REVIEW ON SCRUM AND EXTREME PROGRAMMING FOR SOFTWARE QUALITY ASSURANCE IN INDUSTRIES

Aaditi Sinha*

Dr. D. R. Prajapati**

ABSTRACT

Agile software development encourages people collaboration through the project. Extreme programming is one of most popular and commonly used methodology in agile software development. Extreme Programming (XP) is a lightweight, predictable, efficient and flexible method. Scrum is also a commonly used methodology in agile it was initially developed by Ken Schwaber. As Agile methodologies are appreciated for integrated testing approach, quick response to change and being people centric, some claims also prompted, stating that these methodologies are not mature enough to be practiced on large scale. A humble effort has been made to address all related issues in this paper. This research has been done to understand how fundamentally the different methods Scrum and Extreme Programming, differ based on certain dimensions. This paper deals with the work done on the Scrum and Extreme Programming in the startup industries for the period 1999 to 2014.

* PG student, TQEM, Department of Production and Industrial Engineering, PEC University of Technology (formerly Punjab Engineering College), Chandigarh-160012 (India).

** Assistant Professor in the department of Mechanical Engineering, PEC University of Technology (formerly Punjab Engineering College), Chandigarh-160012 (India).

1. Introduction

Software Quality Assurance (SQA) stands on high level of importance in today's software industry. Sufficient research work is available in this field. SQA is interpreted in different ways and words. NASA Software Quality Assurance Center describes SQA, "Software Quality Assurance (SQA) is a planned and systematic approach to the evaluation of the quality of and adherence to software product standards, processes, and procedures".

Ultimate purpose of quality assurance is to attain better quality in software product. Different approaches and several quality models are followed in this discipline. SQA activities are practiced during project and these activities include process control, documentation, audits and verification and validation.In Agile Development, developers may also be responsible for QA activities. Agile development projects consist of short iterative development and release of product. And projects, following agile development, evolve around the developer and customer who are responsible to maintain product quality. If responsibility for quality, in agile development, is shifted on customer and developer, then the supporting role of QA must be identified. SQA is not only responsible for a particular project but also maintain the processes and culture of organization.User Experience Design (UXD) team approach is introduced collaborating with developers, this UXD team approach seems an attempt to redefine and replace the role of SQA in agile projects. Main focus of this thesis work is to highlight some gaps in agile SQA activities and to put forward suggestions for improvement. Quality management (QM) is essential and vital for all the types of organization. Quality management is a process of ensuring that required level of quality is achieved not only in products but also quality in the process through which these products are produced. It involves defining some appropriate quality standards and procedures to ensure that these are followed. The aim should be to develop a 'quality culture' where achieving quality is seen as everyone's responsibility. Quality Assurance, quality planning and quality control are the activities, which are involved in quality management. In Software engineering, conventional software development is referred as the methodologies that are process oriented in nature. Structure of software development process or software development life- cycle is followed to develop a software product, strictly in conventional development.

As software industry is growing, software products are getting more complex and users' demands are increasing. These factors are, ultimately, increasing the complexity of software development projects. To tackle mounting complexities several models and standards are being followed. Some of the most practiced models are:

- International Standards Organization (ISO 9000)
- International Standards Organization (ISO 9126)
- Capability Maturity Model (CMM)
- Capability Maturity Model Integration (CMMI)
- Software Process Improvement Capability Determination (SPICE)
- Six Sigma

Besides targeting quality of product/service, these models also focus on quality of software development process on organization level. But eventually, the responsibilities of QA are being increased. QA activities have gained the importance of backbone in an organization. These activities are responsible for development process; quality of product and these activities keep the project on track. On managerial level, standards and procedures are established for software development. The role of SQA is to assure that defined standards are documented properly, procedures are followed. For this purpose, product evaluation is conducted, audits and meetings take place to monitor and evaluate that processes are following defined procedures. Solely, on product development side, verification and validation is core Software Quality Assurance (SQA) responsibility and activity to maintain and evaluate product quality. Software Quality Assurance (SQA) activities have gained dramatic importance. To imagine a successful project without the involvement of SQA is almost impossible. Efficient quality assurance is the key to successful project. Besides, all these facts SQA activities are becoming more and more complex. Software Quality Assurance (SQA) personnel conduct surveys, evaluation, meetings and internal audits. They are considered as experts within an organization as they educate and implement SPI in it. Besides, merging QA activities in software development, agile methodologies cut short the organizational role of QA. Developers may be aware of testing and designing but they might be less aware of SPI on organizational level. Literature shows that there is need to redefine the role of SQA in agile development projects in order to increase organization knowledge and maturity for maximum output.

Figure 1 represents a single iteration of agile development approach. It is observed that all required phases of software development (Plan, Design, Development, Test and Analysis) takes place in one iteration. After the completion of one, iteration, the software product is sent for users'/customers' review. The next iteration starts to add required functionality according to feedback of users'/customers' review.

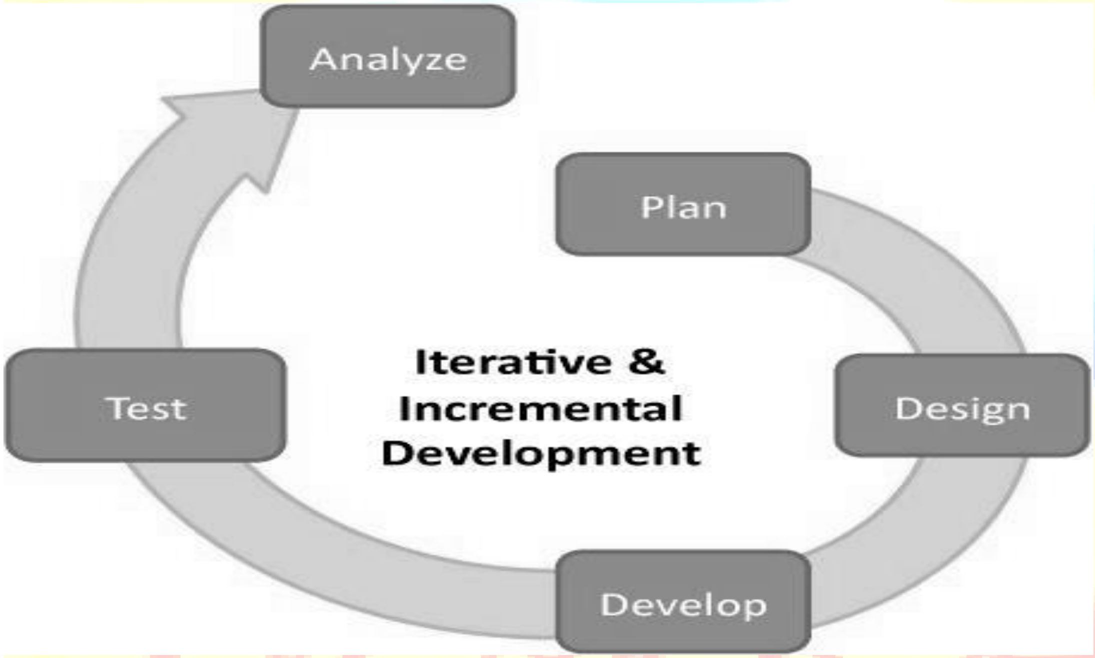

Figure 1 Single Iteration of Agile Development Process

Many Agile methodologies been introduced but commonly used agile methods are:

i. Extreme Programming (XP)
ii. Scrum
iii. Crystal

Agile methodologies are known as lightweight methodologies as they are not process intensive. Agile methodologies have also changed the way of SQA activities. Documentation is not very heavy and it emphasizes customer/user requirements.

Extreme programming is one of the most popular and commonly used methodologies in agile software development. The founder of this methodology is Beck (2001), as he introduced and defines a number of principles and practices to maintain the productivity of development team and to raise the accuracy and quality of produced system. Extreme Programming (XP) is a lightweight, predictable, efficient and flexible method. It was developed to fulfill the need of small team who are dealing with the imprecise and changing requirements to develop the software in a better way. XP contains a set of discipline and practices for software development process. To apply the XP methodology there are some practices that are needed to be followed in the development process.

Scrum is also a commonly used methodology in agile and it was initially developed by Ken Schwaber (2001). The term ‘Scrum’ is derived from a strategy in the game of rugby, where it denotes “getting an out-of-play ball back into the game” with teamwork. Scrum provides project management with framework that includes development tasks like requirement gathering, design and programming are taking place. It does not provide any specific method to be applied; it guides the management how their team should function to maintain the flexibility of the system, in applying the environmental changes.As agile methodologies are appreciated for integrated testing approach, quick response to change and being people centric, some claims also prompted, stating that these methodologies are not mature enough to be practiced on large scale.

2. Review of literature

Various researchers proposed many models of commonly used methodologies like Scrum and Extreme Programming and presented in this section.

Beck (1999) stated that Extreme programming is one of most popular and commonly used methodology in agile software development. Extreme Programming (XP) is a lightweight, predictable, efficient and flexible method. Schwaber (2001) stated that Scrum is also a commonly used methodology in agile it was initially developed by Ken Schwaber. The term

‘Scrum’ is derived from a strategy in the game of rugby where it denotes “getting an out-of-play ball back into the game” with teamwork.

Williams and Cockburn (2003) stated that agile methodologies are developed to ‘‘embrace, rather than reject, higher rates of change”. Highsmith (2004), Ambler(2005) and Turk (2008) postulated that although, agile contains key benefits to produce better results but to adopt a new process, is also a challenging task, to accept that change organizations have to rearrange their prior setup, practitioners and opponents of agile also indicate challenges in agile adoption. Ericksson et al. (2005) defined agility as “agility means to strip away as much of the heaviness, commonly associated with the traditional software-development methodologies, as possible to promote quick response to changing environments, changes in user requirements, accelerated project deadlines and the like”. Salo and Stillp (2005) stated that whenever a process is changed or replaced, it is directly depends upon the behavior and response of that process. In Software development, it is a complex task to reduce the complexity of development a number of processes that are being followed.

Highsmith and Highsmith (2006) described that Scrum contains both managerial and also development processes. Scrum involves Rapid prototyping in this practice team simply take the overview of system requirement form customer. These requirements are not only incomplete but can be changed in the development process. The main practice in scrum is daily 15 minutes meeting to coordinate and integrate the development issues. Talby et al. (2006) mentioned that in agile software development testing and development of software runs parallel in order to achieve higher level of quality. Testing can be valued as the backbone of QA activities and a vital step to attain quality in software product. Agile Manifesto (2007) is the guide for Agilemethodology, which clearly states the 12 principles of Agile Software Development. These principles form the basis of all the processes that follow Agile Methodology

Stamelos (2007) stated that quality is fitness for use, which means the following two things: “(1) quality consists of those product features that meet the needs of the customers and thereby provide product satisfaction. (2) Quality consists of freedom from deficiencies”. Agile process has the ability to maintain and achieve quality of product due to its continuous focus on customer demands and process improvement in respect to active the defined functionality developing system. Sone (2008) stated that it is very common nature of adaptability, when we find something better than existing work and routine we are attracted to adopt it or replace with our existing practices. Rijayasarthi and Turk (2008) stated that organizations believe and experience that after applying agile characteristics, they develop better quality products and attain higher level of error absence. To maintain the software quality and accuracyisa main issue in organizations development goals.

Oxford Dictionary (2009) defines Agile as "the quality of being agile; readiness for motion; nimbleness, activity, dexterity in motion. Agility, with regard to software development, can be expressed as the flexible, ready to change and quick-responsive nature of software development process. Jeffries (2009) observed that in XP every contributor of the project has his integral part in the team. XP teams are formedaround a business representative called "the Customer". With focused on business value, XP teams use simple planning and tracking to decide and predict, what should be done next and when project will be finished. The team produces software in small releases that process the entire test defined by customer.

Weinstein (2010) stated that Extreme Programming (XP)'s minimalism can lead to confusion, ambiguity, and forgotten requirements and design idioms. Extreme Programming (XP)'s answer is to foster close communication by putting all the programmers in the same room, but this may not be sufficient except in the most trivial projects. Putting all your programmers in one room doesn't guarantee they'll communicate effectively Adams (2010) emphasized that before shifting to Agile, the members of his team wanted to focus on interactions with one another, customers and process. Prior to this change, we spent time focused on the reverse—process, customers and then interactions. The result shifted their focus too empowering the team, restoring the focus on customers and executing on the right solutions at the right time. Levinson (2010) explained that tools like Team Foundation Server offers new tools to help you easily visualize the backlog, workload per iteration and workload per developer. Teams can quickly and easily break down the work and make adjustments as needed and get feedback immediately. They can easily manipulate the data and TFS to store and report on the data, helping teams be faster and more accurate in their planning stages and helps them better keep track of their progress during iterations. The utility is very high and can increase productivity of teams that practice agile methods for making their products. Versionone (2011) released the results of their State of Agile Development Survey for 2011 that showed Scrum remained the most popular agile method to be used by more than half of the respondents. It stated the barriers and concerns with various methods of agile methodology reflect what we all understand: that change is hard. And agile development is change. As I interpret the survey, the actual benefits obtained are consistent with what you would expect during an agile adoption. These are the quicker, easier wins. Improved team morale was the fourth benefit obtained from implementing agile, another benefit that should result from an agile adoption.

Standish Group (2012) observed that agile process is the universal remedy for software development project failure. Software applications developed through the agile process have three times the success rate of the traditional waterfall method and a much lower percentage of time and cost overruns. Gregory (2012) stated the integrating user experience design into agile working is a challenge faced by many agile teams. Although the work of user experience designers and developers complements each other, the different goals, processes and working practices of developers and user experience designers pose challenges in practice. The Agile Research Network have documented working practices adopted and lessons learned from the experiences of one company integrating user experience designers into their Agile process.

Caseau (2013) emphasized that the topic of mapping the similarities and differences between lean development and agile is quite interesting because the influences and relationships make for an intricate pattern. Agile methods and its various subsequent developments have been implementing “lean principles” for many years. Lean Software Development, on the other hand, inherits from the knowledge built by agile communities. Khan (2014) observed that XP has a lot to offer for the welfare of each person, and for the collective welfare too. Having a shared set of values and principles is essential in a team, because it is used to help the team to eliminate options and make decisions. Although, value based decisions always seem like choosing the harder option, but the payback is often greater. Angin (2014) observed that agile software development approach makes developing secure software challenging. Existing approaches for extending the agile development process, which enables incremental and iterative software development, fall short of providing a method for efficiently ensuring the security of the software increments produced at the end of each iteration.

3. CONCLUSIONS

Agile methodology is one of the most practiced software development methods. Since, Agile has numerous different types of methods that are based on the 12 Principles stated in the agile manifesto; it is hard to differentiate between all of them. The most widely used methods of agile software development are SCRUM and Extreme Programming (XP). In order to understand their functionality and benefits, it is crucial to know the extent of their difference and similarities. The key features on which the comparative study for SCRUM and Extreme Programming (XP) is based are scheduling pattern, Team Management, Process Management, Customer Satisfaction, Quality Assessment and Overall pattern.

By the means of this study, it was observed that for scheduling and team management, there is no significant difference between SCRUM and Extreme Programming (XP) methods of agile software development while in case of Process Management, Customer Satisfaction, Quality Assessment and Overall pattern there is significant difference. Since, there exists a significant difference between some parameters, it can be postulated that there might also exist a correlation between the two methods based on those parameter.

REFERENCES

- Agile Manifesto, 2007, *Manifesto for Agile Software Development* (Website: http://www.agilemanifesto.org)
- Ambler, S. (2004), “Remaining Agile*”*, *Amby Soft White Paper Industry Guide*, Vol. 3, pp. 45-46.
- Ambler, S., 2005, “Quality in an Agile World”, *American Society for Quality Journal*, Issue 6, pp. 21-28.
- Angin A. (2014), “Agile Software Development: Evaluating the Methods for Your Organization, *Artech House Incorporated Booklet*, Vol. 1, pp.134-140.
- Beck, K. (1999), Embracing change with Extreme Programming, *IEEE Journal for Computational Research*, Vol. 32, Issue 10.

- Caseau, I. (2012), “Biology of Agile Distributed Information system, *IEEE Journal of Computational Research*, Vol. 32, pp. 57-62.
- Dove R. and Turk G. ,2008 ,Relating Agile Development to Agile Operations, Conference on Systems Engineering Research ,Vol. 3, pp.28 – 33.
- Erickson J., Lyytinen K. and Siau K. (2005), Agile Modeling, Agile software development, and extreme programming: the state of research, *Journal of Database Management*, Issue 16.
- Gregory P., 2012, User Design challenges for Agile Development, Available at: http://agileresearchnetwork.org/author/agileres/
- Highsmith A. and Highsmith James A. (2006), Agile Software Development Ecosystems, *Agile Manual by Addison-Wesley*, Vol.7, No.2, pp.5.
- Highsmith J. (2004), Objections to Agile Development, Cutter Consortium *White Paper Proceedings*, Vol. 32, pp.42-45.
- Jeffries R. (2009), What is Extreme programming, Available at : http://www.xprogramming.com/xpmag/whatisxp.htm
- Khan A. (2014), Extreme Programming is as relevant now as before , Available at : http://f3yourmind.net/2014/01/06/the-relevance-of-xp/
- Levinson J. (2010), “Agile Project Management for Everybody”, Available at: http://visualstudiomagazine.com/articles/2010/03/30/agile-project.
- Oxford English Dictionary (2009), website: http://www.dictionary.oed.com
- Rijayasarathy, L. and Turk, D. (2008), Agile Software Development: A Survey of Early Adopters”, *Journal of Information Technology and Management*, Volume 19, No. 2.
- Salo, O., Ronkainen, J. and Stilip, J. (2002), Agile Software Development Methods: Review and Analysis, *Journal for Software Engineering and Development*, Vol. 2, Issue 3, pp. 6 – 12.
- Schwaber K. and Beedle M. (2001), Agile Software Development with Scrum, 1st Edition, *Prentice Hall of India*, Vol. 1 Issue 1.

- Sone S. P. (2008), Mapping agile project management practices to project management challenges for software development, IEEE Software Engineering Methods, Vol.1, Issue 3, pg. 191.

- Stamelos, A. C. (2010) , Make Agile Work for You in TFS (website: http://msdn.microsoft.com/en-us/magazine/hh205635.aspx

- Standish Group (2012), The Standish Group Report CHAOS, Available at: http://www.projectsmart.co.uk/docs/chaos-report.pdf

- Talby, D. and Keren, A., Orit, H. and Yael, D. (2006), Agile Software Testing in a Large-Scale Project, *IEEE Software Engineering Methods*, Vol.2, Issue.3, pp.36.

- Version One (2011), State of Agile Survey Results Show Agile Adoption Stable, Website: http://www.infoq.com/news/2012/02/2011_state_agile

- Williams L., Cockburn A. (2003), Agile Software development: it's about feedback and change, *IEEE Computational Research Articles*, Vol. 11.

- Weinstein A. (2010), Agile development with ICONIX process: people, process, and pragmatism, *Proceedings for Conference on Agile Development*, Vol.1, Issue 2, pp.42-48.
