



## INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

### A Study of Agile Software Development

Shevali Agarwal<sup>\*1</sup>, Anurag Punde<sup>2</sup>

Department of Computer Science, Acropolis Institute of Technology & Research, Indore, India  
Superimpectron@gmail.com

#### Abstract

As we know that till now software development life cycle is the best way that gives assurance of the quality in the project. But as agile comes in to picture, the productivity of the product may increase to the large extent. SDLC and agile both are different in many ways. This paper deals with the comparative study of agile processes and benefits over the different model used in SDLC. The paper will serve as guide to other software development process models. Agile processes have important applications in the areas of software project management, software schedule management, etc. In particular the aim of agile processes is to satisfy the customer, faster development times with lower defects rate. In this paper we will also discuss the advantages and disadvantages of the agile process.

**Keywords:** Agile Development, Software Development Life Cycle (SDLC).

#### Introduction

Agile development software is over and over again referred as “frivolous” process towards the software development process. According to the Agile Manifesto [1] agile is based on a set of principles that focus on customer value, iterative and incremental delivery, intense collaboration, small integrated teams, self-organization and small and continuous improvements [2]. In software development life cycle, there are two main considerations, one is to emphasize on process and the other is the quality of the software and process itself. Agile software processes is an iterative and incremental based development, where requirements are changeable according to customer needs. It helps in adaptive planning, iterative development and time boxing. It is a theoretical framework that promotes foreseen interactions throughout the development cycle. There are several SDLC models like spiral, waterfall, RAD which has their own advantages. SDLC is a framework that describes the activities performed at each stage of a software development life cycle [3]. It depends on the various applications to choose the specific model. In this paper, however, we will study the agile processes and its methodologies. Agile process is itself a software development process [4]. Agile process is an iterative approach in which customer satisfaction is at highest priority as the customer has direct involvement in the evaluating the software [5].

#### **This paper is organized as follows:**

Section 2: discuss the models of the agile process.

Section3: Discuss the Characteristics of Agile processes.

Section4: discusses the advantages of Agile Process.

Section5: discusses the disadvantages of Agile Process

Section6: discuss the conclusion.

#### Models of the Agile Process

There are various models available for agile process which helps in the development of the software. The use of agile models increases the productivity of the software.

##### **Scrum**

Scrum is the framework in which you can apply various processes and techniques within which complex products can be developed [6]. Scrum is simple. It can be easily understood and implemented. The key principle of the scrum is that whenever customer wants to change their requirement they can change the same. This cannot be possible with the use of traditional model. Instead of understands the full problem, scrum model forces or focuses upon the team ability to deliver quickly and respond to promising requirements.

##### **Extreme Programming**

Extreme Programming (XP) is an agile software development methodology focused on a set of values, principles, and practices that its founders believe results in higher quality software and a higher quality of life for the development team. XP is the most prescriptive of the agile methods when it comes to appropriate engineering practices for software development. The key belief of teams practicing XP is that if a team follows a specific minimal set of engineering practices **extremely** well,

they will be much more successful and have much more rewarding software development experiences.

Some of the principles are as follows [7]:

- **Humanity:** balancing individual and team needs
- **Economics:** built on the time value of money and the option value of systems and teams
- **Mutual benefit:** "the most important XP principle and the most difficult to adhere to"
- **Self-similarity:** make the small echo the large (and vice versa)
- **Flow:** from lean manufacturing, not from psychology – deliver a steady flow of value by engaging in all development activities simultaneously
- **Opportunity:** see problems as opportunities
- **Redundancy**
- **Failure:** "If you're having trouble succeeding, fail."

#### **Feature Driven Development**

FDD is a model-driven short-iteration process that consists of five basic activities [8]. For accurate state reporting and keeping track of the software development project, [milestones](#) that mark the progress made on each feature are defined.

Activities of Feature driven development:

##### ➤ **Develop overall model**

In this we will propose high level walkthrough models to find the scope of the system and its context and then we will divide it into small groups, which are then onwards used for close review.

##### ➤ **Plan by feature**

After the feature list had been completed, the next step was to produce the development plan.

##### ➤ **Design by feature**

A [design package](#) was produced for each feature. A chief programmer selected a small group of features that are to be developed within two weeks.

##### ➤ **Build by feature**

After a successful design inspection a per feature activity to produce a completed client-valued function (feature) is being produced.

#### **Characteristics of Agile**

- ❖ **Modularity:** it is a way of defining the application in an efficient manner. It also increases the understandability of the application. Modularity allows a process to be broken into methods called activities. A software development process prescribes a set of activities capable of transforming the vision of the software system into reality.

- ❖ **Iterative:** Agile process is an iterative process. Because it complete the whole applications into small cycles. It identified all the problems in that small cycles and recover them soon. It also iteratively interacts with the client for any changes.
- ❖ **Time Bound:** It is basically depends upon the actual planning for the application to be done by the company and strictly follows by the employee. The time limit of the application normally lies between the one week to six weeks in the duration.
- ❖ **People Oriented:** The developers can increase the efficiency, productivity, quality and performance of the product by closely examine and make proper changes in the application.
- ❖ **Collaborative:** Agile processes foster communication among team members. Communication is a vital part of any software development project. When a project is developed in pieces, understanding how the pieces fit together is vital to creating the finished product. There is more to integration than simple communication. Quickly integrating a large project while increments are being developed in parallel, requires collaboration.

#### **Advantages**

- ❖ There is a regular involvement of the customer at any stage of the application development. In case of misunderstandings can be developed in the mid of the project, the customer can be developed their vision for the final product.
- ❖ Agile methods are iterative in nature. By this we mean that if any changes occur during the product phase we can do it easily. As we know that nothing is static. So Agility will make a better and efficient way to develop the application.
- ❖ Most of the time this happens the customer requirement and expectation is not fulfill by the company. This is either because the misunderstanding or because of the lack of communication between the customer and the company.
- ❖ It motivated to the development team by providing a way to decide their own guidelines for how to begin and end up the product in the time limit.
- ❖ The regularly interaction with customer can fulfill the expectation of them and also increase the quality and productivity of the product.

### Disadvantages

- ❖ In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- ❖ There is lack of emphasis on necessary designing and documentation.
- ❖ The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- ❖ Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

[9] Expert reference series of white paper.

### Conclusion

Agile software development process is somehow some way differ from the models of Software development life cycle. Agile is increasing the customer interaction throughout the application; it is iterative and more over is raising the productivity and the quality of the application. We can use agile processes whenever we need to perform certain changes to be implemented. Agile gives better way to improve the productivity and the quality.

### References

- [1] K. Beck et al. (2001, August 2012). Manifesto for Agile Software Development. The Agile Manifesto [Online]. Available: [www.agilemanifesto.org](http://www.agilemanifesto.org).
- [2] V. Lalsing, S. Kishnah, S. Pudaruth, "People factors in agile software development and project management", International Journal of Software Engineering & Applications (IJSEA) 3(1), pp. 117-137, 2012.
- [3] Tobin J Lehman, Akhilesh Sharma , "Software Development as a service: Agile Experiences", in annual SRII Global Conference (2011).
- [4] A. Ahmed, S. Ahmad, Dr. N Ehsan, E. Mirza, S.Z. Sarwar, "Agile Software Development: mpact on Productivity and Qulaity", in the Proceedings of IEEE ICMIT.(2010).
- [5] B.Boehm and R.Turner, "Balancing Agility and Discipline: A Guide for the Perplexed, Addison, Wesley, 2003.
- [6] Kem Shawber Scrum Guide May 2009.
- [7] <http://xp123.com/articles/review-extreme-programming-explained-2e/>
- [8] Palmer, S.R., & Felsing, J.M. (2002). A Practical Guide to Feature-Driven Development. Prentice Hall. (ISBN 0-13-067615-2).