

XCRUMBAN: A Proposed Agile Methodology

Nikhath Ikram¹ and Dr. Harsh Dev²

Assistant Professor, Computer Science & Engineering, Pranveer Singh Institute of Technology, Kanpur, India¹

Professor and Dean Research, Pranveer Singh Institute of Technology, Kanpur, India²

nk24786@gmail.com

Abstract: *With the recent advancement of technology in this era, software industries has grown outrageously. Software industry has shown such great hike in technology which is non-comparable to any other industries. Various methods have been established which improves the software quality one such method is Agile. Agile software development has gained a lot of attention because of its simplicity and ease of use. Agile software development is an approach which produces quality software with remarkable team interaction and more of customer involvement. Agile method is basically ideally suited for a scenario where requirements are changing in continuous manner. One of the most important advantage of using Agile is, it takes less time for software release, easy to understand and require less documentation. This paper deals with various agile methods, their comparison, advantages, shortcomings and XCRUMBAN, a new proposed framework to overcome those mentioned shortcomings.*

Keywords: XCRUMBAN, SRCUM, Extreme Programming, Feature Driven Development, Crystal, Adaptive Software development, Dynamic System Development Method.

I. INTRODUCTION

Agile means something that is able to move easily and quickly. Agile software development is a framework which is conceptually designed to develop iterations throughout a project's life. There is continuous feedback on output which is evaluated at the end of each iteration. Requirements are broken and categorized into tasks which are small in size, these requirements are developed in an iterative manner over shorter period of time. There is a time-box technique in which for an activity to be performed there is a fixed time slot.

In Agile process system variables are first defined, then the architecture is designed and requirements are specified, module is build according to the specified requirement and whether it fulfil the user requirements or not, integrate all the module to form a system and test the system as a whole, if the system is complete, deliver it else requirements are again specified and process continue till the system is delivered.[2] Agile Manifestoes is the base foundation of building quality software in agile development which comprises of strong team players, less documentation, enthusiastic customer interaction and adapting continuous changing requirements. With such base foundation agile development is able to adapt and respond positively to the changing requirement and maintain the quality of the software.

Agile method has continuous integration, verification and validation in terms of software development, demo products are made and delivered in continuous manner, thus maintaining the likelihood of the customer in the final product. Few valuable aspect of agile which makes it an exceptionally good is it focus on customer value, is self-organized, emphasizes on improvement which is done in continuous manner and delivery is done iteratively in short intervals. In this paper Agile methodologies are discussed in detail along with their pros and cons, later in this paper a new technique(XCRUMBAN)is proposed which is the hybrid of all Agile Methodologies to overcome the cons of individual method and which can be used in a case study.[1]

II. LITERATURE SURVEY

Agile has various methodologies which can be classified as given in Figure 1 below:

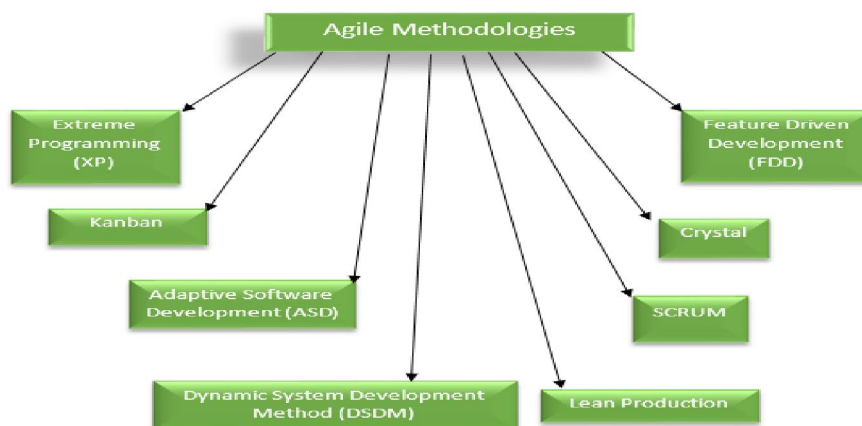


Figure 1: Agile Methodologies

Table given below shows the proper description along with pros and cons in each agile methodology. By studying this we can get clear idea about how they access, their advantages as well as their limitations.

Agile Method	Description	Pros	Shortcomings
Extreme Programming (XP) ^[7]	XP is considered useful where there is continuously change in requirements. XP improves the productivity of the system by frequent release of the product. There is a checkpoint in the system where the change in requirements can be implemented easily. XP focuses on customers rather than on process.	Simplicity Time and cost efficient method Constant feedback Fast Open Communication with the team members	Quality is not maintained Focus on code not on architecture Lots of work-pressure with hard deadlines Lack of Documentation
SCRUM	It is one of the agile methodologies which focuses on task management in team oriented development organization. It is inspired from rugby game. It focuses on team development and usually have small team comprises of 6-8 members. SCRUM comprises of Scrum Master (responsible for assembling the team, handling meetings), Product Owner (person who manages product backlog and is responsible for delivery of modules after each sprint) and Scrum Team (group of people who manages work so as to finish it within a sprint).	Create system transparency Continuous feedback from stakeholders and customers Short sprints helps to incorporate the changes sooner Effective use of time and money	Least bothered about deadline Need experienced team members High chances of project failure Quality maintenance is tough Daily meetings leads to frustration in team member
Kanban ^[14]	Worked in just-in-time concept. It develop product in one large development cycle. It's is not iterative, instead it is incremental. Developers focus on small amount of work at a time.	Limitation on number of running task is possible Continuous deliveries Commitment and prioritization is	Time related issues Team size not defined Misinterpretation in communication

		optional Can add functionality whenever there is an availability in terms of capacity	
Crystal	Crystal indicates the risk in human life. Basic aspects on which Crystal depend is people, interaction, community, communication, skills and talents. It is based upon three concepts: Chartering (teams are developed, initial planning is done), cyclic Delivery (there are more than one delivery cycle which works individually) and wrap-Up (mostly deployment and post deployment reviews)	Frequent Deliveries Closer communication Needs proper technical organization Helps to make perfect product	Project is not so direct Not ideal for scattered team No product dependency on customer requirements
Adaptive Software Development(ASD)	Here independent agents collaborate to develop a solution for a problem which is beyond an individual to work upon. It focuses on time-boxing, collaboration of teams, is component based and perform mission driven planning.	Develop better products Early delivery of product Good transparency between customers and developers	User oriented environment Product cost is high due to continuous testing Time taking development process
Dynamic System Development Method(DSD M) ^[9]	It is a rapid application development, provides framework for Agile development. User involvement is major and they are given power of decision making. Focuses on frequent product delivery. It has 7 phases: Pre-project, Feasibility study, Business study, Functional model iteration, Design and build iteration, Implementation and Post-project.	Easy to use Time efficient Complex functions are added in a continuous manner More of user involvement	Way too expensive Not ideal for smaller low budget processes Not many experts are available because it is a new method
Feature Driven Development (FDD) ^[12]	FDD focuses on design and build. There are specific short phases that are handled individually. FDD comprises of domain object modeling, product development by feature, inspections, management of configurations, feature teams, progress and result evaluation.	Easy maintainable Ideal for larger projects Less time Work is done in parallel Proper tracking of progress Recognition of best practices for product development	Not ideal for smaller projects No written documentation Not well defined iterations Main developer has great dependency of product
Lean Production	Lean software development is a set of principles that can be applied to software development to decrease programming effort, budgeting, and defect rates by one third. The	The elimination of waste, thus increasing overall efficiency of the	Project dependent upon the cohesiveness of the team

	principles were adapted from lean manufacturing. This approach is beneficial to an organization because agile iterations eliminate extensive pre-planned specifications. User stories rather than large upfront specs are easily understood by each team member and simpler to communicate.	development process More functionalities can be added to the product Creates more motivated teams	Customers must be aware of what they want beforehand There must a person with right business skills
--	--	---	--

Table: Overview of Agile Methodologies

III. XCRUMBAN: THE PROPOSED FRAMEWORK

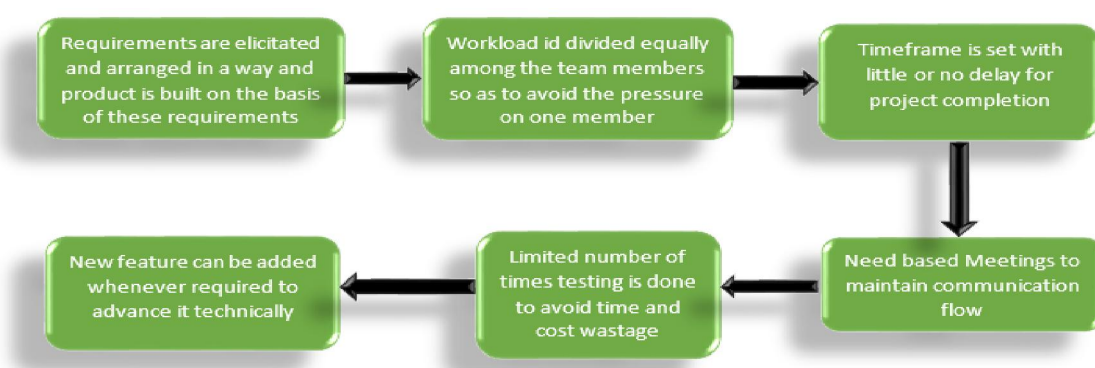


Figure 2: XCRUMBAN

After studying various agile methodologies, we can clearly see the positive points and shortcomings of each of them individually. The main motive was to develop a framework which could overcome the shortcoming of all these methods and which could lead to easy and fast completion of project.

XCRUMBAN is direct and requirements are arranged systematically and the product is built on the basis of these requirements (unlike Crystal), workload is divided proportionately among the team members so as to avoid the piling up of load on one team member (unlike Feature driven development). Time related issues is minimized by setting a timeframe to complete a project (unlike Kanban). There is a set deadline with little or no delay allowed to overcome delays in every stage of product development (as in case of Extreme Programming method). Additionally the team size is clearly defined and need based meetings are held with structured motive to resolve the problems in development, this avoids any sort of misinterpretation among the team members and thus communication flow is smooth (unlike Kanban). There is fixed time for meetings so the team members don't get frustrated (unlike SCRUM). Also testing here is not done in enormous number, there is fixed number of testing to be done on the project, thus reducing the time wastage and reducing the overall cost of the product (unlike Adaptive Software Development). New features are added whenever there is a need to add so as to make the project advance (unlike Dynamic System Development Method) and enhance the quality of the product by continuously working on its improvement (unlike Extreme Programming and SCRUM).

IV. RESULTS/DISCUSSION

In our college premises, we have taken various group of students with same project some working with our proposed framework and some groups working with other frameworks. We have clearly seen the difference in the result of the groups, a group that worked on our framework showed better result in terms of completion time and quality of the product. Following factors that have actually affected the completion time and quality of the project is as listed below:

- Communication among the team members, with the help/availability of need based meetings.
- Testing tool.

- Instead of having fixed deadlines for each stage, we can have deadline with some delay so that the time saved in one stage can be utilized in another if needed.

V. CONCLUSION AND FUTURE WORK

This framework can be further recommended to industries as it is tried and tested in various projects in our college premises and has given tremendously successful results. Furthermore, for getting even better results in future, depending upon the type of the software development problem, we conclude more stress to be given for a particular stage.

REFERENCES

- [1]. D. Cohen, M. Lindvall, and P. Costa, *Advances in Software Engineering*, chapter An Introduction to Agile Methods. Elsevier, Amsterdam, 2004.
- [2]. Amit Sharma and Rajesh K. Bawa, A Multilevel Hybrid Approach for Selection of Agile Development Method Using AHP, PROMETHEE and Fuzzy Logic, *STRUCTURAL INTEGRITY AND LIFE* Vol. 17, No 1 (2017), pp. 49–54 March 2017.
- [3]. Sharma, A., Sharma, R., A systematic review of agile software development methodologies, *Proc. of the Nat. Conf. on Innov. and Develop. in Engineering and Management*, 2015.
- [4]. Keramati, H., Hosseinabadi, S.H.M., Integrating software development security activities with Agile methodologies. 6th IEEE/ACS Int. Conf. on Comp. Systems and Appl., AICCSA 2008, Doha, Qatar, pp. 749-754
- [5]. Sandhya Tarwani and Anuradha Chug, *Agile Methodologies in Software Maintenance: A Systematic review*, *Informatica* 40 (2016) 415–426, November 2016.
- [6]. Sonia Thakur and Amandeep Kaur, Role of Agile Methodology in Software Development, *International Journal of Computer Science and Mobile Computing*, Vol. 2, Issue. 10, October 2013.
- [7]. Giulio Concas, Ernesto Damiani, Marco Scotto and Giancarlo Succi, *Agile Processes in Software Engineering and Extreme Programming*, 8th International Conference, XP 2007 Como, Italy, June 2007.
- [8]. Orit Hazzan and Yael Dubinsky, *Agile Software Engineering*, Undergraduate Topics in Computer Science ISSN 1863-7310, 2008.
- [9]. A.H. Mohammad et al., "Agile Software Methodologies: Strength and Weakness," *Int. J. of Engineering Science and Technology*, Vol. 5, No. 03, March 2003, pp. 455-459.
- [10]. L. Williams, "A Survey of Agile Development Methodologies", ©Laurie 2007.
- [11]. M.R.J. Qureshi, "Agile software development methodology for medium and large projects," *IET Software*, Vol. 6, No. 4, ©2012 The Institution of engineering and technology, pp.358-363.
- [12]. F.K.Y. Chan and J.Y.L. Thong, "Acceptance of Agile Methodologies: A critical Review and conceptual framework," *J. Decision Support Systems*, Vol. 46, No. 4, 2009, pp. 803-814.
- [13]. L.R. Vijayarathay and D. Turk, "Agile Software Development: A survey of early adopter," *Journal of Information Technology Management*, Vol. 11, no. 2, 2008, pp. 1-8.
- [14]. S. Ilieva et al., "Analyses of an agile methodology implementation," in *Proc. 30th EUROMICRO Conference*, ©IEEE, 2004, pp. 326-333.