluation will encourage them to identify alternative practices must clearly and unambiguously accomplish a result that meets the goals of the project and organization. Due to the theory that stands behind the standards and guidelines of agility [6], many practitioners within the industry have concerns as what the impact of agile development will have on the continued use and success these standards will have. Agile Development brings with it some new ways of thinking as it is people-oriented not process-oriented. The following table identifies strengths and weaknesses of current processes and reveals agile development/acquisition risks.

## II. ISSUES THAT AN AGILE MATURITY MODEL SHOULD ADDRESS

Let us discuss common problems with process documentation, and aspects of using process documentation in Agile Model. The following (Table I) are critical issues that the agile methodology should address.

TABLE

PROJECT RISKS IN AGILE PROJECTS

I

| Risk                               | Category      | Impact on the Project (Probability) |
|------------------------------------|---------------|-------------------------------------|
| Too much of informal communication | Communication | High                                |

<sup>1</sup> Department of Computer Science & Engineering Rayalaseema University, Kurnool, Andhra Pradesh, India

<sup>2</sup> Potti Sriramulu Chalavadi Mallikharjuna Rao College of Engineering and Technology Vijayawada, Andhra Pradesh, India

|   |                         |        |
|---|-------------------------|--------|
| Reporting is too formal   |                         | Medium |
| Inexperienced project management may lack the necessary skills to maximize the efforts of the team.   | Management              | High   |
| No formal guidelines and standards may lead to inconsistent deliverables and poor quality   |                         | High   |
| Poor matrix management results because of split in the time and efforts of the associates across multiple modules and may report to more than one project leader, leading to lack of focus and dedication to the project.       |                         | Medium |
| No proper change control can lead to uncontrolled scope creep.  | Process                 | High   |
| No proper documented development process results in poor project making planning and coordination difficult.  |                         | High   |
| No formal quality processes result in poor quality deliverables.  |                         | High   |
| Frequent change to existing statement of requirements affects the scope and scale of the project which is not well-defined or clear.  | Requirements Management | High   |
| The details of the requirements that are signed off by the customer and are subject to continual change   |                         | High   |
| Scope creep that incorporates new requirements after initial requirements definition and agreed schedules. The project plan and schedules lead to an increase in budget or timescales as a result of any unforeseen activities. |                         | High   |
| Poor documentation leads to insufficient planning activities, planning of reusable tools and new & untried technologies resulting in difficulties of team   | Documentation           | High   |

|   |  |  |
|---|--|--|
| coordination, poor communication and making it more difficult to understand the impact of change which is a time consuming process. |  |  |
|---|--|--|

### III. POOR DEFINITION OF PROJECT SCOPE AND OBJECTIVES

Active user involvement and also involvement from a technical user representative in the business, creates an environment that fosters close collaboration and cooperation to make a business case for agile projects. Due to their agile (i.e. unpredictable) nature, it is recommended that the principles of incremental model can help to surface issues in early iterations of the software. As it is practically impossible to plan every detail of the product to be developed up front, frequent delivery of iterations and continuous testing [7][8][9][10] can help to get an initial solution out and get real customer feedback to inform the priorities for further development.

### IV. POOR ESTIMATION

Most of the traditional projects force project end dates despite best estimates due to the inclusion of non-essential requirements at the outset and difficulty to visualize the solution too early in a project [7]. The same is true with agile methods where tasks should be broken down into micro pieces and the progress is measured on a daily basis. The 'velocity' is calculated as the number of estimated hours delivered only when a feature is 100% complete [7]. Further, it is essential for constant collaboration of the users and developers keep this approach self correcting and resulting in the maintenance of the accuracy of estimating practically difficult. So, the theory of keeping agile requirements lightweight and visual [8] can be issues as the projects have to be stuck to the broad scope of the project. The project members' commitment to the requirements gathering cannot be decided in the planning phase. In addition new iterations invite opportunities to make adjustments in unfamiliar or changing technologies or lack of required technical skills [7][8][9].

### V. QUALITY ASSURANCE FOR KEY DELIVERABLES

Working software is the key measure of quality. Though agile principles help us to surface issues early, frequent delivery of iterations and continuous testing can help to mitigate the risk of the identifying issues, which may otherwise go unnoticed until much later in the project.

### VI. ALIGNMENT OF EXISTING AGILE ACTIVITIES AND ESTABLISH A VISION FOR AN AGILE MATURITY MODEL.

The following activities (Table II) of a software development including their contributions (%) are listed below.

TABLE II  
PROJECT ACTIVITIES AND THEIR CONTRIBUTIONS (%) IN SDLC

II

| Activities         | Percentage |
|--------------------|------------|
| Quality Activities | 10%        |
| Documentation      | 25%        |

|                    |     |
|--------------------|-----|
| Meetings           | 20% |
| Project management | 5%  |
| Coding & Testing   | 25% |
| Rework Activities  | 15% |

The core idea is to maximize **customer value** while minimizing non-value activities. Lean and Agile are both methods of improving the effectiveness and performance of work processes. Lean primarily taken from Toyota Production System, Agile methodology has borrowed heavily from Lean thinking and practices. Agile methods and principles are just management tools. The analysis of the goals and practices of agile methodology only act as guidelines. So, the following adjustments need to be made as a part of possible alternative ways of implementing the goals.

- a. Documentation and coding & Testing[10] is the most expensive and time consuming activity
- b. Meetings and communications is the second most expensive consuming activity
- c. Rework activities is the third most expensive and time consuming activity
- d. Quality Activities is the fourth most time consuming activity
- e. Project management is the fifth most time consuming activity

From the above figures, it is obvious that rework activities defects have to be prevented and avoided to some extent. It is also obvious that documentation costs have to be controlled, and this leads directly to the concept of reusing portions of specifications, designs, user manuals, and other paper-based documents.

By improving reusability [11] for design, coding, and testing artifacts by more than 50%, the development activities are reduced by 50% from requirements until delivery and the development productivity rates are improved by more than 50% thus avoiding schedule slippage. At the same time the development costs and maintenance costs by reduced by more than 50% thus avoiding cost over runs. For any new projects the reusability for design should approach at least 40%, code and test by at least 50% and documentation of planning, design, manuals, and other paper materials by at least 60%. At the same time, we need to ensure that no more than 5% schedule slippage is allowed and no more than 5% cost over runs is allowed.

In order to ensure that proper project management we need to allocate 15 days a year for training in software management topics for using better project management tools and methods, and especially for better planning and estimating methods. We have to allocate at least 15 days a year for training in software process improvement topics to ensure successful process improvement program at organization level. Lean process means creating more value for customers with available or adequate resources. A lean process should consider customer value and at the same time focuses its key processes to continuously improve it. To accomplish this, lean thinking changes its focus to optimization of suitable tools or technologies.

## VII. CONCLUSION

Lean process is aimed at creating 'value' and results. Process documentation is very important component of software development. On the other hand there is a need for agile projects to respond to changing customer requirements or business requirements when project requirements

incrementally change. So, those changing requirements, design, etc. have to be documented to develop high quality, low cost, and within short sprint time and within budget. Though the success or failure of implementing agile methodologies has nothing to do with documentation, it does not mean that no documentation is required.

The following are the actions needed to address the deficiencies and if changed will meet the objectives.

- a. Motivation for improving documentation
- b. Moving towards Lean Processes
- c. Process documentation usually
- d. Maintaining good definition and writing principles.
- e. Processes need to be defined by well thought out diagrams or models.

## REFERENCES

- [1] Jerzy Kisielnicki, Anna Maria Misiak, "Effectiveness of Agile Implementation Methods in Business Intelligence Projects from an End-user Perspective," *Informing Science: the International Journal of an Emerging Transdiscipline*, vol. 19, pp. 161-172, 2016.
- [2] Santhosh John, Nazaraf Shah, and Leonid Smalov, "Incremental and Iterative Agile Methodology (IIAM):Hybrid Approach for Ontology Design towards Semantic Web Based Educational Systems Development," *International Journal of Knowledge Engineering*, Vol. 2, No. 1, pp:13-19, March 2016.
- [3] Jyothi, Veerapaneni Esther; Srikanth, Kaitepalli; Rao, K Nageswara, "EFFECTIVE IMPLEMENTATION OF AGILE PRACTICES - OBJECT ORIENTED METRICS TOOL TO IMPROVE SOFTWARE QUALITY," *International Journal of Software Engineering & Applications*, Vol.3, No.4, pp:13-24, July 2012.
- [4] V. Esther Jyothi, K. Nageswara Rao, "Effective Implementation of Agile Practices – A Collaborative and Innovative Framework," *International Journal of Software Engineering & Applications (IJSEA)*, Vol.3, No.4, pp: 41-48, July 2012.
- [5] Jyothi, Veerapaneni Esther; Rao, K Nageswara, "Effective Implementation of Agile Practices - Incoordination with Lean Kanban," *International Journal on Computer Science and Engineering*, ISSN : 0975-3397, Vol. 4, No. 01, pp:87-91, January 2012.
- [6] MALIK HNEIF, SIEW HOCK OW, "REVIEW OF AGILE METHODOLOGIES IN SOFTWARE DEVELOPMENT," *International Journal of Research and Reviews in Applied Sciences*, ISSN: 2076-734X, EISSN: 2076-7366, Volume 1, Issue 1, pp:1-8, October 2009.
- [7] Mihai Liviu DESPA, "Comparative study on software development methodologies," *Database Systems Journal*, vol. V, no. 3, pp: 37-56, 2014.
- [8] Muhammad Amir, Khalid Khan, Adnan Khan and M.N.A. Khan, "An Appraisal of Agile Software Development Process," *International Journal of Advanced Science and Technology*, Vol.58, pp:75-86, 2013.
- [9] Hassan Hajjdiab and Al Shaima Taleb, "Adopting Agile Software Development: Issues and Challenges," *International Journal of Managing Value and Supply Chains (IJMVSC)*, Vol. 2, No. 3, pp:1-10, September 2011.
- [10] B. K. Madhu, Megha Jigalur and V. Lokesh, "A study on Agile Software Testing: Emergence and techniques," *African Journal of Mathematics and Computer Science Research*, ISSN 2006-9731, Vol. 3(11), pp. 288-289, November 2010.
- [11] Naresh Kumar Nagwani and Pradeep Singh, "An Agile Methodology Based Model for Change-Oriented Software Engineering," *International Journal of Recent Trends in Engineering*, Vol 1, No. 1, May 2009.