

# Effective Implementation of Agile Software Development with a Framework, Metric Tool, and in Association with Cloud and Lean Kanban

V. Esther Jyothi<sup>1</sup>, K. Nageswara Rao<sup>2</sup>

<sup>1</sup>Ph.D Research Scholar, Department of CSE, Rayalaseema University, Kurnool, Andhra Pradesh, India

<sup>2</sup>Research Guide and Principal, PSCMR College of Engineering and Technology, Vijayawada, India

**Abstract**— *Delivering software in traditional ways is challenged by agile software development to provide a different approach for developing a product. Agile methods aim at fast, light and efficient than any other vigorous method to develop and support customers business without being chaotic. A Collaborative and Innovative Framework presented suggests the solutions for the levelheaded difficulties in Agile that leads to a software product that proves in practice to be of much higher quality. Also an object oriented software metric tool “Metric Analyzer” is proposed that detects and eliminates code smells in the code base by refactoring.*

**Keywords**— *Agile, Cloud Computing, Lean Kanban, Metrics, Refactoring.*

## I. INTRODUCTION

Agility is increasingly becoming the driving force in many organizations. Agile development - once a predominantly team-based practice, is grabbing the attention of the business. Software professionals are getting more knowledgeable about agile development and are now scaling it more broadly within their organizations. It's not just knowledgeable; the agile community is applying what they know about the methodologies more broadly in the workplace based on the success they have seen within single teams[1]. Agile momentum has taken off and its successes are being embraced at the enterprise level. Agile helps organizations complete projects faster and more people are recognizing that agile development is beneficial to business. It is proved that implementing agile on the whole delivers what organizations hope for.

My hypothesis is that the advantages of adopting an agile software development approach lies in the 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. This realistic, people oriented method to software development demands software practitioners. Several adopters experienced enrichments in competence, superiority, work inspiration, and consumer satisfaction. User involvement is given high priority in the working style of agile, drawing user's right in to the heart the development process [11]. 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.

This research paper is the result of an immense research work on the effective implementation of agile software development. It summarizes my research work for Doctor of Philosophy and is organized as follows: Section II explains the collaborative and innovative framework for agile software development; Section III presents the way of detecting the code smells in the code base of a java project by using our tool called “MetricAnalyzer”; Section IV focuses on the impact of cloud computing during various phases of agile software development that affect the efficiency of software delivery in time and on budget and explains how agile software development is accelerated using cloud; Section V discusses the approaches that can be considered while implementing iterative agile in coordination with lean kanban method and issues to produce defect free product.

## II. FRAMEWORK

A collaborative and innovative framework (Fig. 1) is a hybrid approach in which XP engineering practices are implemented in the scrum sprint. Sprints are iterative cycles where the functionality is developed or enhanced to produce new increments [13]. Each Sprint includes the traditional phases of software development: requirements, analysis, design, evolution and delivery phases[5].

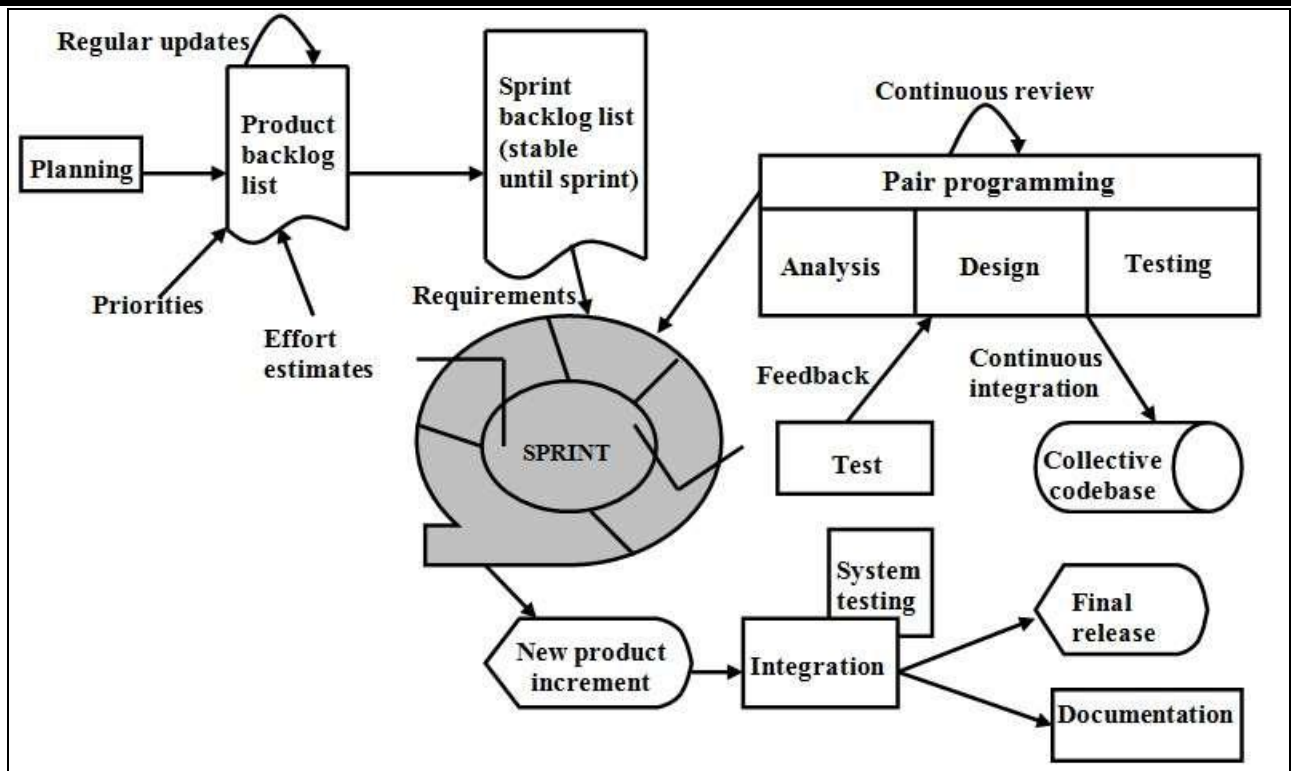


Fig. 1: Collaborative and Innovative Framework

These scrum phases are implemented using extreme programming methodology. The functional tests created by the customer are run at the end of each iteration.

Here the key characteristics of XP are included such as refactoring - restructuring the system by removing duplication, improving communication, simplifying and adding flexibility without changing its functionality, pair programming – two people write the code at one computer which is great for complex and critical logic, collective code ownership – code belongs to the project not to any individual engineer, continuous integration – a new piece of code is integrated into the code-base as soon as it is ready [9]. Thus the system is integrated and built many times a day.

All tests are run and passed for the changes in the code to be accepted. Extra testing and checking of the performance of the system before the system can be released to the customer. At this stage, new changes may still be found and the decision has to be made if they are included in the current release [3]. The postponed ideas and suggestions should be documented for later implementation. Communication and coordination between project members should be enabled at all times. Any resistance against XP practices from project members, management or customers may be enough to fail the process. Ultimately better results can be obtained by tailoring some of the scrum principles such as the daily scrum meeting to keep track of the progress of the scrum

team continuously and they also serve as planning meetings. Developing software on time and within budget is not good enough if the product developed is full of defects and customers today are demanding higher quality software than ever before [10] [12]. Now-a-days the software market is mature enough and users want to be assured of quality. Due to time-to-market pressures or cost considerations, the developer may limit the software quality assurance function and not choose to conduct independent reviews.

### III. METRIC ANALYZER

In Agile software development, refactoring helps to improve software quality. The proposed object oriented software metric tool “MetricAnalyzer” showed us great results after refactoring the code compared to the original code base of different projects. This tool was very much useful in eliminating code smells from the code and maintains threshold values of the considered object oriented metrics [6].

The following are few screenshots of the “MetricAnalyzer” tool for the chosen six metrics TNOA (Total Number of Attributes) (Fig.2), WMPC (Weighted Methods per Class), MIT (Max Inheritance Tree) (Fig.3), NOCP (Number of classes Per Package), CCIM (Cyclomatic Complexity) (Fig.4), SIZE (Total Lines of Code). The red highlighter shows that the value exceeded the metric threshold value.

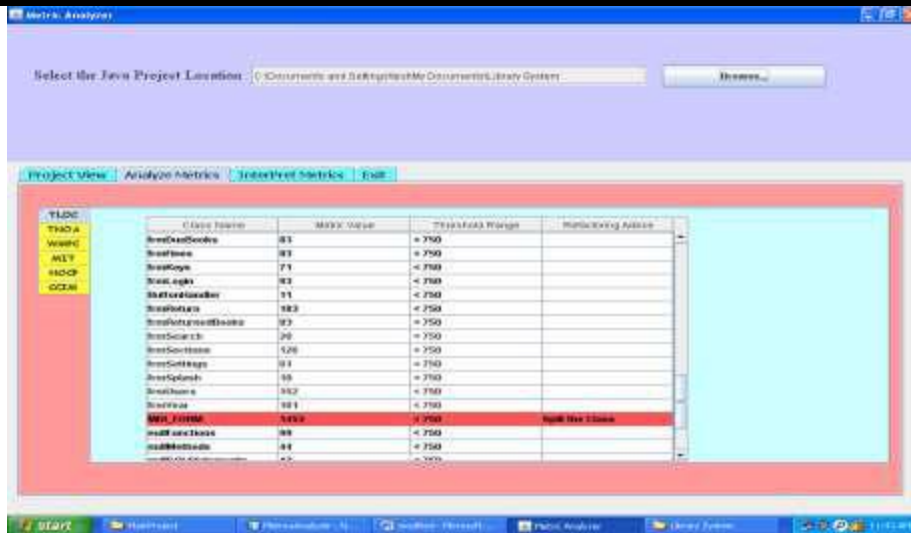


Fig. 2: Screen shot for SIZE metric in the “MetricAnalyzer” tool



Fig. 3: Screen shot for MIT metric in the “MetricAnalyzer” tool

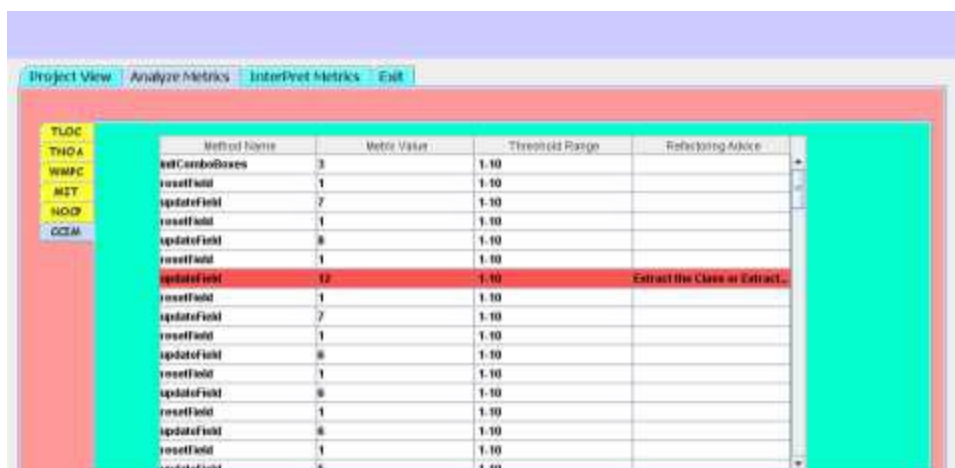


Fig. 4: Screen shot for CCIM metric in the “MetricAnalyzer” tool

Since object oriented technology exploits objects and not algorithms as its elemental building blocks, the approach to software metrics for object oriented programs must be different from the standard metrics set. Some metrics, such as lines of code and cyclomatic complexity, have become accepted as "standard" for traditional functional/ procedural programs, but for object-oriented, there are many proposed object oriented metrics in the literature. Object-oriented design and development is becoming very popular in today's software development environment. It entails not only a special approach to design and implementation; it requires different software metrics.

Software refactoring is a technique to enhance the maintainability of software, improve reusability and understandability of the software. "MetricAnalyzer" applies the Object Oriented metrics on the code base and these metric values are then interpreted. Then various refactoring techniques were used to improve the code design and along with that we also studied the impact of refactoring on the software quality through various metrics. The red highlighter showed the values which exceeded the metric threshold value and this made us to refactor those parts of the code base and improve the quality of software.

#### IV. AGILE AND CLOUD

Even though the release cycles are speedier with agile software development there can be delays with respect to the insufficient availability of underlying platforms. But when agile using the cloud can overcome the delays and greatly enhance the speed of continuous integration and release cycles because of the large number of resources availability in the cloud [8]. Effective implementation of agile practices is much more achievable with the collaboration of cloud.

Below are the benefits and opportunities to maximize the agility of the software development process and practices with the help of cloud computing:

**Reduces the lead time** – Agile development teams are limited to one physical server. But when cloud instances are used, practically an unlimited number of servers are available for development teams. This leads to the reduction of lead time since they need not to wait for physical server to become free to continue their work.

**Speed up release cycles** – Even though agile methodology is used for software development, there can be delays with respect to the availability of underlying platforms. Having a large number of resources available in the cloud, the agile development team can overcome

the delays and greatly enhance the speed of continuous integration and release cycles.

**Accelerates efficiency and effectiveness** – Even though the agile software development team performs several activities in parallel and also in serial, the parallel activities are delayed due to lack of sufficient services. Cloud computing can push it towards parallel activity by providing the required services which accelerates efficient and effective agile software development.

**Encourages innovation** – Cloud encourages the development teams to innovate and experiment new strategies because of the wide range of available services. We can provision platform as a service environments for all stages such as development, testing, staging and production. This will make sure; they have all the base services, such as databases, application servers, artifact deployers, service handlers and API managers setup in a consistent manner and ready to go. When the developer is done developing the application or service, the artifact can be moved to the next environment for testing, and all that is required is to point to the testing environment and test the application by the QA team (Mc Breen, 2003). This is development governance with cloud computing.

Project creation, repository creation, build plans, triggering builds upon commits and deploying artifacts onto the servers upon successful builds can all be automated with a cloud based integrated setup [2]. In other words, continuous integration just got simpler. By incorporating the cloud into the agile development process, customer feedback becomes easier to obtain, because the application is being tested in a cloud-based environment.

#### V. LEAN KANBAN

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 [4]. 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. For the effective work distribution among the agile and kanban teams to work in coordination, the workflow should follow the sorter analogy (Fig. 5) for process.

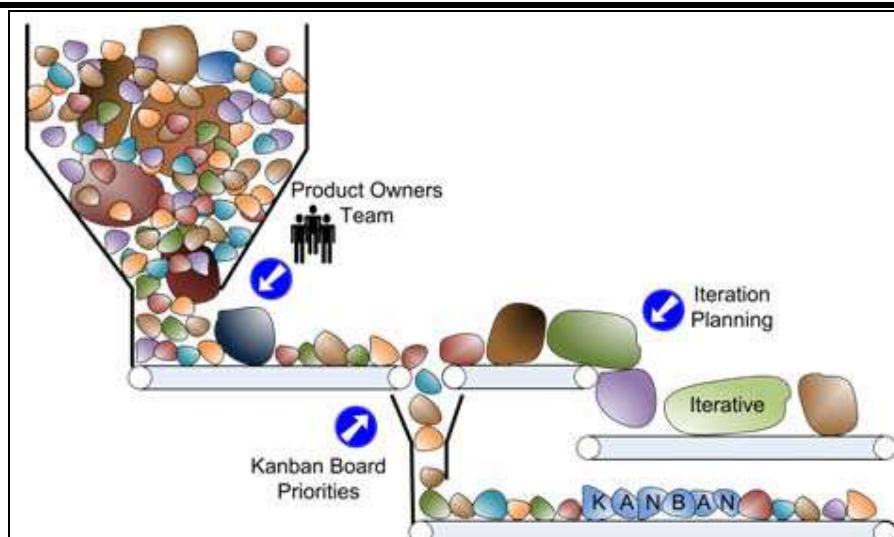


Fig. 5: The sorter analogy for the workflow process

Irrespective of the method of agility 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.
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.

Mc Breen defines agile quality assurance as the development of software that can respond to change, as the customer requires it to change. Beck defines agile software quality interims of efficiency, timeliness, cost effective, ease of use, maintainability, integrity,

robustness, extendibility and reusability. 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 [7].
4. Doing it right the first time keeps the maintenance cost low and learns how to make the most of the testing process.
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. Every role in the team should involve in testing activities to produce defect free product [14].

## VI CONCLUSION

Agile momentum continues strongly and widely across different organizations and the most widely used agile methodology is scrum undoubtedly. Keeping aside the misconceptions related to agile, organizations can succeed at the enterprise level. The proposed framework ensures transparency in communication and creates an environment of collective accountability and continuous progress. And our object oriented software metric tool



Metric Analyzer proposed has showed us great results after refactoring the code compared to the original code base of different projects.

#### ACKNOWLEDGEMENTS

The author Esther Jyothi Veerapaneni want to thank Dr. D. Punyaseshudu, Director, Research Studies, Rayalaseema University for his valuable suggestions during the research work and also specially thank the research guide Dr. K. Nageswara Rao for his continuous assessment and contribution throughout the research.

#### REFERENCES

- [1] Moran, Alan (2015). Managing Agile: Strategy, Implementation, Organisation and People. Springer Verlag. ISBN 978-3-319-16262-1.
- [2] Bowen, Janine Anthony (2011). Cloud Computing: Issues in Data Privacy/Security and Commercial Considerations. Computer and Internet Lawyer Trade Journal, 28 (8), 8.
- [3] Coram, M., and Bohner, S. (2005), 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.
- [4] Veerapaneni Esther Jyothi, K. Nageswara Rao (2012) Effective Implementation of Agile Practices, International Journal on Computer Science and Engineering, Vol. 4 No. 01 Page No. 87
- [5] Veerapaneni Esther Jyothi, K. Nageswara Rao (2010) Effective implementation of agile practices – A collaborative and innovative framework, CiiT International Journal of Software Engineering and Technology, September 2010
- [6] Veerapaneni Esther Jyothi, Kaitepalli Srikanth, K. Nageswara Rao (2012) Effective Implementation of Agile Practices – Object Oriented Metrics Tool To improve Software Quality, International Journal of Software Engineering & Applications, Vol.3, No.4, July 2012.
- [7] Salem, A. M. (2010), A Model for Enhancing Requirements Traceability and Analysis. International Journal of Advanced Computer Science and Applications - IJACSA, 1(5), 14-21.
- [8] Veerapaneni Esther Jyothi, K. Nageswara Rao (2014) Effective implementation of agile practices – In Collaboration with Cloud Computing, International Journal of Current Engineering and Technology, Vol.4, No.3 (June 2014).
- [9] "Extreme Programming Core Practices." Extreme Programming Core Practices. N.p., n.d. Web. 17 Nov. 2013. <<http://c2.com/cgi/wiki?ExtremeProgrammingCorePractices>>.
- [10] "SCRUM Guide." Scrum Guide. N.p., n.d. Web. 13 Nov. 2013. <[https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum\\_Guide.pdf](https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide.pdf)>.
- [11] McHugh, Martin; McCaffery, Fergal; Coady, Garret (2014-11-04). Mitasiunas, Antanas; Rout, Terry; O'Connor, Rory V.; et al., eds. "An Agile Implementation within a Medical Device Software Organisation". *Software Process Improvement and Capability Determination*. Communications in Computer and Information Science. Springer International Publishing. **477**: 190–201. Doi: 10.1007/978-3-319-13036-1\_17. ISBN 978-3-319-13035-4.
- [12] Deemer, Pete, et al. "The scrum primer." Scrum Primer is an in-depth introduction to the theory and practice of Scrum, albeit primarily from a software development perspective, available at: <http://assets.scrumtraininginstitute.com/downloads/1/scrumprimer121.pdf> 1285931497 (2010).
- [13] "Prezi Presentation: Why Scrum@ IT RM by Christian Popp." *YouTube*. YouTube, n.d. Web. 10 Nov. 2014.
- [14] <http://www.stickyminds.com/interview/agile-mindset-and-agile-trends-2015-interview-jeff-nielsen>

#### AUTHORS PROFILE



**Mrs. Veerapaneni Esther Jyothi** is a Microsoft Certified Professional and Solution Developer, currently doing research work in the area of software engineering. She is working as an Assistant Professor in the department of Computer Applications, Velagapudi Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India. She has published papers in various reputed international journals.



**Dr. K. Nageswara Rao** is a Principal of Potti Sriramulu Chalavadi Mallikharjuna Rao College of Engineering and Technology, Vijayawada, Andhra Pradesh, India. 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, and Data mining.