CHAPTER 1

INTRODUCTION

1.1 GENERAL

Every decade starting from 1960s had its distinct impact on the software development approaches. During 1960s the limited requirements on software development as understood by the developers were incorporated in a straight forward manner. Based on an intuitive plan in their minds, it was implemented as a source code in one of the high level languages available during that time. There were no other activity phases involved during those years. In the early 1970s Royce proposed the waterfall model which had distinctive phases during development such as analysis, design, coding and testing. Many projects failed during those days and the customer was also unhappy as his expectations on the quality of delivered software were not fulfilled. During 1980s, the iterative model of software development lifecycle taking account of customer feedback during development became more popular. Object oriented programming approach to software development got established. The decade of 1990s experienced diverse trends in software development models. Some of them emphasized faster adaptation to changing needs. Although there were many different successful software development approaches that evolved, most of them followed stringent practices with emphasis on extensive documentation and hence they were called heavy weight methodologies. The traditional software development teams have a composition of individuals with different organizational roles such as business analyst, designers, developers, and testers. All these software models with rigorous documentation depend on verification and validation to ensure quality. This document centric approach for high quality of software delivered was considered very bureaucratic by a significant number of software professionals. On the other hand they emphasized end product quality directly and they did not wish to follow prescribed formal methods rigorously to achieve the above objective.

The traditional software development method deals with project phases such as requirements gathering, analysis, design, implementation, testing, and maintenance. As each activity phase gets completed, the project manager needs to get a sign off from the customer. In all these projects, project manager plays a vital role in allocating tasks to different developers, coordinating them, tracking progress and act as an intermediary between the customer and the members of development team. Such a process followed in traditional approach is called as "plan driven and contract based". While many adopted such models that had stringent practices, a few people criticized these practices. Some of the critics, championing alternate approaches to meet evolving customer needs, joined hands together to form a group and proposed 'agile' (meaning swift) approaches leading to several light weight models for software development.

Agile software development teams are self-organizing and self motivating (Cockburn and Highsmith 2001, Schwaber 2011). They have individuals who manage their own work load, who coordinate and communicate within the team members and who participate in decision making within the team and outside the team (Highsmith 2009). Agile teams exhibit autonomy, cross fertilization of ideas and mutual trust among stakeholders (Takeuchi and Nanoka 1986). Common features in these proposals were well recognized when Fowler and sixteen other developers released their `Agile Manifesto' (Fowler and Highsmith 2001). The manifesto contains four salient points which break down to twelve different agile principles. These agile approaches were not prescriptive with rigorous documentation like in conventional development models. On the other hand, they set guidelines to quickly realize the needed software through close collaboration with end users resulting in a product delivered through a series of increments. These methods take note of inherent shortcomings of software developers and users while they recognize the need for tolerance or deviations from the earlier disciplined approach.

Agile methods are called light weight methods as the weight of documentation is less and the practitioners spend more time on collaborative activity. Some of the agile methods that are popularly used are Extreme Programming and Scrum framework. However, traditional approach is still followed in most software companies because of its stability and predictability. Many of these organizations are not willing to take a risk of transitioning to an agile approach as the agile methods are not yet well demonstrated for mission critical projects or the ones with long lasting applications.

Traditional approach requires a complete set of requirements in order to move to the analysis phase. But, that is not the case with respect to many real life projects. Idea behind agile approach is to achieve better software quality even with evolving project needs. According to Larman (2004), iterative and incremental development methodologies are based on developing the system in small and manageable pieces combined with continuous integration.

1.2 WHY CHOOSE AGILE SOFTWARE DEVELOPMENT AS THE RESEARCH TOPIC?

As an employee of a software development organization, an attempt was made to study and assimilate the software needs of industry.

Success and failure rates in different categories of projects were observed and assessments were made on the current trends in the industry after discussing the issues with experts who work on agile software development projects. After discussing with practitioners, it was found that the software industry here lacks a sense of persistent vision when trying to implement agile software development approach. A thorough analysis of the existing methods was expected to help evolving a set of recommendations that would favor successful transition to agile approach and result in higher rates of project success with better product quality.

1.3 OBJECTIVE OF THE STUDY

Objective of study is to bring out an understanding in the usage of agile methodologies for practitioners who are desirous of adopting agile methodology and who wish to obtain better results in their development projects. Clarity in understanding agile software process will benefit all stakeholders. It will help in molding new comers into the agile development team fast.

1.4 SCOPE OF THE STUDY

In the software development projects to-day, change in requirements has become the order of the day. Developer organization wishes to get quick returns on investments, good rapport with the customer and deliver frequent minimum marketable feature releases in the market. A study has been conducted by observations, face to face discussions and interactions with practitioners at different levels and belonging to various software organizations that practice agile software development. Aim of the study is to identify the existing gaps in the current practice and to list certain recommendations that will make contributions from participating professionals effective when working with agile software projects.

1.5 LIMITATIONS OF THE STUDY

A questionnaire was circulated to a limited set of professionals working in Indian software organizations. Interviewees were selected from various roles with varied levels of experience. Quality of data collection and the research result depended on skill and experience of the participants. Main limitation in the study is that the set of practices that have worked for one particular team may not work for another team. Care needs to be taken while generalizing the conclusions from this study.

1.6 STRUCTURE OF THESIS

Chapter 1: Introduction discusses about the context and interest in agile development method as the research topic.

Chapter 2: Review of literature describes the study made on traditional and agile software development methodologies based on published literature.

Chapter 3: Research approach deals with the way in which the research was conducted. It explains the strategies followed in collecting the relevant data.

Chapter 4: Software Process Improvement discusses the ways of improving the software process with both traditional and agile software development approaches.

Chapter 5: Roles of agile coach deals with the role played by agile coach in newly transitioned agile teams.

Chapter 6: Activity of team members discusses the shift in emphasis that they face when adopting the agile software practices.

Chapter 7: Contributions of customer and their representatives deals with the customer contribution in the agile projects

Chapter 8: Impact of top management on agile teams deal with the impact of support that the top management gives in implementing the agile software project in their organization.

Chapter 9: Conclusion describes the contribution of this thesis and the ideas for future work.