Chapter 2

LITERATURE REVIEW

2.1 Introduction

The objective of this chapter is to review the literature available on agile methodologies. Extensive research has already been done in the area of agile methodology. This section highlights the work of distinguished researchers.

Victor Szalvay [35] demonstrates the shortcomings of the waterfall approach while providing a solution in iterative, and more specifically, Agile methods. Waterfall is an “over the fence” approach; the requirements are solicited from the user and some time later the finished product is presented to the user. This is entirely unnatural because customers find it difficult to specify software perfectly without seeing it evolve and its progress. Here Scrum describes as, a popular agile project management method, that introduced the concept of empirical process control for the management of complex, changing software projects. Scrum states that straightforward defined processes alone cannot be used to effectively manage complex and dynamic software projects. Agile project management approaches balance the four variables (code, schedule, requirements, and quality) in software development which are interconnected. In the example it is said that rather than locking into decisions at the beginning of a project, organizations can reduce risks by leaving options open to decide at a better time when more accurate information is available. Agile is simply the latest theory that is widely replacing the traditional approach.

Asif Qumer et. al.[4] presents six aspects of an agile software development methodology: agility, abstraction, process, people, product and tools. This set of aspects is an attempt to provide a guiding vision or mental-model for an agile methodology. This paper explicitly describes in detail these aspects that are part of
our agile software solution framework (ASSF). These six aspects can be combined to generate various situation-specific configurations of agile methodologies by using a method engineering approach.

Sylvia Ilieva et. al. [31] presents eXPERT approach designed for projects aiming to deliver quality software, i.e. with low defect rate, in a short time. The projects that will gain from using eXPERT are projects that have no clear detail specification in the beginning & for projects that have no clear definitions about the final result. The client only has a broad-brush picture of the imagine system without knowing its detailed frames and final features. Prior to applying eXPERT approach, the client commitment to the project should be carefully assessed. The benefits achieved with a committed client can be easily lost if communication fails.

Mona Singh [25] describes SCRUM approaches which possess key challenges for usability. First, product goals are set without an adequate study of the user’s needs and context. The user stories selected may not be good enough from the usability perspective. Second, user stories of usability import may not be prioritized high enough. Third, given the fact that a product owner thinks in terms of the minimal marketable set of features in a just-in-time process, it is difficult for the development team to get a holistic view of the desired product or features. This report proposes U-SCRUM as a variant of the SCRUM methodology. U-SCRUM was based on the experience with having two product owners, one focused on usability and the other on the more conventional functions. The result is that U-SCRUM yields improved usability than SCRUM.

Malik Hneif et. al. [20] describes that 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 i.e agile methodologies are evolved mainly to solve such problem. The new methodologies include modifications to software development processes, to make them more productive and flexible. Agile methodologies are not best suited for all projects. When communication between the developer and the customer is difficult, agile methodologies will not give the best
results. These methodologies exhibit optimum results when there is a strong communication between the developer and the customer, and the development team compromises skilled team members. When there is a big chance for misunderstanding about the exact customer’s requirements, or when the deadlines and budgets are tight, then agile methodologies are among the best software development approaches to apply.

Kuda Nageswara Rao et. al. [15] used qualitative and quantitative research methodologies for achieving some objectives. The customer found it difficult to define their needs because of the fast changing technology and agile methods were designed to define the changing requirements in software environments. The methodologies like XP, DSDM, Lean Development and Feature Driven Development are described in this paper. Scrum fits well into small projects. Requirements can be prioritized in a well-structured manner in Scrum but tight customer collaboration is not possible in this method. XP can also be used for small projects but customer collaboration is not very strong in XP. DSDM approach is efficient in terms of budget and time. It provides a technique-independent process and is flexible in terms of requirement evolution. This approach is heavier than XP and Scrum but DSDM is based on user involvement which is not possible in every project. Agile approaches are meant to increase fastness and flexibility in the software projects.

Veerapaneni Esther Jyothi et.al. [34] describes various methodologies and compare various approaches. According to him XP is a deliverable and disciplined approach to agile software development and XP is successful because it stresses customer satisfaction and allows the software developers to confidently respond to changing software requirements even late in the lifecycle. Scrum approach has been developed for managing the systems development process and concentrates on how the team members should function in order to produce the system flexibly in a constantly changing environment. 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, it is preferred to fix time and resources, and then adjust the amount of functionality accordingly. DSDM is a non-profit and non proprietory framework for rapid application development.
The hybrid approach that is using scrum and XP engineering practices is also suggested by the author. Since Scrum doesn’t have any engineering practices and XP doesn’t have any management practices, XP with scrum projects allows better value metrics process for measuring and managing initiative ROI. The author also suggests that companies can have a Quality Facilitator to eliminate all the discussed worthwhile risks to the extent possible to produce a defect free product.

Hesam Chiniforooshan Esfahani et.al. [10] discusses that in agile methods heavy dependence on the human factor possess new challenges for organizations intent on adopting them. A goal oriented approach is proposed in this paper which is for modeling social aspects of agile methods to complements other more activity oriented views. It focuses on the representation of the relationships and dependencies between the roles involved in the process, the responsibilities of each role & the skills needed to play each role. With this approach, organizations may visualize & evaluate how well the different agile methods may fit with their existing internal structure and minimize the possible risks derived from their adoption.

C.Balan. et.al. [6] emphasis the need to adopt agile methods for the development of Cyber Forencies Tools. The development of these tools is more challenging as more and more variations of digital devices are marked rapidly.

Kapil Kumar et. al. [14] gives more emphasis on change issue. In modern competitive software environment changes are very frequent. So to handle these changes, a proposed model is given i.e CFDD (Cognizant Feature Driven Development) where changes are handled separately from the development phase and the benefit of this model is its simplicity and quick development with less choice of failure.

Vahid Rahiman et. al. [33] tells that agile methods are appropriate methods for the development of Mobile software. For this purpose, a hybrid method engineering approach is discussed.
Mira Kajko Mattsson et. al. [24] discussed the problems regarding agile development. The case studies and the problems regarding development are discussed. These are grouped in six classes: Culture, Time Zone, Communication, Customer Collaboration, Trust, Training and Technical issues. The solutions to solve such problems are also discussed in this paper. Time zone problems can be removed by shifting the working hours of both onshore and offshore teams by adjusting direct meetings to the time zones or by creating asynchronous meetings via project managers. Holiday and Vacancy problem was partly remedied by creating a team calendar aiding in project planning. Trust problem was resolved by frequent meetings in various forms such as video conferencing, personal rotations and team building exercise & technical and training problems were overcome by presenting detailed descriptions of daily work or allowing teams to work with the technology they were comfortable with and support each other regarding various technical problems. So in this way problems as well as solutions are discussed in this paper.

Mikio Aoyama [23] describes that internet changed software development’s top priority from what to when. Reduced time-to-market has become the competitive edge that leading companies strive for. Thus, reducing the development cycle is now one of software engineering’s most important missions. Conventional software process models, such as the waterfall model, are monolithic and slow, focusing as they do on a single long cycle time. Further, conventional process management is based on the volume of requirements. The greater the requirements’ functionality, the longer the project’s time-to-delivery and the lower the process’s flexibility and productivity. Conventional management principles also lead to various scheduling problems, such as achieving on-time delivery, because it is hard to estimate the exact volume of work involved at the project planning stage. To address these issues, ASP alters traditional management principles as follows:

- The process architecture shifts from monolithic to modular,
- The process dynamics shift from volume-based to time-based.
Agility in software development means not only quick delivery of software products but also quick adaptation to changing requirements.

Lene Pries-Heje[18] considered scrum a good agile project management approach. It works so well because it provides communication, social integration, control, and coordination mechanisms that are especially useful for distributed and agile project management. It also can build up relations and networks within and for the project team and can build up trust even at a distance. Other reasons are also mentioned in this paper for considering it a good agile approach.

Stefan Cronholm [30] focus on the movement from traditional to agile methods. Various terms regarding expected benefits of agile methods like ‘rationality’, ‘team work’, ‘adaptability’, ‘structure’, ‘less documentation’, ‘late changes’, ‘simplicity’, ‘creativity’ and ‘improvisation’ are discussed in this paper. Some common preserved effects like rationality and structure are also described in this paper which is common in both methods i.e. traditional or agile methods.

Michael Coram and Shawn Bohner [22] describe the impact of agile methods on software project management. To illuminate some of the strengths and weaknesses of project, the project managers can make more informed decisions and this impact is measured in terms of People, Process, and Project. Agile Methods offer a reasonable approach for the high degree of change and uncertainty in today’s software development. There are proven principles employed in Agile Methods that, when applied singularly under the right circumstances, result in lower risk projects and ultimately better productivity and quality. Additionally, when these are combined with other agile principles, there can be a synergy that provides even more traction on the project goals.

A.Khan and S.Balbo [1] describes that the basic difference between agile and heavy methods is the weight. Agilists believe that agility can not be achieved with heaviness. Heavy Agile is an oxymoron. One has to be light to be called agile, and the reason for this lightness is to counterattack change. The differences between agile and heavy methods are described in this paper.
Addicam V. Sanjay [2] gives the overview of different agile methods. The characteristics of various agile methodologies, history behind agile methodologies, areas where to use agile methods and where it should not be used is also discussed in this paper.

Glen Litton Van der Vyver [8] presents an overview of agile methodologies and the key features of the agile corporation. It is examined whether agile methodologies are used in practice and what organizational factors facilitate or inhibit their adoption. Agility demands that the corporation position itself to thrive in uncertain situations. Agility impacts structure but process is the most important factor. Processes must be minimalist, adaptive and rapid.

Naresh Kumar [26] says Extreme programming is a good agile methodology when the team size is generally small i.e. from 2 to 10 and XP is not suitable for distributed teams. Test driven development (TDD) is the modern art of development which is also integrated with the agile methodology for faster and accurate development of requirements. It also supports the change request implementations more accurately in faster manners. According to author, Incremental software models are very popular because of its simplicity of executing the requirements in stages. The designers prefer this model because of its properties of executing in steps. Since most of time all the requirements are not clear to the development team and requirements generally changes at time to time and are clear to the team at middle of the development, so incremental model support the development team to provide this type of environment where the efforts can be utilized towards the successful execution of the project.

Marilyn Crowell [21] describes that agile methods may not be suitable for all types of projects; especially large mission-critical software projects, those projects with more than 10-12 team members, and projects with several junior programmers. For most software projects today, which don’t need to follow a “process control model” there may be great benefits to going agile. Agile development methods strive for technical excellence, which produces much higher quality software for less money and in less time.
Alexandre Magno Figueiredo[3] considered scrum to be an excellent alternative not only for software projects, but for projects where the urgency is high, creating a team spirit, expectations alignment and change preparation.

Jeffrey A. Livermore [13] says that traditional methodologies are proven cumbersome to fulfill the requirements of the user. So agile methodologies came into existence. The use of agile methodologies improve the development process by removing barriers to accepting business requirement changes during the development process and enables software developers to produce higher quality software in a shorter period of time. Under management’s control several factors like training on the methodology, active management involvement and support, access to external resources, and company size impact the implementation of an agile SDM. Different methodologies require different changes to the organization’s management and software development cultures.

2.2 Conclusion

This chapter gives the literature review on agile methodologies and the areas where agile approaches can be used. The work of distinguished researchers are highlighted and discussed in this chapter.