Agile Software Development: An Alternative Approach to Software Development

Garima Verma

DIT University, Dehradun, Uttrakhand

Email-id: garimaverma76@gmail.com

Abstract- Traditionally, software development has been managed through plan based approaches that are full of limitations such as poor quality, not meeting customer requirements, unrealistic project development durations etc. To overcome all these problems, Agile software Development approach is being tattooed as an alternative approach. In this approach, software projects are developed using Iterative and Incremental methods using cross-functional teams and self-organizing. The main focus, in these methods, is to determine requirements and solutions through collaboration between these teams through adaptive planning, evolutionary development and delivery and time-boxed iteration. There is a lot of flexibility and speed lent to development process in this approach to meet customer requirements better and in time.

This paper is an attempt to understand Agile Software Development approach in detail to critically examine and compare its utility as an alternative approach to the development of software projects.

Key Words– Agile, time-boxed, adaptive planning, evolutionary development

I. INTRODUCTION

Software development has gone through significant changes in the recent times. Earlier it was traditional waterfall approach and now it is Agile Software Development approach. As a traditional approach, waterfall approach has been more in use in software development. One of the key reasons for this is even complex software systems could be built in a sequential, phase-wise manner. In this, a detailed requirements analysis is done at the beginning by a team of Requirement Analysts, which then is sent for implementation of master design for further development of software. It somewhat works like a conveyer belt in a production line. Requirement Analysts, Software designers and Software Developers work in tandem to develop quality software. The system specifications are collected by a team of requirement analysts, and then sent to the software designers for creating diagram documents that contain the codes which finally is sent to a team of software developers. All

this is done using GANTT or PERT charts which clearly shows the dependencies for each member of the developmental team. Despite this, only 9% to 16% of the projects have been found to be developed on-time and on-budget. Moreover, the problem of meeting changing customer requirements fully has been another critical issue that cannot be dealt with satisfaction.

Due to the problems inherent to the traditional approach, many improvements were suggested by experienced practitioners and labeled their methods as "Agile Software Development". It has been found that Agile Software Development approach has several advantages in today's scenario when customers have become highly demanding in terms of quality and completion time. The key differentiating merit of this approach is speed and flexibility. Since we use incremental method here, therefore, it very easy to incorporate new changes that a customer wants. This, in the long run also reduces the overall cost of producing the software project.

In this paper, we have made an attempt to critically examine and compare two approaches of software development, one is the traditional waterfall approach and other is "Agile Software Development" approach.

II. LITERATURE REVIEW

Dingsoyr 2008 characterizes the status and main challenges for research on agile software development, and propose a preliminary roadmap, focusing on providing more empirical research, primarily on experienced agile teams and organizations, connecting better to existing streams of research in more established fields, giving more attention to management-oriented approaches, and finally give more emphasis to the core ideas in agile software development in order to increase our understanding.

Malik 2008 [1] said that new software development methodologies were introduced to fit the new culture of the software development companies. Agile methodologies were introduced to meet the new requirements of the software development companies.

Frauke, Armin 2003 made a comparative analysis between both approaches in term of requirement engineering methods.
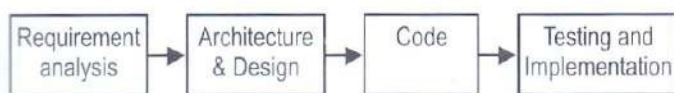
III. PURPOSE OF THE STUDY

The purpose of this study to analyze two software development approaches i.e. traditional and agile software development. In this study, we shall compare these two approaches and bring out their benefits and limitations. In addition, we wish to find out the efficacy of agile software development approach and discuss how it can be improved in future.

IV. TRADITIONAL DEVELOPMENT METHOD

Generally when we talk about the traditional way of software development then we think about an approach which is sequential as well as phase wise. It starts from problem identification, analyzing, designing, and then documenting and validating requirements for the system to be developed. The requirement is a phase in which developer try to gather different kind of requirements from the customers in the beginning of the project. After that they go for analysis phase in which all the requirements are analyzed and represented using different methods so that they can be assured that whatever requirements they have collected are correct and accurate. After analysis they prepare a document called as SRS it acts like a contract between customer and developer so that there should not be any problem related with requirements can occur. After analysis developer go for design phase in which they design all the documents like flowchart, pseudo code, algorithms etc for the problem. After that programmer does coding for the project and then testing and maintenance.

The most popular method of traditional approach is water fall model, after this model there are many more model like V-Model, Spiral Model, etc.

Phase of Traditional Method

V. AGILE DEVELOPMENT METHODS

Agile tends the quality of being agile; readiness, motion, active etc. The agile development is less documents centric and more code oriented. Methods for Agile software development constitute a set of practices for software development that have been created by experienced practitioners.

There are various Agile methods used for software development. In this paper we are describing the most common agile methods.

A. Extreme Programming- This is based on values of simplicity, communication, feedback and courage. (Beck 1999) XP has evolved from the problems caused by the long development cycles of traditional development models. It works by bringing the whole team together in the presence of simple practices, with enough feedback to enable the team to see where they are.

B. Scrum – Scrum is a method for managing the system development process by applying ideas on flexibility, adaptability and productivity from industrial process control theory (Schewaber 1995, schewaber and Beedle 2002). Scrum focuses on how a team should work together to produce quality work in a changing environment.

C. Crystal family of methodologies- The crystal family of methodologies includes a number of different methodologies for selecting the most suitable methodology for each individual project (Cockburn 2002). The different members of the family can be tailored to fit varying circumstances. Each member of the crystal family is marked with a color indicating the "heaviness" of the methodology, i.e. the darker the color the heavier the methodology : clear, yellow, orange, red etc.

D. Feature driven development – FDD is a short iteration process for software development focusing on the design and building phase instead of covering the entire software development process, but rather focuses on the design and building phases (palmer and Felsing 2002). However, it has been designed to work with the other activities of a software development project and does not require any specific process modes to be used.

E. Dynamic Systems Development Method- It provides a framework for rapid application development. The fundamental idea behind DSDM is that instead of fixing the amount of functionality in a product, and then adjusting time and resources to reach that functionality (Stapleton 1997). It is preferred to fix time and resources, and then adjust the amount of functionality accordingly.

VI. TRADITIONAL VS. AGILE SOFTWARE DEVELOPMENT APPROACH

Although traditional method is very systematic and step wise but one of the biggest drawback of this method is that it requires gathering of all the requirements of the project at the beginning, design is completed next, and finally the master design is implemented into production of quality software. This approach holds that complex system can be built in a single pass, without going back and revisiting requirements or design ideas in light of changing business or technology conditions. But actually this is not possible that any customer can provide requirements accurately in the beginning of the project. And if the requirement gets changed in between the developmental phase or cycle then either it is impossible to accommodate these changes or will cost a lot more to customer as well as developer. This problem is very well taken care of in Agile Software Developmental approach. Actually, requirement analysis, in this approach, is done at every stage i.e. requirement, design, testing and implementation stage. Because of this, customer's changing requirements can be incorporated at any point of time. This saves a lot of costs for the company and results in quality software developments also.

Traditional development approach is document centric approach and for every small change we need to prepare a new document. Agile methods, on the other hands, are people-oriented than process-oriented. These rely on people's expertise, competency and direct collaboration rather than rigorous, document centric processes to produce high-quality software.

Further, Agile methods are adaptive rather than predictive. With traditional methods, most of the software process is planned in detail for a large time frame. This work well if the application domain and software technologies are well understood by the developer in advance. Whereas, agile methods are developed to adapt and thrive on frequent changes.

VII. LIMITATIONS OF AGILE METHODS

Agile development aims to support early and quick development of working code that meets the changing needs of the customer. Agile supporters claim that code is the only deliverable that matters, whereas, agile opponents found that emphasis on code will lead to memory loss, because the amount of documentation and modeling done is not enough.

Agile methodologies are not suitable for greenfield engineering and not suitable for maintenance, since there will be not much Documentation for the systems. The second limitation is that agile methodologies depend heavily on the user involvement, and thus, the success of the project will depend on the cooperation and communication of the user. Another limitation is that agile methodologies concentrate work quality on the skills and behaviors of the developers, as the design of the modules and sub-modules are created mainly by single developer.

To get the advantages of applying agile methodologies in the development, there is a set of assumptions that are assumed to be true. To mention some are-- cooperation and face to face relation between the customers and the development team; evolving and changing requirements of the project; developers having good individual skills and experiences; in addition to many more. If these assumptions do not apply to a software development project, then it is better to look for other methodologies to apply for the development process, in order to get better results.

VIII. CONCLUSIONS

Software development methodologies have evolved since the 70s. Agile methodologies came into existence after the need for a light way to do software development in order to accommodate changing requirements environment. Agile methodologies provide some practices that facilitate communication between the developer and the customer, and undergo develop-deliver-feedback cycles, to have more specific view of the requirements, and be ready for any change at any time. The main aim of agile methodologies is to deliver what is needed when it is needed. Agile methodologies include a set of software development approaches. They have some variations, but still they share the same basic concepts. The main agile methodologies that are being used include XP, Agile Modeling, and SCRUM. XP is the coding of what the customer specifies, and the testing of that code.

Agile methodologies are not best suited for all

projects. When communication between the developer and the customer is difficult, or when the development team includes mainly beginners, agile methodologies will not give the best results.

IX. FUTURE SCOPE

It has been analyzed in this paper that agile software development is the future of software development methods. However it's too early to conclude that this approach is flawless. There are many areas that need to be researched before coming to a conclusion regarding efficacy of this approach. For example – most of the studies till now have included single customer representative. What if there are more stakeholders who need to be consulted in each iteration. Also we need to have empirical findings to substantiate above mentioned conclusion. In addition this approach needs to be tested on different platforms and different types of software catering to different industries.

REFERENCES

[1] Malik, Siew Hock, "Review of Agile Methodologies in software development", International journals of research and reviews in Applied Sciences, vol 1, issue 1, Oct 2009.

[2] Tore Dyba "Empirical studies of agile software development: A systematic Approach," Elsevier.com , 2008.

[3] Jeffrey A. Livermore, "Factors that significantly impact the implementation of an agile software development methodology", Journal of software, Vol 3, no-4, 2008

[4] Mary Poppendieck, tom Poppendieck, "Lean Software Development An Agile Toolkit", Addison-Wesley, 2003, Pg 32.

[5] Pekka, Outi, Jussi and Warsta, "Agile software development methods- review and analysis", VTT Electronics, 2002

[6] D. Turk, R. France and B. Rumpe, "Limitations of agile software processes", In Proceedings of the Third International Conference on Extreme Programming and Flexible Processes in Software Engineering, 2002.

[7] Cockbburn, A., "Agile Software Development", Boston, Addison –wesley, 2002.

[8] Palmer, S.R. and Felsing, "A practical guide to feature driven development", Upper saddle river, NJ, Pretntice hall, 2002.

[9] Schewaber and Beedle, "Agile software development with scrum", UUper saddle River, NJ, Prentice Hall, 2002.

[10] Glass, R., "Agile versus traditional: Make love, not war!", Cutter IT Journal, Vol. 14, No. 12, December 2001, pg 12-18

[11] Beck, K. "Embracing change with extreme programming", IEEE Computer 32(10): 70-77, 1999

[12] Stapleton, J., "Dynamic System Development Method-the method in practice", Addison Wesley, 1997

Garima Verma is working as Assistant Professor in Faculty of Computational Science and Applications at DIT University, Dehradun. She received Gold Medal in M.Tech. (CSE) from Uttarakhand Technical University. She has more than 12 yrs of industry and academic experience.