# ANALYSIS AND DESIGN OF A NOVEL AGILE METHODOLOGY

Geeta Kocher*

Dr. Naveeta Mehta**

## *ABSTRACT*

*The rapid development in the software engineering field requires new technologies, as traditional approaches are inadequate. Customers expect the software to conform to their needs, even though they are not always able to define them exactly beforehand. The general trend in software development methodologies has been to change from plan–driven approaches to more iterative incremental development approaches. This change has given birth to a group of methodologies called "agile methodologies. Various methodologies are available in agile like Extreme Programming, Scrum, Feature Driven Development, DSDM etc which are discussed briefly in this paper. Each methodology has its pros and cons. There is not a single methodology which is suitable in all, so an new agile methodology is required which is having the features like independent of project type, according to the situation select its usage based and then develop a plan based on Feature List / Functionality/ Sprint/ Iteration, so that the maximum return on investment can be attained in minimum time. In this paper a new agile methodology is proposed which can be helpful in achieving such objectives.*

***Keywords:*** *Extreme Programming, Scrum, Feature Driven Development, DSDM.*

*Research Scholar, Maharishi Markendeshwar University, Mullana, Ambala

** Associate Professor, Maharishi Markendeshwar University, Mullana, Ambala

## 1.0 INTRODUCTION

The success of any computer software depends on the user's satisfaction. When software fulfills the user's requirements, it succeeds but the software fails if its users are dissatisfied. Old software development approaches are not able to satisfy the new requirements of the market in the best way. As a result, new software development approaches are evolved & agile methodologies came into existence [4, 6].

Conforming to the plans had not been the primary goal anymore and the projects had to respond to requirement changes instead. Agile development methods promise to bring a solution to this need. They take a different perspective to software development in comparison to the traditional models.

Agile means cutting down the big picture in to small size bits & fit them together at the right time. It involves planning what one wants and then adopting these plans to the results. It is a people oriented approach to software development which enables the people to respond effectively to change. The software is prepared according to the changing requirements. These approaches are meant to increase fastness and flexibility in the software projects [5]. Agile can be said as adaptive approach which is a sound choice for software development or web design projects.

Agility is dynamic, context-specific, aggressively change embracing and growth-oriented.

→ **"The core concept in agile is quick response to change**."

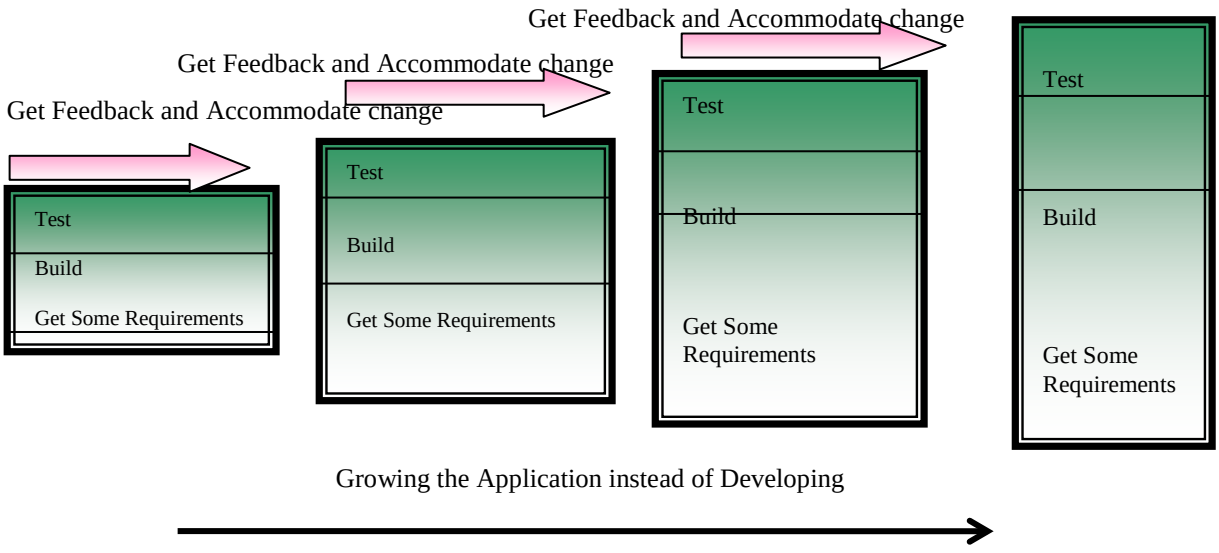


**Figure 1 Agile Approach [1]**

# 2.0 AGILE METHODOLOGIES

## 2.1 Extreme Programming(XP)

XP can be described as a set of rules rather than methodology of working and is intended to improve software quality and responsiveness to the change of customer's requirements. It is a deliverable and disciplined approach to software management which takes a code-centric view of the activity.XP is targeted at small co-located teams developing new critical products. The main objective of this approach is to deliver what the customer needs at the time it is needed [3]. XP is known as a methodology that "reduces bugs" in software, which is a great selling point.

XP is a new way of building the right software, and building it fast, to a high quality and explain how it has been tried on real projects that proved it work with a small development team[2,8].

## 2.2 Scrum

Scrum's primary goal is to deliver software that, after each and every iteration, provides the highest business value. Scrum is an agile software development process designed to add energy, focus, clarity, and transparency to project teams developing software systems. It is based on a 30-day iteration called a "Sprint." Technically Sprints can be either two or four weeks, but the generally accepted default is usually four weeks[7]. It will not define the way how software is developed, what documents are to be produced, how requirements should be gathered rather it is a guide how an agile implementation team should be managed. Scrum is not an independent stand alone development methodology; rather it is a wrapper for existing engineering processes. It manages and controls development work incrementally, improves communications, removes the problems to development with all in environment with rapidly changing requirements.

## 2.3 Lean Development(LD)

LD is more of a software development management philosophy than a methodology.In 1989, Professor James Womack and consultant Daniel Jones published 'Lean Thinking' a survey of the lean manufacturing techniques that helped create the "Japanese miracle" of the late 1980s and early 1990s. They chronicled the ideas of lean manufacturing, with their focus on eliminating waste, creating a smooth "flow" of work on the factory floor, and expecting workers to contribute high skill levels and an ownership mentality.

The fundamental principle is to create change tolerant software.LD is based on the Lean Thinking concept from Lean Production, that let customers delay their decisions about what

exactly they want for as long as possible, and when they ask for something it must be given to them so fast they don't have time to change their minds. It is used in any software development project where there is need for radical change.

### 2.4 Feature Driven Development(FDD)

Feature Driven Development is an agile process that can be applied to moderately sized and larger software projects[7] has some common characteristics with XP such as Short iterations, Customer representative on the team & using the notion of features which are comparable to the XP. Prioritization are encouraged in FDD and it is ensured that most valuable features are delivered. FDD feature short iteration cycles i.e. Two weeks. This methodology recommends individual code ownership and clearly defined roles.

### 2.5 Dynamic Systems Development Method (DSDM)

DSDM is a lightweight software methodology, which has fix time for the life of a project, and resources are fixed as far as possible. DSDM is focused on customer solutions and business value[5,9].

A comparison table is made for all methodologies, on the basis of the various parameters like usability, flexibility, acceptability and suitability (Table 1). Finally, after comparing all the agile methodologies, we can conclude that the XP is best among all. Inspite of being the best among the existing agile methodologies, the XP has some drawbacks which shall be overcome with the proposed agile methodology.

**TABLE 1 : COMPARISON BETWEEN VARIOUS METHODOLOGIES**

| Parameters | XP | Scrum | FDD | DSDM |
|---|---|---|---|---|
| Team Size | 3-16 | 5-9 | 6-15 | 2-6 |
| Practices | Engineering | Management Practices | Feature Based | Based on engineering and management practices |
| Flexibility | Allowed to do changes with in a sprint | Don't allow to do changes with in a sprint. | Allows changes with in a feature | Allowed to change based upon a fixed time line and fixed resources. |
| Usability | Focus on usability | Ignores usability | Focus on particular feature | Allowed to change based upon a fixed time line and fixed resources |

| | | | | |
|---|---|---|---|---|
| Length of sprints | 1 or 2 weeks | 2 weeks to 1 month | Takes 2 or less weeks .if takes further time, it is decomposed | Fixed time |
| Acceptability | High | High | Low | Low |
| Validation of software or testing | validation all the times | Validation is completed after a sprint during sprint review not at each step with in a sprint | At last stage | Testing in all phases of a project |
| Suitability | Accepts certain / uncertain requirements | Requirements certainty is required | Proves better when requirements are stable | --- |
| Project Size | Small | Small | Moderate and large size | Small and Large |

## 3.0 PROPOSED AGILE METHODOLOGY

Different Agile Methodologies have been proposed and exploited. These various methodologies exhibits various abilities and disabilities to produce a successful software product. But an agile methodology is required which is having the features like independent of project type, according to the situation select its usage based and then develop a plan based on Feature List / Functionality/ Sprint/ Iteration , so that the maximum return on investment can be attained in minimum time.

The proposed methodology exhibits the following features like small team size, more adaptable to changes, having design phase, work week according to the project need, proper documentation and supporting code-refactoring with code ownership. This proposed methodology is a methodology that provides guidance for the major phases to be followed during development of software.

Proposed Agile Methodology consists of six phases. It starts by estimating the project requirements, size, complexity and risk of the project Then the usage, team size as well as flexibility is measured. In the third step it is decided which methodology will be suitable for fulfill the above requirements. In the fourth step, how the project should be developed is decided. In the next step, the final product. is obtained. In the sixth step, the developed

product is handover to the customer and on the basis of customer suggestions implement/review or maintain the system.

The blueprint of the proposed agile methodology is as follows.

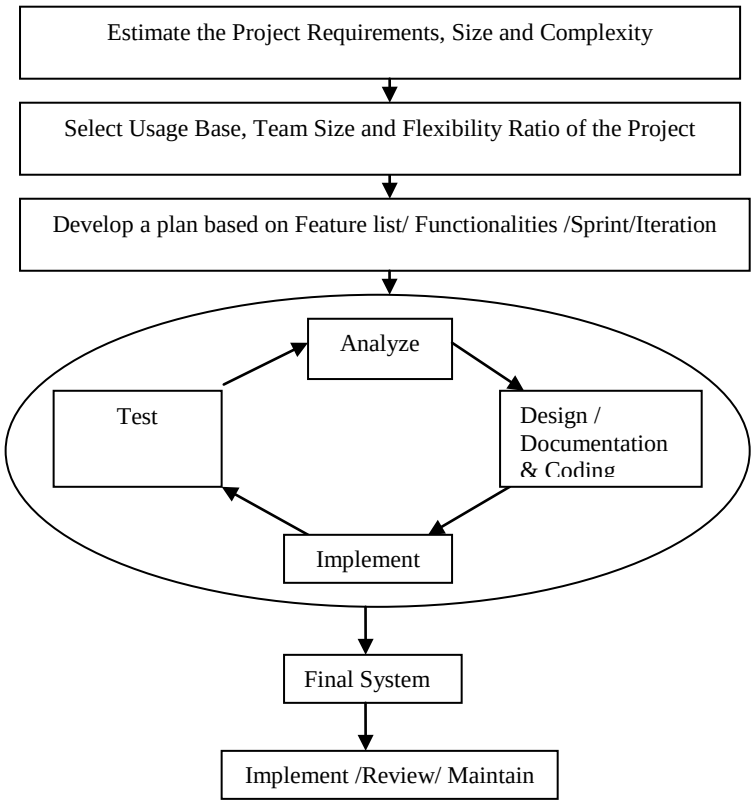


**Figure 2: Proposed Agile Methodology**

**Step1: Estimate the Project Requirements, Size and Complexity and Risk:**

In first step of proposed methodology, all requirements are taken from the customer and after taking customer requirements size, complexity and risk of the project is measured.

- **Project Size**

  After taking the customer requirements the next work is to measure the project size. To measure the project size various parameters can be used like how many people are needed in the development team, what will be the estimated cost of the project, what is the complexity of the project, what will be the development time frame of the project and on the basis of these parameters project size can be estimated.

- **Complexity estimate of Project**

  Different matrixes are used to measure the complexity and cyclomatic complexity is one of them. It is calculated by counting the number of decision points found in the code and provides a single ordinal number that can be compared to the complexity of

other programs. Cyclomatic complexity for a software module is based on the following equation:

$$CC=E-N+ p$$

Where

- CC = Cyclomatic Complexity
- E = the number of edges of the graph
- N = the number of nodes of the graph
- p = the number of connected components

Here following guidelines can be used to understand the equation.

- Start with 1 for a straight path through the routine.
- Add 1 for each of the following keywords or their equivalent: if, while, repeat, for, and, or.
- Add 1 for each case in a switch statement.
- **Risk Evaluation**

After calculating the complexity, risk is evaluated. Some level of risks are available in projects and it is important to identify and manage those risks to save the project from failure. The following table is used to measure the risk assessment.

**Table 2: Risk Evaluation**

| Cyclomatic Complexity | Risk Evaluation |
|---|---|
| 1 to 10 | A simple program , without very much risk |
| 11 to 20 | A more complex program, moderate risk |
| 21 to 50 | A complex, high risk program |
| >50 | An un-testable program means very high risk |

**Step 2: Usage Base / Team Size and Flexibility Ratio of the Project:**

After completing the first step next step is to measure the usage base, team size and flexibility ratio of the project. To achieve second step, start it from the Usage base.

- **Usage base:** Usage base means how the project is utilized for the benefit of the customer. Here it is checked that at which place the project is utilized and who are the potential users of the project, then software is developed according to the usage base of the customer.
- **Team Size:** The team size is selected after considering the usage base of the customer. It is decided that what should be the team size so that project should be

completed within time. Different methodologies have different team sizes. Teams should be small and variable in size.

- **Flexibility Ratio of the project**: Project should be highly flexible to the requirements as it is the main feature of agile methodology. Flexibility is valuable (and thus desirable) when new information can be obtained, i.e. one expects to "learn" about the future. The value of this flexibility is directly related to the value of this information.

$$\text{Flexibility Ratio of Project} = \frac{Change\ in\ \text{Re}quirements}{Total\ \text{Re}quirements} * 100$$

**Step 3: Develop a Plan based on Feature list/Functionalities/Sprint/Iteration**

During this step, a plan is get developed which decides that the final project will evolve with sprints/ iteration/feature driven/functionality based. After this, the actual development starts.

**Step 4:** This step involves four phases which repeats until the final project is ready. The details of four phases are as follows.

I. First of all requirements are prioritized. Then highly prioritized requirements are firstly analyzed for their validity. The possibility of incorporating the requirements in the system to be developed is also studied.

II. In design phase overall system architecture is defined, documented and then coding is done.

III. In Implementation phase work is started for developing it.

IV. In testing phase the prepared model is tested by user and takes suggestions as well as new requirements from the user and on the basis of these process repeats until the project is fully completed.

**Step 5: Final System**

In this phase project is ready and a final system is prepared.

**Step 6: Implement /Review / Maintain**

The system which is developed in the previous stage is ready for implementing. Project is handover to the customer and take the reviews of the customer. If still any changes are required from the customer then do changes in the product and what steps should be taken for maintaining it is also decided in this phase. So proposed methodology completes a project in the six steps with the faster and easier development of the system.

## 4.0 CONCLUSION

The proposed agile methodology appears to cover all the likely phases of the software development. A plan is developed based on Sprint/ Iteration / Feature list according to the flexibility ratio of the project which also influence the final output that results faster and easier development of the software. The proposed methodology exhibits the following features like small team size, more adaptable to changes, having design phase, work week according to the project need, proper documentation and supporting code-refactoring with code ownership. Therefore the proposed agile methodology addresses the issues of developing a new software system with the rapidly changing environments, thus it has great applicability in agile development environment.

## 5.0 REFERENCES


1. A.Khan, S.Babloo," A Tale of two Methodologies: Heavyweight versus Agile", The Tenth Australian World Wide Web Conference, from 3rd to 7th July 2004
2. Beck, K. & Fowler, M., Planning Extreme Programming (XP Series). Addison Wesley, 2001.
3. "Extreme Programming"www.extremeprogramming.org
4. Jeffrey A. Livermore," Factors that Significantly Impact the implementation of an Agile Software
5. Kuda Nageswara Rao et. al.," A Study of the Agile Software Development Methods, Applicability and Implications in Industry", International Journal of Software Engineering and Its Applications Vol. 5 No. 2, April, 2011.
6. Malik Hneif, Siew Hock Ow, "Review of Agile Methodologies in Software Development", International Journal of Research and Reviews in Applied Sciences, Volume 1, Issue 1, October 2009.
7. Roger. S. Pressman, "Software Engineering – A Practitioner's Approach", Sixth addition, 2005.
8. Sylvia Ilieva et. al, "Analyses of an agile methodology implementation", IEEE Proceedings of the 30th EUROMICRO Conference,2004.
9. Veerapaneni Esther Jyothi et. al., "Effective implementation of agile practices", International Journal of Advanced Computer Science and Applications, Vol. 2, No.3, March 2011.