Chapter 1

INTRODUCTION

1.1 Motivation

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. To succeed in software development, the discipline is needed in software design. Software development is a very crucial task on which success of software depends.

There are various traditional methods to develop a software like Waterfall Model, Spiral Model RAD Model etc. But there are some drawbacks in Traditional Models like they give more emphasis on documentation, less involvement of user, generally the traditional models are sequential, if any changes are required in the software, means going right back to the beginning and start developing new code all over again. This is not possible for most of the projects today. 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.

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. In agile software development, responding to change is emphasized.

So the focus of dissertation is to tell about the shortcomings of traditional methods of software development, study different existing agile methodologies and select the best one and try to propose a new approach.
Figure 1.1 - Traditional Approach

Figure 1.2 - Agile Approach
1.2 Agile Methodology Overview

In the year 2001, a wise software professionals group convened to discuss commonality among these methodologies. These software professionals coined the term "agile" to encompass all that was in common among the methodologies.

Agile means cutting down the big picture into small size bits & fit them together at the right time. Agile involves planning what one wants and then adopting these plans to the results. Waterfall or Classical model can be said as predictive where as agile approach can be said as adaptive approach. It is a sound choice for software development or web design projects.

1.2.1 Meaning of Agile

It is a low over-head method that emphasis value and principles rather than processes. It is based on empirical rather than defined method. In Empirical methods the study depends on practical experience rather than theories, whereas in defined method like Waterfall plans come first and then enforce those plans.

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

→ “The core concept in agile is quick response to change.”

1.2.2 Definitions of Agile According To Various Authors

According to Unhelkar (2010), agile methods are developer-centric, starting with the developer writing part of the code and demonstrating it to the user in an effort to engage them in a conversation. This interaction through the code further clarifies and illustrates the requirements that are built-in in the code [32].
According to Badr’s (2006) agile approach to software development is the process of rapidly creating the end application by introducing efficiency measures [5].

1.3 Challenges

Though, agile is a very popular practice of software development, now days. But the major challenges for agile are as follows:

1.3.1 Quality, Cost, Time and Scope

In practice, majority of projects have fixed budget and having a mandatory deadline. Agile being flexible allows frequent changes in scope. The change in scope means that cost, quality or time has to change.

Hence, even though a project follows agile methodology, at ‘certain point’ the flexibility has to stop.

1.3.2 ‘Ready to use’ product

‘Ready to use’ product with limited features is always available throughout the development cycle is one of the benefit of the agile methodology. Hence, due to any external constraints if the project has to cut short, business have a fall back product.

For the software development, it is not believed that this benefit can be realized in practice for most of the projects. Majority of projects need performance testing and tuning. The basic principle of performance testing warrants a stable functional application to be tested on. An application under agile development process is always developing and may have functional defects. Hence, performance testing can only be performed after substantial number of deliveries. During the development stage, the deliverable is never in ‘ready to use’ form unless it is performance tested.
1.3.3 Inability to ‘design’ for future requirements

Irrespective of best design models and most experienced design personnel on a project team, it is very hard to design a system on the basis of unseen requirements. This often leads to ‘rework’ at various stages in development and testing.

1.3.4 External and Internal Dependencies

Majority of projects have external dependencies. These dependencies are out of control of core project team. External teams will not always follow the agile methodology and hence, will deliver at a certain point in time. Once external deliveries are made only then dependant work can only be undertaken.

The problem arises when these dependencies are only discovered during the development process. Working on same agile project, there can also be internal dependencies between the different agile teams. The selection of work items in iteration may heavily dependent on the deliverables from other teams. This made planning in each iteration very tough, and huge coordination efforts are needed here.

1.3.5 Need of experts in agile team

It is must that each member of team must be an expert in his respective field because each iteration is only for a short duration. In reality it is not possible. Hence, often there is a danger to overrun the estimated time for work items.

1.3.6 Lack of Agile Development Methodologies

Different agile methodologies exist but all these are situation dependent like sprint or iteration. There is a lack of agile methodologies that can be implemented in all situations.
1.4 Need of Agile

1. **Evolving Requirements**
   
   Customer requirements are continuously changing due to various legislative issues and business needs. Customers do not have a clear vision about the specifications of their requirements at the early stages [11].

2. **Customer involvement**

   If the customer is not properly involved in the project, it leads to higher chances of project failure. The involvement of customers is very less in traditional methods, so the customers are generally dissatisfied. So for the satisfaction of customers, alternative approaches are found and for this reason agile approaches came into existence. In agile approaches customer is involved throughout the project or software development.

3. **Miscommunications**

   Miscommunications between developers and customers are also one cause of the need of new methodologies. In old methodologies if once wrong information is given, it continues till the end of the product and result of this is the build product or software not according to the requirements of the customer but in agile because the customer is involved, so the chances of miscommunication is about zero percent and if it happens, the problem can be solved.

1.5 Conclusion

This chapter explored the need of agile methodologies in software development field. The limitations of traditional methods motivate the user to adopt agile approach. Various challenges are discussed in this chapter which needs to be resolved. The next chapter summarizes the literature review on various agile methodologies.