Agile Methods for Software Development

Harshit Manchanda*, Ashwani Agarwal*, Divyanshu Bhati*, Ilango P
Scope, VIT University, Vellore

Abstract: The smooth development of any software project is mainly characterized by the software development methodology that is adopted by it. One of the best software development methodology is the Agile methodology where no prior planning is needed to handle the future problems, it in fact works on the principle of handling the situations on the go. It is known to deal with any scenarios spontaneously without any prior planning. The Agile software development refers to the set of rules and regulations which specifies the requirements and proposes solutions to software development problems through the collaboration of multidisciplinary teams of self-organization. However, the agile methodologies are not considered to be too secure since the number of developers working on a software project which is following the agile development methodology is not fixed. The requirements from customers are not fixed and this makes the development phase a little less secure. The collaboration of agile methodology with any other software development methodology makes it more reliable and secure. This paper presents the review of several different cases of software development phases which works on the principles of agile development methodologies.

Keywords: Agile methodology, Iterative development, Real time development, Extreme programming, Crystal methods, Collaboration, Scrum.

1. Introduction

Software development process (SDM) developed over the last few decades to optimize and improve software development process; for example: Rational Unified Process, Open Unified Process, Agile Software Development, Unified Essential Process, Microsoft Solutions Executive, Unified Process and Agile Unified Process. Today, there are many new software companies started and more to come. In despite this, there are many defects in software products and bankruptcy business stories. Many of them because processes Software inadequacy or methods used. In general, the selection of software methodology on the basis of experience or, in some cases, rational use. Extreme Programming (XP) is the most agile application methodology. XP is based on a series of concept that include having business customers on site, by Programming, the collective ownership of the code, the continuous code Integration, small launches, design testing before writing Code meetings stand-up, refactoring and work 40 hours Weeks. It is observed that the implementation of collaborative practices and Agile values are difficult. Therefore, educators must pay attention to the teaching of these practices and course values of Software development.

2. Organization of the document

The paper is structured as follows. The first section gives a brief Introduction of the agile methodologies used for the development of a software development project. The next section showcases the review of the paper that are addressed in the reference section. The following section follows the results, graphs and visualisations deduced from the related papers which is followed by a concluding section giving an outlook of the reviewed papers.

3. Literature Review

The paper [1] focus on the benefits of implementing Agile Methodology in solving real world problems. The author took the case of Mexican Airlines. In 2001, after the attacks on the World Trade Centre in New York the company was suffering from the economic depression. As a result the way the company did business and the planning priorities changed in few months. Since the traditional system was not automated the situation demands a fully automate software system for better planning and decision making, within limited amount of time and least cost of expenditure. One of the authors was involved in all the phases of the software development as a project leader giving advice to the client as to how to adopt Agile Methodologies as a way to give results more quickly, implementing usable and acceptable quality software, based on a perspective of incremental deliveries. According to author, as the result of adopting Agile Methodology for software development reflect significant reduction in the cost and time.

As the result of automated software the company’s decision making process reduced from four days to few hours and hence add up to significant response in handling economic stress. Thus the following conclusion can be made from the above findings that the maturity of the software development teams is a primary condition to be able to choose the appropriate methodology and is a dominant factor in developing them. It is very important to develop new ways to work with the users and consumers for good software
development. It is also important to have a compact team that is less changing and remains fixed for most of the part.

The paper [2] focuses on the security issue related to the use of Agile Methodology for software development approach. According to the author the approach is being criticize for the product security issue, so the author attempts to identify the challenges helps the secure development of software using agile practices. While conducting the survey the author uses Interview as the main source of collecting data. As the result of the survey the author identifies “Continuous-integrations”, “Planning-game” and “Pair-programming” phases of agile methodology as security challenges while developing software. At last the author admits the lack of knowledge about software security as the limitation to the research.

The research [3] aims to examining the suitability of agile methodologies for some kinds of information systems such as Decision Support System. In this paper they have compared various development strategies and compared against agile development process. The conclusion that they make out of this is that there is no quick fix that always leads software development. This is especially true in the rapidly changing technological environment. However, for the development of DSS, the author recommend agile methods because of the reasons mentioned. Discussion on the current development approaches DSS illustrates the importance of the concept of evolution for the development of SAD. Here they have divided the various factors of the problem into three main categories: human factors, conceptual factors and technical factors. This methodological proposal arose from the combination and adaptation of models and methodologies that are widely used in Information system research, such as; Socio-Technical Model for Information system. The terms such as adaptation capture and evolution of the organic nature of the development of a DSS. He did not research to investigate the development of SAD using new SE methodologies, such as agile development methods. The good news for the agile community and the DSS community is that XP seeds have enormous potential to grow and prosper in the development of DSS. We have described the first step of an ongoing research effort to set up a software development process for DSS construction, the next steps of this research are underway, should agile development methodologies be very relevant for development SAD who is not satisfied with traditional methodologies.

In the paper [4] author aims to check the compatibility of Agile methodology frameworks with the Mobile application development process and hence enlighten the path connecting them. The paper highlights the types of mobile application and its characteristics. The author recommend few points before choosing suitable Agile methodology such as examining mobile application type and background Also application demand should be considered based on current technology and mobile devices usage. As the result of the study the author finds, agile software development has its potential to be fit in the mobile application development. Moreover author propose Dynamic systems development method (DSDM) best suitable in large scale mobile application due to large team and mobile application characteristics.

The paper [5] proposes that right selection of a software development process is an essential activity in any software project. It has a major impact on customer satisfaction and the well-being of businesses. Normally, selection is based on experience or, in some cases, rational. In Third World countries and particularly in Sudan, where they carried out this research, choosing a software methodology is always a struggle in software companies. It is probably because most of these companies are starting up or having modest resources, such as small and medium-sized enterprises (SMEs). At present, the Government of Sudan (the main buyer of local software) is pushing the software industry to produce quality products and reduce the risk of purchasing software products from foreign companies. Quality products are real concern for these companies. Typically, software companies or start up SMEs looking for rapid and agile development, accompanied by plug & play engineering practices. Agile software development process have recently become on the road to quality assurance. Since the last decade, agile methods have increased popularity and use in software industries. These methods have come to meet rapidly evolving needs, satisfy customers, promote interaction, communication and produce high quality products. There is much debate in the literature on agile methods, especially in the new and used. However, academic research on how these SMEs can choose the appropriate method remains limited. The purpose of this article is to fill this gap in a systematic review of the literature on the most Agile software development methods. This document has three objectives: first, to propose a definition and a discussion of the most agile methods of agile software development. On the other hand, works the challenges of SMEs and formulates software in the criteria of comparison. Third, compare these methods and shows the similarities and differences with the defined criteria. On the basis of this analysis, future research needs are identified and discussed.

The paper [6] articulates the representation of a multicriteria approach based on a more intelligent approach, which can be useful to support decisions regarding the choice of agile software development methodology best suited for small and medium
This research project has studied the application of Agile SDM using an online survey sent to practitioners around the world. These survey data were used to identify factors associated with the implementation of Agile SDM.

Factors include training, participation of management, access to external resources and size of the company have been approved in the implementation of an agile software development method. Factors such as the use of models, development of an implementation plan, and implementation of development team development and software for the use of Internet or intranet will have no impact on the application of a software development method agile.

Many of the factors that affect the application of an agile development methodology are under the control of management. Organizations that are considering implementing an agile methodology may want to handle some of these factors to increase the chances of success of their methodology.

Many software development projects do not achieve their goals or are disrupted. This leads to economic problems, loss of well-being and may even endanger humans. In many cases, an inappropriate choice of software development methodology (SDM) - and if it does not follow an ideology towards development - can be identified as the root problem. This paper defines a set of parameters in choosing an appropriate methodology, however, is not an insignificant attempt. The conflicting objectives of conventional methodologies and agile approaches add additional complexity. The author identify the common features that can be found in each MDS and have a structured approach to classification. The author also present a mapping of existing methodologies to highlight the applicability of their approach. This allows the comparison of basic, integral and agile SDM. In addition, the author discuss the results and highlight their implications for the choice of an SDM. The author contribute to the theoretical knowledge and advise the companies at the same time.

The author present the work in the characterization of software development methodologies. Based on a thorough analysis of background and related work, the author present their scientific approach to design. The common features of iterative MDS were identified and presented as a framework. Each methodology can be checked for its fundamental ideas in relation to a number of characteristics, i.e. for the disciplines of support, stages of development, strategy procedure, definition of depth, mandatory practices and assistance procurement.

To illustrate and evaluate their approach, they applied it to goods and methodologies that highlighted possible mapping. Mapping the cascade model, the
Rational Unified Process, Scrum and Extreme Programming shows the feasibility of their work. In addition, these sections provide a number of additional reading sources. Finally, the author discuss the impact of their results.

A number of problems were raised and future work in various directions can be based on their work. In particular, developing a method to allow companies to select SDM specific project to meet their needs in terms of time, budget and project quality is very attractive, but highly non-trivial. Their framework can also be combined with the work of other authors or used in conjunction with it. A combination of theoretical and empirical studies is particularly attractive. Their work will continue with refinements and extensions of their framework.

The paper [9] focuses on the reasons why many software development projects do not achieve their goals or are disrupted. This leads to economic problems, loss of well-being and may even endanger humans. In many cases, an inappropriate choice of software development methodology (SDM) - and if it does not follow an ideology towards development - can be identified as the root problem. This paper defines a set of parameters in choosing an appropriate methodology, however, is not an insignificant attempt. The conflicting objectives of conventional methodologies and agile approaches add additional complexity. The author identify the common features that can be found in each MDS and have a structured approach to classification. The author also present a mapping of existing methodologies to highlight the applicability of their approach. This allows the comparison of basic, integral and agile SDM. In addition, the author discuss the results and highlight their implications for the choice of an SDM. The author contribute to the theoretical knowledge and advise the companies at the same time.

The author present the work in the characterization of software development methodologies. Based on a thorough analysis of background and related work, the author present their scientific approach to design. The common features of iterative MDS were identified and presented as a framework. Each methodology can be checked for its fundamental ideas in relation to a number of characteristics, i.e. for the disciplines of support, stages of development, strategy procedure, definition of depth, mandatory practices and assistance procurement. To illustrate and evaluate their approach, they applied it to goods and methodologies that highlighted possible mapping. Mapping the cascade model, the Rational Unified Process, Scrum and Extreme Programming shows the feasibility of their work. In addition, these sections provide a number of additional reading sources. Finally, the author discuss the impact of their results.

A number of problems were raised and future work in various directions can be based on their work. In particular, developing a method to allow companies to select SDM specific project to meet their needs in terms of time, budget and project quality is very attractive, but highly non-trivial. Their framework can also be combined with the work of other authors or used in conjunction with it. A combination of theoretical and empirical studies is particularly attractive. Their work will continue with refinements and extensions of their framework.

The paper [10] describes the problems faced in Agile software development methods. Agile software development methods have been developed and evolved since early 1990s. Due to the short development life cycle through an iterative and incremental process, the agile methods have been used widely in business sectors where requirements are relatively unstable. This paper explains the differences between traditional software development methods and agile software development methods, and introduces the characteristics of one of the popular agile methods, Scrum. Finally, the paper describes the issues and challenges discovered through an in-depth case study in a company which has employed Scrum for many projects. The insights presented in the paper can be used in organizations that are in the process of agile software development using Scrum.

Agile software development methods have been developed to provide greater customer satisfaction, shorten the development lifecycle, reduce error rates and reflect the changing needs of the company during the development process. This article presents the characteristics of traditional methods of software development and of agile methods of software development, and the differences between them. This article also describes the functions, ceremonies, and artifacts of Scrum, which is one of the most agile methods of software development best known in the industry. This article also presents five issues and challenges, including documentation, communication, user engagement, work environment and Scrum ceremonies, discovered through the in-depth case study of a software company that makes small and medium size web applications. If all five problems and challenges are addressed and resolved before the project begins, organizations will have less difficulty producing high-quality software products using Scrum.

This paper [11] is describing that Extreme Programming (XP) is one of the most widely used agile methods for software development. It intends to improve the quality of software and responsiveness to changing customer needs. Despite the fact that the use of XP offers Many benefits and an agile methodology has been widely used, XP does not offer the same benefits in terms of Means and large software
projects. Some of the reasons are the lack of documentation, the lack of Unaware of risk awareness during software development. Due to the increasing demand for agile approaches, the Study addresses the problem of the ability to manage medium and large XP projects. Most companies using XP Development methodology for medium to large projects facing this problem, reflecting the importance of this problem. AT In the study, the XP model is extended which also offers its advantages Large-scale projects. An extended XP evaluation, are conducted three independent industrial case studies. The results are concluded mainly on the basis of autopsy Rate of analysis and failure by KLOC. Extended XP model it indicates its adaptation for medium and large projects complete post-mortem analysis within 2 - 4 h (as recommended for small projects). An average increase the time and effort to publish the mortem analysis note.

KLOC due to project has increased, while an average decrease in the time and effort of the release of because learning and development author Extended XP project teams compared to one XP already taking rations for AQ Test the quality point has extended the XP model is better, Equal or less than the existing XP model. It is to highlight that Quality XP model expanded is better than the current XP due to fewer KLOC failures for all three case studies. Evaluation, the author He found evidence that the proposed extended XP the model is suitable for the development of medium and large Projects. However, the need for validation proposed extended model will be directed XP Comparing come with future versions. The results provide a proof that can be extended XP Beneficial for medium and large software development projects.

The paper [12] describes agile methodology was adopted for many Software development projects because of their ability to manage With changes in product requirements, while traditional The methods are most suitable for projects Have clearly defined requirements. Because the differences in the two methodologies and their different Approaches to solving development problems is Necessary to understand individual approaches. Arrange for benefits to be compared and synthesized. This study will be necessary to understand the differences and Diversity of both methodologies using the checklist Table to select and adopt the appropriate methodology for Specific development projects. The results of this study are not Only give positive answers, but also offer suggestions for Better integration. The third step was combined with traditional. Method and should take about three Months to complete. This method was adopted to solve problems. Problems of design. The project was a success in terms of time. The satisfaction of delivery and the user, but that does not mean that appropriate approach for all projects, either inside or outside our team. We have learned from this project as “Software The choice of development methodology should be considered Characteristics and characteristics of the project Organizational Environment”.

4. Diagrams and visualizations

Figure 1. Scrum, example of an agile methodology

Figure 4. Different specification depths

| Table 1: differences of DSS from other information systems in term |
|----------------------|----------------|---------|---------|
|                      | User | Data | Models | Knowledge |
| DBMS                 | X    |      |        |          |
| MIS                  | X    | X    |        |          |
| ES                   |      |      | X      | X        |
| DSS/ EIS             | X    | X    | X      | X        |
5. Conclusion
The paper describes the problems faced in Agile software development methods. Agile software development methods have been developed and evolved since early 1990s. Due to the short development life cycle through an iterative and incremental process, the agile methods have been used widely in business sectors where requirements are relatively unstable. This paper explains the differences between traditional software development methods and agile software development methods, and introduces the characteristics of one of the popular agile methods, Scrum. Finally, the paper describes the issues and challenges discovered through an in-depth case study in a company which has employed Scrum for many projects. The insights presented in the paper can be used in organizations that are in the process of agile software development using Scrum.

Agile software development methods have been developed to provide greater customer satisfaction, shorten the development lifecycle, reduce error rates and reflect the changing needs of the company during the development process. This article presents the characteristics of traditional methods of software development and of agile methods of software development, and the differences between them. This article also describes the functions, ceremonies, and artifacts of Scrum, which is one of the most agile methods of software development best known in the industry. This article also presents five issues and challenges, including documentation, communication, user engagement, work environment and Scrum ceremonies, discovered through the in-depth case study of a software company that makes small and medium size web applications. If all five problems and challenges are addressed and resolved before the project begins, organizations will have less difficulty producing high-quality software products using Scrum.

6. References