Effective Implementation of Agile Practices – Incoordination with Lean Kanban

Veerapaneni Esther Jyothi
Assistant Professor, Department of Computer Applications,
V.R.Siddhartha Engineering College,
Kanuru, Vijayawada – 7, Andhra Pradesh, India.

K. Nageswara Rao
Professor & Head, Department of Computer Science and Engineering,
P.V.P.Siddhartha Institute of Technology,
Kanuru, Vijayawada – 7, Andhra Pradesh, India.

***Abstract* — Increased productivity, return on investment and decreased defect rate are the major concerns in now-a days. Agile software development methodologies are in their way into the software mainstream. Agility created revolution in various fields of software development especially in mobile-based applications and internet-based applications development because of the improved quality of the software product. This research paper is the result of immense research work on the implementation of agile software development in coordination with lean kanban which facilitates continuous improvement. This paper portrays the key practices of iterative agile and lean kanban and also focuses on the approaches that can be considered to reduce the defect rate in the software development and increase the productivity.**

Keywords-defect rate, kanban, quality assurance, agile

I. INTRODUCTION

The focal thought behind developing high quality software depends on the techniques followed in the preliminary stage; the approach of agile comes into lime light where the QA actions become an essential part from the outset of the improvement activities itself. The proficient elimination of overproduction is the focal thought behind Kanban which improve the performance. Hence Kanban is fundamentally an implication of taking organization beyond all the measures of production by representing all requirements from each facet of production. It assists the soft flow of manufacturing. The kanban implementation follows pull system rather than push system. The word Pull in lean manufacturing portrays the demand of the customer for product. Pull system (make-to-order) currently regularly substitute push system (make-to-stock), which usually have been based on the expectation of demand. Push systems, which are profoundly contingent on the scheduling of demand (and manufacture ability identical), are habitually portrayed by a high level of in-progress collections, which are normally heavy and expensive to uphold.

The paper is organized as follows. Section II describes the key factors of iterative agile software development process. Section III explains the kanban way of software development process, Section IV discusses the approaches that can be considered while implementing iterative agile in coordination with lean kanban method and finally section V discusses on the issues to produce defect free product.

II. ITERATIVE AGILE

User involvement is given high priority in the working style of agile, drawing user's right in to the heart the development process. The development of functionality is based on the user's desire. Agile processes is the discussing point here in this paper regarding how we can reduce development risk and avoid last minute surprises [7]. Defect rate can be lowered with phased integration and continuous user feedback. Despite being moving towards improved quality software, this also enables the principle of small iterative incremental releases which helps us to attain quicker ROI through quick release of business functionality [4]. While transferring to agile methodology from the traditional pattern it produced the benefits above expectations which replicated in the decrease in fault rate as well as producing high quality software.

One of the advantages of adopting an agile software development approach is a considerable raise in software quality. Amazingly there are two kinds of benefits for the customers who try agile. The first one is Quality and the second benefit is lower defect rates. Agile methods are a rising movement in the software field [5]. This realistic, people oriented method to software development demands software practitioners. Several adopters experienced enrichments in competence, superiority, work inspiration and consumer satisfaction [8].

The Agile methodology can be summarized as below

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.
- Deliver product based on need from high priority down.
- Eliminating waste wherever possible.
- Refactoring, pair programming and frequent integration are the key factors [3].
- Interactive, incremental and test driven development.

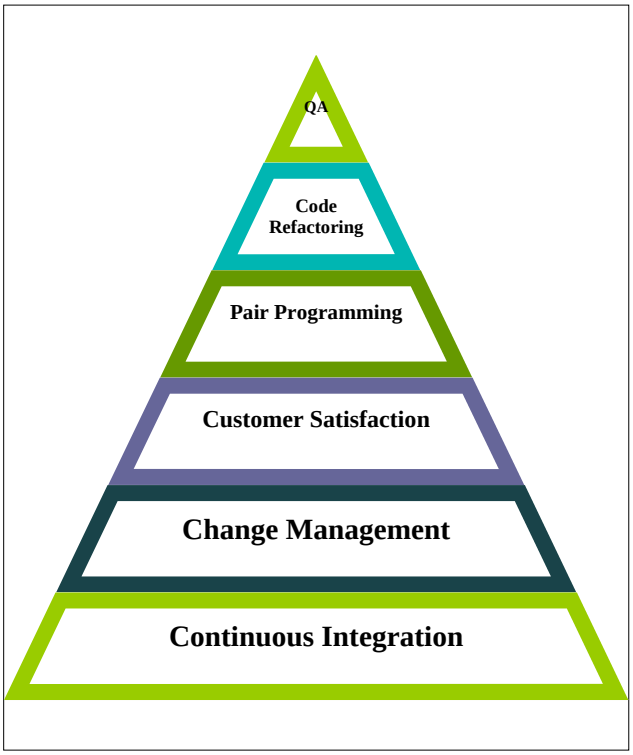

Figure 1. Best practices of Agile software development methodologies

III. LEAN KANBAN

Kanban is a Japanese production technique that uses cards or electronic signals in a computer system to monitor and control workflow in a factory. The kanban system was first developed by Toyota in Japan. The kanban technique uses a “pull” method where a signal is sent to pull the material into the production area when it is needed as part of the manufacturing process.

There five qualities that Kanban has According to David Anderson, author of Kanban – Effective Evolutionary Transform for your Know-how Home business. Primarily, it makes the workflow visible. Secondly, it confines work in progress. Kanban is a primary element of a pull procedure. Pull structure indicates that production is only set in action when there is a demand or purchase in place. Third, it manages flow. It makes sure that the progress of function by way of each phase and aspect of workflow is observed, calculated and recorded. Afterwards let the management appropriately verify future enhancements and constantly build the

system. Fourth, it makes system strategies open. The "visibility" accomplished by way of Kanban permits executives and employees to speak about new explicit policies, permitting them to understand more and to make improvements through a far more balanced way as all bases are put on record.

Eventually, it allows mutual development. Kanban supports Kaizen, or the practice of recognizing and applying little but concrete improvements that ends in evolutionary amendments. These traits are the result of thriving accomplishment. Potential execution, on the other hand, can be attained to start with as a outcome of instruction.

The Kanban method guarantees that manufacture systems react to the genuine requirements, ultimately of its consumers, and not to guesses or presumptions of order. The Kanban method will activate the routine movement, production or supply of products or parts, when inventory levels fall to lowest. Companies that utilize lean manufacturing skills aim their functions to be receptive to the ever-changing level of consumer order. Those companies that are capable of making customer order need not to support the levels of inventory that conventional group make-to-stock producers depend up on. The consistent supply of item in direct reaction to customer orders will produce confidence in the consumer base and persuade recur business.

Advantages of Pull/Kanban method implementation

- Pull/Kanban techniques training create a more efficient, effective operation and Quality improvement.
- Increase customer satisfaction and retention.
- Increases profit margin and increase turnover.
- Promote international trade.
- Improve employee motivation, awareness and morale.
- Reduce cost and increases productivity.
- Provides quick responses to change.
- Avoids over-production and eliminates wastes.

The lean method builds excellence in to the system so that any diversions from quality will quickly end the production line [6]. The quality diversion can be usually identified by the system, and showed to the employees, who will be answerable for.

IV. INEGRATED APPROACH

As the years are been passed by, the software development teams have been taken transition from waterfall team to a well tuned and simple agile team. And now in the recent days agile teams are incorporating kanban principles to maintain continuous improvement. For this the team's mindset has to be changed in same aspects during the development process. One such important factor is that the team should not just deliver code but should deliver tested and completed product. So, this type of transitions plays major role in the increased productivity. The lean way of development methodology started out to justify the agile way of development methodology but now it has come up in its own way.

Irrespective of the method of agilility the teams / organizations follow the transition from agile to integrate agile & kanban teams can take place by incorporating the following approaches:

1. The concept of limiting work in progress (WIP) and the importance of flow through the system are to be the key factors to be taken care of while the product is under development.
2. The product owner team has to marshal all incoming work and help maintain a regular schedule of meetings to breakout user stories with the help of development team.
3. The standing meetings have to be conducted once in a week for story planning and estimation where in the development team should also be involved with product owner so that the teams will work in synchronization.
4. The kanban board can be used by the kanban teams instead of agile story boards which maintain the work items. Each work item corresponds to the user story.
5. As the daily standup meetings and iterative planning meetings are the necessary activities to be followed by the iterative agile teams, the kanban teams should also schedule 15 minutes for daily standup meeting [9].
6. Usually the iterative agile development teams are responsible for large-scale projects but the kanban teams can be effective while working for small –scale project along with bugs fixing.

For the effective work distribution among the agile and kanban teams to work in coordination, the workflow should follow the sorter analogy for process.

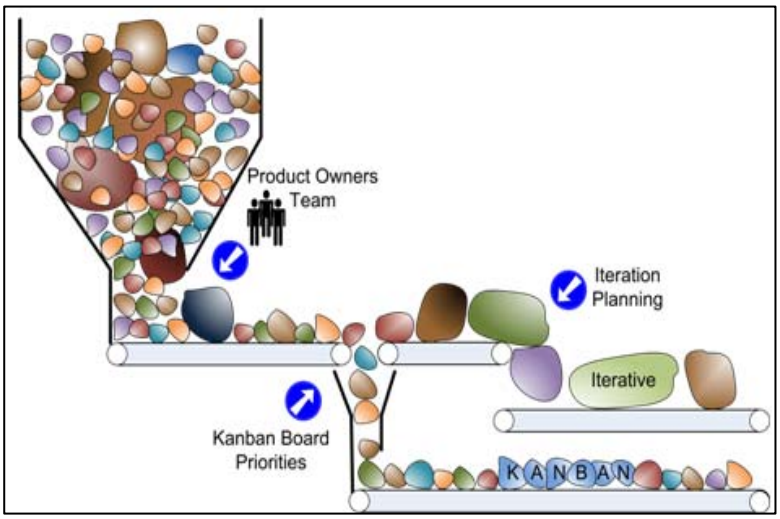

Figure 2. The Sorter analogy for the workflow process

V. QUALITY ASSURANCE

Mc Breen [1] defines agile quality assurance as the development of software that can respond to change, as the customer requires it to change. Beck [2] defines agile software quality interims of efficiency, timeliness, cost effective, ease of use, maintainability, integrity, robustness, extendibility and reusability.

TABLE I. SOFTWARE QUALITY FACTORS IN SCRUM, XP, DSDM AND RUP

SOFTWARE QUALITY FACTORS	SCRUM	XP	DSDM	RUP
Efficiency	Sprints	Pair Programming	Feasibility study	Use case modeling
Timeliness	Iterative life cycles	Incremental delivery	Incremental	Incremental
Cost Effectiveness	Feedback loop approach	Iterative Small releases	Iterative and frequent delivery	Iterative development
Maintainability	Iterative development	Incremental development	Iterative development	Iterative development
Integrity	Graphical workflow	Object-oriented	Architecture driven	Component based
Testability	Parallel with development	Test driven development	Integrated throughout	Unit testing
Reusability	Refactoring	Refactoring	Object-oriented UML patterns	Software patterns

Some of the key factors that can influence quality assurance activities are as follows.

1. The turnaround time of bug fixes can be reduced by the early identification of defects.
2. Better functionality test cases should be formulated.
3. To ensure that standard processes are in place and traceability is maintained, QA is to be tasked perfectly and effectively.
4. Doing it right the first time keeps the maintenance cost low and learn how to make the most of the testing process [10].
5. Reduce the time of regression testing by proper implementation & maintenance of traceability matrix.
6. Effective test automation suits can be used for better and faster regression testing.

One of the key findings of agile software development is that the quality assurance activities should become a part from the kick off of the development activities itself. To make the agile more efficient the whole agile team should be responsible for quality [1]. Every role in the team should involve in testing activities to produce defect free product

VI. CONCLUSION

In this research paper the transition from traditional software development to the iterative agile way of development and now to the integrated implementation of iterative agile incordination with lean kanban has been discussed. Various approaches has been discussed to improve the productivity, return on investment and also to produce a defect free product or software. Future research should address the metrics for the integrated implementation approach although there are limited metrics in practice.

REFERENCES

[1] Mc Breen, P: Quality Assurance and Testing in Agile Projects. Mc Breen consulting, 2003. http;// www.mcbreen.ab.ca/talks/CAMUG.pdf
[2] Beck, K : Extreme Programming explained : Embrace change. Second edition Kent Beck Publisher. 2000.
[3] E. Arisholm, Gallis, H.E., Dybå, T., Sjøberg, D., Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise", IEEE Transactions in Software Engineering, pp. 65-86, 2007.
[4] Alleman, Glen. Agile Program Management: Moving from Principles to Practice. Agile Product & Project Management, Vol. 6 No. 9, Cutter Consortium, September, 2005.
[5] Schwaber, K. and Beedle, M. (2002). Agile Software Development with Scrum. Upper Saddle River, NJ, Prentice-Hall.
[6] Poppendieck, Mary and Tom Poppendieck, Lean Software Development: An Agile Toolkit. Addison Wesley Professional, 2003.
[7] Craig Larman, Victor R. Basili. Iterative and Incremental Development: A Brief History, 0018-9162/03/ © 2003 IEEE
[8] Coram, M., and Bohner, S., "The Impact of Agile Methods on Software Project Management", In Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS), April 2005, pp. 363-370.
[9] Jeffries, R., Anderson, A. and Hendrickson, C. (2001). Extreme Programming Installed. Upper Saddle River, NJ, Addison-Wesley.
[10] Weisert, C., (2002), The #1 Serious Flaw in Extreme Programming (XP), Information Disciplines, Inc., Chicago. [Online]. Available from: http://www.idinews.com/Xtreme1.html.

AUTHORS PROFILE

Mrs. Veerapaneni Esther Jyothi is a Microsoft certified professional, currently pursuing Ph.D (Computer Science and Engineering) from Rayalaseema University, Kurnool. She is working as an Asst. Professor in the department of Computer Applications, Velagapudi Siddhartha Engineering College and also has Industrial experience. She has published papers in reputed international conferences recently. Her areas of interest include Software Engineering, Object Oriented Analysis and Design and DOT NET.

Dr. K. Nageswara Rao is currently working as Professor and Head in the Department of Computer Science Engineering, Prasad V. Potluri Siddhartha Institute of Technology, Kanuru, Vijayawada-7. He has an excellent academic and research experience. He has contributed various research papers in the journals, conferences of International/national. His area of interest includes Artificial Intelligence, Software Engineering, Robotics, & Datamining.