International Conference on “Advances In Computing, Communication And Intelligence” ICACCI-2014 At Mauritius
Special Issue of International Journal of Electronics, Communication & Soft Computing Science and Engineering, ISSN: 2277-9477

A Novel Efficient Software Engineering Models for Agile Methods

Ms. Shweta S. Gulhane Dr. Mohammad Atique Dr. P.P. Karde Dr. V.M. Thakare

Abstract:- **Software Engineering is an approach to developing software that attempts to treat it as a formal process more like traditional engineering than the craft that many programmers believe it is. In proposed methodology Extended Iterative – Agile model used. In extended Iterative - Agile model first step is to finalise the requirement, architecture and design. Development phase begins as soon as requirements design and architecture is completed. Development phase follows agile method. Scrum masters ensures smooth development as per requirement and accommodates new changes, since architecture & design is already in place a scalable architecture allows new changes to be accommodated. This model reduces risk of changing architecture time to time. It increases the productivity and optimizes the requirement of resources. If there is any concern in development phase then it is attended by Team Lead and addressed with the help of Project Manager in case of higher complexity. This involves removing any impediments to progress, facilitating meetings, and doing things like working with the product owner to make sure the product backlog is in good shape and ready for the next sprint.**

Keywords – **Software engineering, Agile methods, Extreme programming (XP), Agile software development, Iterative model**

I. INTRODUCTION

A software development process, also known as a software development life cycle (SDLC), is a structure imposed on the development of a software product. It is often considered as a subset of system development life cycle. There are several models for such processes, each describing approaches to a variety of activities that take place during the process.

A software development process is a structure imposed on the development of a software product. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. It aims to be the standard that defines all the tasks required for developing and maintaining software. Software Engineering processes are composed of many activities, notably the following:

- Requirements Analysis
- Specification
- Software architecture
- Implementation
- Testing
- Documentation
- Training and Support
- Maintenance

Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen tight interactions throughout the development cycle. Agile software development has had a huge impact on how software is developed worldwide [1]. Agile methods address the challenge of an unpredictable world, emphasizing the value component people and their relationships bring to software development. Agile methods for software developed emerged in the mid 1990s and focus on agility for software development. In essence, agility means responding to changes quickly and efficiently. There are many specific agile development methods. Most promote development, teamwork, collaboration, and process adaptability throughout the life-cycle of the project.

II. BACKGROUND

Agile Software Development is presently an emerging discipline in the field of Software Engineering. It is presently advocated by many software professionals. The Agile software development principles that are followed and advocated emerged from the traditional software development principles and various experiences based on the successes and failures in software projects. New methods, now called agile methods are were designed to define the changing requirements in software environments. Traditional methods refer to the older and commonly used methods like the waterfall methods. These traditional methods have often been criticized to be far from the real ways software engineers functioning in developing the software [1].

There are a number of agile software development methods. Methods for agile software development represent a set of practices for software development that have been created by experienced people. The most common methods are

- Extreme Programming (XP)
- Dynamic Software Development Method (DSDM)
- Scrum and
- Crystal

The highest criterion of these methods is development of the software and customer satisfaction through continues delivery of software. This is achieved by having short iterations in the development process. The iterations focus on timely delivery of working code that provides substantial value to the customer. More importantly, is a very limited work that has been produced to assess the applicability of agile methods. This is an indication is a need for investigation [2].

Extreme programming (XP) is one of the most widely used agile methodologies for software development. It intends to improve software quality and responsiveness to changing customer requirements. Certain scenarios which are discussed are weak documentation, lack of strong architecture and ignorance to risk awareness during the software development. Most of the companies that employ XP as a development methodology for medium and large projects face this problem, which echoes the importance of this problem. To address this problem, in this study XP model is extended in such a way that it equally offers its benefits for medium- and large-scale projects. As an evaluation of the extended XP, three independent industrial case studies are conducted. The case studies are described and results are presented in the study. The results provide evidence that the extended XP can be beneficial for medium and large software development projects. [3]

III. PREVIOUS WORK DONE

In 2005 a study looked at the state of research on XP, agile software development and agile modeling. With respect to XP, authors reported a small number of case studies and experience reports promoting XP’s success. A better established stream of research supports pair programming, and some studies also support iterative development. The authors recommended the separate study of other core XP practices to identify which once work. Furthermore in previous methodology saw challenges in matching agile software development methods with standards such as ISO [1].

According to [3] the method presented two case studies using adapted XP model. One case study was conducted with IBM for 12 months and other one was conducted with Sabre Airline Solutions for 3 months. An extreme programming evaluation framework (XP-EF) to adapt XP model. The framework used feedback loop throughout the project for the evaluation of agile team and procedures involved. Framework needs further validation through more case studies and needs to be improved particularly with respect to the XP adherence metrics. Traditional methodologies are not fulfilling the need of software industry to achieve fast development without compromising quality whereas agile methodologies cannot be directly implemented for medium and large development projects because of inadequate documentation, weak architecture and lack of risk management [3].

Agile development methods are suitable for small teams, but for larger projects, other processes are more appropriate. A clear finding from the previous work done published on agile methods is that there is a need to increase both the number and the quality software development [2].

IV. EXISTING METHODOLOGIES

To reduces risk of unexpected result at the time of production stage studies fell into four thematic groups. Introduction and Adoption studies didn’t provide a unified view of current practices. Instead they offered a broad picture of experience and some contradictory findings. A recurring study theme was human and social factors and how they affect and are affected by agile development methods. In this methodology researcher have studied conversation, standardization and progress tracking and describe them as mechanisms for creating awareness in teams and organizations [1].

To achieve fast development without compromising quality there is need to adapt the agile methodologies for medium and large projects. Extended XP model is used for medium and large projects in order to meet the industrial demand of an agile methodology. The main phases of extended XP methodology are

- Project Planning
- Analysis and Risk Management
- Design and Development
- Testing

The testing phase of the extended XP model is executed in the same manner as of the existing XP model. Fig. 1 shows the extended XP model. The extended XP model is multidimensional in nature and it is equally suitable for incremental (like existing XP) and parallel development (like traditional process model). It supports the development of medium and large software projects [3].

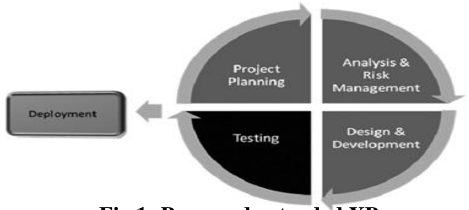


Fig 1: Proposed extended XP process model

To increase both the number and quality software development need to focus on following objectives:

To examine, gain insight into the agile methods and practices, To find out the issues in where , when and how agile methods are used, Strengths and weaknesses of agile methods, Understanding the applicability of agile methods, Understanding transition from traditional methods to agile methods and its effects on the organization. Agile methodologies like XP are capable of quickly adapting to the changing requirements of the customers. XP accommodates frequent changes in the software product and is tailored made for such purposes. XP facilitates frequent releases of the working software [2].

V. ANALYSIS AND DISCUSSIONS

XP seemed difficult to introduce in large, complex organizations but easier in other organization types. The benefits appeared in customer collaboration, work processes for handling defects, learning among developers, thinking ahead for management, focusing on current work for engineers and software estimation. In this research methodology researcher have studied conversation, standardization and progress tracking and described them as mechanisms for creating awareness in teams and organizations. By using comparative studies these studies compared variations of traditional development to variations of agile development. They showed that traditional and agile development methods use different project management practices [1].

To increase both the number and quality software development need to focus on current agile methods, benefits of agile methods and their level of applicability in industry means is there any relation between software developed and the agile philosophy adopted as well as any relation between size of the project/company and agile methods and impacts of using the agile methods in industry. If some agile methods are non popular, so it is of immediate concern to know how the agile methods are differ from each other. The given method found that when software development is done with several teams, there are problems with communication and coordination. Even face-face communication is not enough as there is overburden of requirements for planning. Certain projects had too many stakeholders that communication became difficult [2].

Project planning plays an important role for the success or failure of a project. A software project will deteriorate after few years if the project planning is not properly made and documented. Analysis phase improves quality of software through proper documentation. ‘Analysis and Risk Management’ phase in the proposed XP model has many benefits for example, stable requirements, strong architecture and risk management plan. Stable requirements facilitate a development team to achieve strong architectural design by aiding the factor of reuse and easy to evolve a software. Risk management plan helps a team to cater potential risks regarding the failure of a project. This is the phase where an analyst gathers detailed requirements/user stories.

The need for the statistical validation of extended XP model proposal will be addressed in days to come by comparing with market trends demand. The companies are not really encouraging in terns of using variety of agile methods. This is because of lack of sound proficiency in this field. It is therefore important that empirical research and investigation should be carried in this field [3].

Advantages and Disadvantages of three methods\ shown in Table 1.

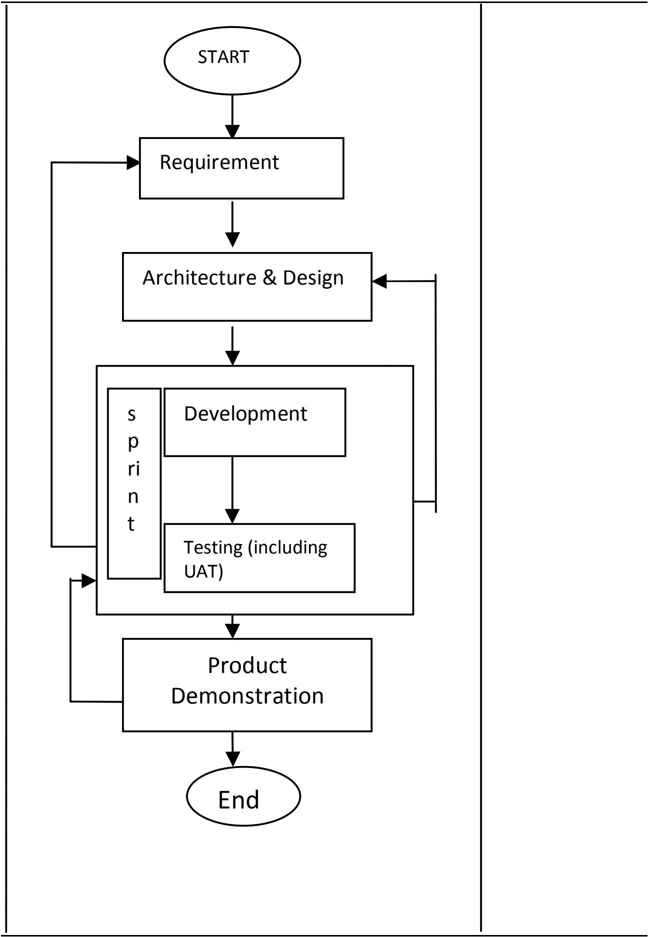
Methods	Advantages	Disadvantages
Extreme Programming (XP)	When having only one developer, pair programming and continuous review practices have to be dropped out.	Customer collaboration is not very strong.
Scrum	It fits into small projects.	Customer is offside and tight customer collaboration is not possible.
Dynamic Systems Development Method	This is heavier than XP & Scrum. It provides a technique-independent process and is flexible in terms of requirement evolution.	It is based on user involvement which is not possible in every project.

Table 1: Advantages and Disadvantages of three methods

VI. PROPOSED METHODOLOGY

In these previous methods the literature repeatedly mentions of lack attentions to design and architectural issues. The unsustainability of the on- site customers role for long periods and the difficulty of introducing agile method into large and complex projects. Dynamic system development model based on user involvement which is not possible in every project. Tests to be written for *everything* is very hard to do. It requires a good coach to initiate and ingrain the rules into peoples' heads. And it requires the people within the team to care enough to not let things get sour. But, it's almost impossible to get a team to be that meticulous. During the course of project there is continuous involvement of resources means for development, testing, architecture phases. Failure probability increases if large projects are handled by inexperienced members. If scrum master and business sponsor has clear product vision, it reduces the chances of failures. Therefore to overcome these problems the proposed method Extended Iterative - Agile Method used.

Extended Iterative - Agile Method: It is simulated and comprehensive model of iterative & agile process. This proposed method overcomes the design and architectural issues. It will also helpful in large and complex projects. Fig.2 shows the proposed extended iterative model.



`Fig 2: Extended Iterative – Agile Model

VII. POSSIBLE OUTCOME AND RESULTS

In this extended Iterative - Agile model firstly finalise the requirement, architecture and design. The requirement process can be divided into requirements analysis, requirements specifications and requirement validations. It may not be possible to obtain all the requirements in one round of discussion with the client. It will take round of discussions. Once these requirements are obtained they need to be documented and a document called “system requirement specification”. Validation phase begins post SRS is documented. Validation insures that there is no communication gap between the client and the development team. The validated system specifications document has to be converted into design. Design is for both hardware and software. Architecture & Design is then finalised with client. Once Requirements, Architecture & Design is finalised the development phase begins. In the development phase requirements are divided into smaller, more easily managed modules (product backlog) the way it is done in Extreme Programing. This smaller module reduces risk of changing architecture time to time. It increases the productivity and optimizes the requirement of resources.

Optimised resources reduce the engagement of all the members from day one which further provides lean team and significant cost reduction. Development phase follows agile method. Scrum masters ensures smooth development as per requirement and accommodates new changes, since architecture & design is already in place a scalable architecture allows new changes to be accommodated. The Scrum Masters further looks in to risks and its mitigation. This involves removing any impediments to progress, facilitating meetings, and working with the product owner closely to make sure the product backlog is in good shape and ready for the next sprint.

VIII. Conclusion

Extended Iterative - Agile Method utilizes the resources at the appropriate time and ensures Architecture and Design follows the requirements. New requirements are also accommodated through scalable architecture in the sprints planned during development phase. This methodology bridges the gap of challenges seen in Iterative as well as Extreme programing.

There is slight flexibility one can have during development when you're working on a design developed by different team as well as requirements captured by different team. However it may not function the way extreme programing offers completely new requirements at larger extent during development.

IX. Future Scope

There is larger scope of adapting to traditional basics of software development as well as new methodologies of addressing changing market trends. Inclusion of this methodology would allow industry to accommodate rapidly changing business trends.

References

[1] Tore Dyba and Torgeir Dingsoyr, "What Do We Know about Agile Software Development?", IEEE COMPUTER SOCIETY, VOL. NO. 0740-7459/09, PP. 6-9, 2009.

[2] Kuda Nageswara Rao, G. kavita naidu, Praneeth Chakka, "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, PP. 35-46, APRIL, 2011.

[3] M. Rizwan Jameel Qureshi, "Agile software development methodology for medium and large projects", IET SOFTWARE, VOL. 6, NO. 4, PP. 358- 363, 14 AUGUST, 2011.

[4] T.Dyba, B. Kitchenham and M. Jorgensen, " Evidence- Based Software Engineering for Practitioners", IEEE Software, VOL 22, NO. 1, PP. 58-65, 2005.

[5] M. Aoyama, " Web- based agile software development", IEEE software, VOL. 15, NO. 6, PP. 56-65, 1998.

[6] B. Boehm, " Get ready for agile methods with care", IEEE Computer, VOL. 35, NO. 1, PP. 64-69, 2002.

[7] T. Dyba, " Improvisation in small software organizations", IEEE Software, VOL. 17, NO. 5, PP. 82-87, 2000.

[8] Boehm B., Turner R., " Management challenges to implementing agile processes in traditional development organizations", IEEE Software, VOL. 22, NO. 5, PP. 30-39, 2005.

[9] Lindvall M., Muthing D., Dagnino A., et al.: "Agile software development in large organizations", IEEE Computer, VOL. 37, NO. 12, PP. 26-34, 2004.