ISSN:0975-9646

Impact of Different Methodologies in Software Development Process

Preeti Rai, Saru Dhir

Amity University
Sector-126, Noida, UP, India

***Abstract*—A major challenge in any engineering endeavour is taking a poorly organised, uncertain, inconsistent, incomplete, and over generalized requirements specification and turning it into a well-structured design. In order to manage quality and to save cost and time, many companies are shifting from traditional software development lifecycle models to the agile environment. This paper aims to describe the impact and comparison of various traditional methodologies and new methodology. This paper explores the reasons for which software industries shifted from Traditional RE to Agile RE. This research has performed comparative studies of different software development methodologies, when and where they can be used. This paper also gives introduction to agile software development methodology and how to apply them.**

***Keywords*— SDLC, ADE, ASDM, RE**

I. Introduction

To maintain the software expansion process, many effective quality systems are developed; which address an organization's commercial requirements. Designers employ various types of system development process model to direct the project's life cycle. Various activities may be done in various phases by an specific or team doing software development process. Classic activities performed in software development process includes: System planning, System requirements and benefits exploration, Project endorsement and project scoping, System design, Development, Software integration & testing, System integration & testing and documentation, Implementation and Maintenance. The major issue in software development is to regulate, how to instrument, using certain skills and within certain boundaries.

Software development lifecycle (SDLC) gives a brief outlook of how the dissimilar problem-solving events may be done in various chapters by an discrete or team doing software advance. The various SDLCs models are-

1) Waterfall
2) Iterative
3) Iterative and Incremental
4) Evolutionary Prototyping
5) Ad-hoc or Code-And-Fix SDLC

Numerous agile ideas have been around since 70's or smooth before and their substance is documented as a response against various traditional methods. Agile methods on an entire are new; they have healthy roots in the history of software engineering. The term agile can be defined by quickness, correctness, and ease of quantity and it has also increased the public consideration in late 1990's, Agile approaches were well recognized to progress systems more quickly with partial time spent on analysis and design.

II. Background

Various lifecycle development models have been created and also applied to large-scale system and software development projects, which are used in government sector various industries and many more different areas.

As a solution for the traditional software expansion approaches agile software development approaches (ASDMs) were well-known. Overhead the historical four centuries, outmoded waterfall expansion methods have been broadly rummage-sale for significant projects in the software manufacturing and in the government sector because of their straightforward, logical, and planned nature and for their capability to provide inevitability, control, and high declaration.

According to survey, the waterfall model is being used by a huge portion of the software engineering world [12]. In another learning of 1,027 IT projects in United Kingdom, Thomas (2001) have reported about scope management related to attempting waterfall practices and clarify that it was the solitary largest donating factor for dissatisfaction. Laffingwell (2007) has defined four main key expectations with the waterfall model which turned out to be not precise. The practices of prototyping and the object-oriented are main conducts to contrivance the waterfall, incremental, and spiral models. Agile is not a set of tools neither single methodology, but according to a philosophy which was formulated in 2001 with an initial 17 participants and it was in adjustment to the document-driven software development practices—ex waterfall. Hue designates; that in agile system enterprise is used, in place of formal construction, which can present a story that how system is worked by unfolding all segments and adornment of the system.

III. Related Work

A survey was conduct and in that according to Dennis, Wixom, and Teagarden (2005), the approximating phase, in waterfall model occupies typically about 15% of the total Systems of Development Life Cycle and in analysis phase, about 15% of the model, analyses the current system about its problems and then also classifies how to design the new system through requirements assembling. Now, coming to the agile methodologies, a survey conducted in 2008 which shows that 67% of agile practitioners experienced improvements in their software development process [12]. And also a survey conducted by Dr. Dobb's Journal shows 41% of development projects have now adopted agile methodology, and also agile techniques are being used on 65% of such projects [9]. A research study conducted by the Standish Group regarding 8,380 projects from 365 respondents representing companies across major industry , shows that only a small percentage of projects (16.2%) with traditional methods were completed on-time and budget with all features and functions specified and 52.7% of the projects were completed over-budget, and time estimate, and offering less features and functions; 31.1% of projects were cancelled at some point during the development cycle[9]. In a study of failure influences in 1027 IT projects in the UK, scope management associated with Waterfall practices was quoted to be the largest difficulties in 82% of the projects and approximately 13% of the projects surveyed didn't fail (Taylor 2000). A large project study, Chaos 2000 by The Standish Group showed that 45% of requirements in early specifications were never used (Johnson 2002).

Parrish (2004) argues that agile software development methods (ASDMs) provide increased quality, shorter time to market, better efficiency, and greater customer satisfaction. Miller and Larson(2005) also believe that ASDMs emphasize close collaboration between the users and developers of a project, and relatively quick development cycles that can react to changing requirements. Australian group, Shine Technologies (2003) surveyed 131 defendants of teams and companies that had applied agile methodologies [1] 93% stated that productivity was better ,49% stated that costs were reduced and [3] 88% stated that quality was much better, and 83% stated that business satisfaction was also great. Spencer (2005) stated that "our implementation of agile applies helps us find bugs earlier, and also to achieve higher quality, and helps us work well with QA.

IV. Software Development Life Cycles

Software development project has to go through the various stages such as- Requirement gathering, writing useful specifications, Creating policy design documents, Implementation coding testing and also quality assurance.

A. Requirement engineering

Requirements engineering has additional growing attention in both academia and industry, as today's software challenging systems are very much expected to provide a highly user-centric functions and abilities.

Traditional RE and Agile RE vary on the basis of their planning and control mechanisms. Research is carried out by a literature study which is rationalized by various testers in various corporations that which approach they are using, its impact and benefits, and also about the approach with better success rate than other. With the help of this results we have assessed that Agile RE performs much better than Traditional RE in large administrations which are branded by evolution of variations throughout the progress phase of the software life cycle.

B. Traditional Models

There are numerous development models for developing the software in different paradigms. Some of these models are explained as below:

1) Waterfall Model: It includes system, requirements, initial and detailed design, implementation, testing, operations, and maintenance. In the Boehm-Waterfall software engineering methodology the process interchanges from stage to stage. A corporation has involvement in building accounting systems, I/O controllers, and then building another such product based on the current enterprises is best managed with the waterfall model.

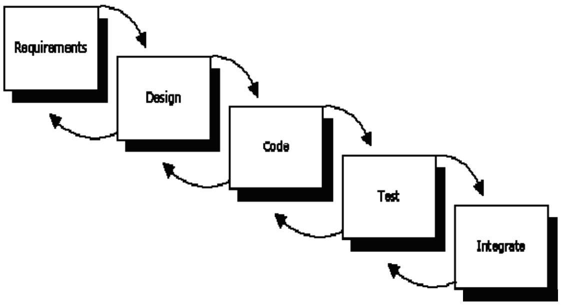

Fig 1: SDLC Model

Waterfall model are lacked in: Risk requirement put on hold until late stage. Document confirmation is done late and also change in requirements was taken adversely in waterfall. Operational problems exposed too late in the process (acceptance testing), alteration was based on functionality, not on quality features.

2) Prototyping Model: Prototyping is the process of building a working replica of a system. And it may be used with the waterfall, to create technical feasibility when technical risk is too high. It can also be used to better knowledge and extractment of user requirements. The basic goal is limitation of cost by understanding the problem before binding enough resources.

Prototyping model is lacked in: It is generally applicable to new and more original system and less to current one. Sometimes it can also produce a system with not a very good performance.

3) Incremental Model: A project which is using the incremental model may start with general objectives, which is also defined as provisions and those requirements are implemented by following the next coming portion of the objectives,tilll all objectives are performed it push difficult problems to the future to prove early success and this model can be useful for the projects where the basic software functionality is required at the begining.whereas in agile ,contextual figures are increasingly shaped and then review is done with the help of samples, to the customer or within the existing team members or may be with the product possessors. The incremental expansion should be measured when it is perilous to develop the whole system at once.

Incremental model is lacked in: It needs good planning and design. It also needs a strong and complete description of the whole system before it can be broken down and built incrementally. The total cost is higher in comparison to waterfall.

4) Spiral Model: The spiral size agrees to system size, while the detachment between the coils of the spiral indicates resources. It has basically four phases: Preparation, Risk Analysis, Engineering and Calculation. It is basically useful in ADE projects, as they are risky in nature. Business projects are more outmoded and generally use developed technology so this model basically applicable to business applications, particularly in cases where there is no guaranteed success and alsowhere applications require large calculations, ex, in decision support systems. During the risk examination phase, a process is undertaken to classify risk and substitute solutions, and at the end of which a prototype is produced. In the spiral model, progress is characterized by the pointy component, and the cost is signified by the radius of the spiral.

Spiral model is lacked in: It is a **c**ostly model to use. Risk analysis requires people with a particular expertise as the success of the project is dependent on the risk analysis phase. It does not work well for smaller projects.

V. AGILE METHODOLOGY

Various organizations are disinclined to avoid their traditional methods and jump into agile methods. Their disinclination is the result of various issues, which also includes the agile methods expressively reduce the amount of documentation and rely heavily on tacit knowledge, for mission/safety-critical projects, and it also belief that these methods are not passable for highly stable projects, a major concern that agile methods can be successful only with talented individuals who require many degrees of freedom, and that agile methods are not appropriate for very large-scale projects.

A. Agile Model

As the development for the traditional software development methods started to fail-Agile software development methods (ASDMs) were developed. The determination towards agile started in the mid-1990s this methodology can be describe as iterative and incremental development including flexibility throughout the systems development life cycle, minimal groundwork, light and fast development cycles, people-centric development, customer relationship, and regular delivery. In 2001, seventeen experts met at Snowbird, Utah, to discuss if there was anything in common among the various agile methods (Cockburn, 2007), and they formed the Strategy for Agile Software Development (Becket al., 2001), which also uncovered what items were unrushed precious by ASDMs. Highest priority is to accomplish the customer demands early and provide continuous delivery of esteemed software. It also appreciates the changing requirements, even if it affects the development process and the most proficient and working method of freeing information to and within a development team is face-to-face discussion. Working software is the primary measure of any progress. Agile actions endorse maintainable progress. The best styles, requirements and designs appear from self-organizing teams. At continuous breaks, the team reflects on how to become more effective, then tunes and adjusts its performance accordingly.

The twelve principles behind agile philosophy are:

P1. Our highest priority is to satisfy the customer through early and continuous delivery of products.

P2. Appreciate changing necessities, for the user modest benefit.

P3. Deliver working software frequently, within few days with a preference to the shorter timescale.

P4. Business customers and developers must work together on daily basis throughout the project.

P5. Projects should be developed around motivated people and they should be provided by the atmosphere and also by the support and trust to get the work completed.

P6. The most effective method of conveying information to and within a development team is face-to-face conversation.

P7. Working software is the primary measure of progress.

P8. Agile processes endorse supportable enlargement. The guarantors, designers, and employers should be able to maintain a continuous step open-endedly.

P9. Continuous attention to technical excellence and good design enhances agility.

P10. Simplicity the art of maximizing the amount of work not done is essential.

P11. The best constructions, necessities, and strategies combined from self-organizing squads.

P12. At regular intervals, the team reflects on how to become more efficient, then adjusts its behaviour accordingly.

B. The Process of Agile Software Development

Agile SDLC contains the six basic phases as shown in Figure –

- Pre project preparation
- Start
- Construction
- Production
- Selection
- Departure

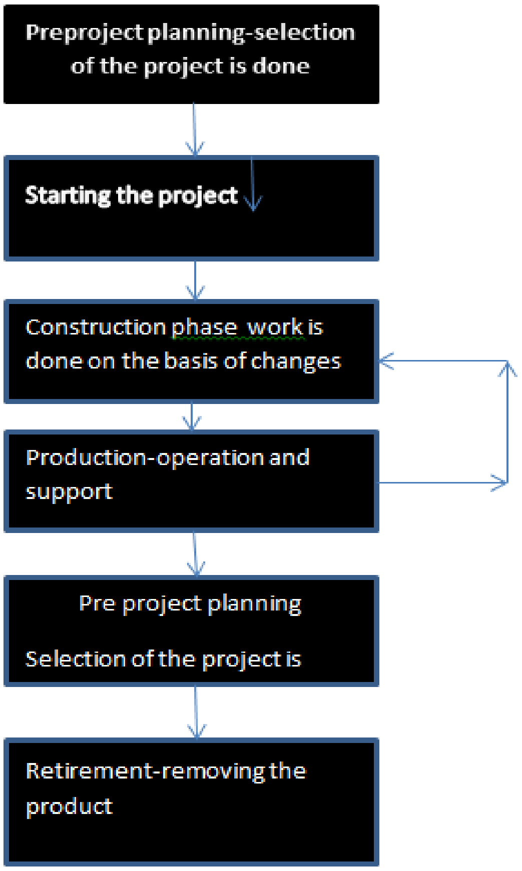
Preproject planning-selection of the project is done
Starting the project
Construction phase work is done on the basis of changes
Production-operation and support
Pre project planning
Selection of the project is
Retirement-removing the product

Fig 2: Process of Agile Software Development

1) Pre project scheduling: The basic of the project and the market features are well-defined in this phase.

2) Start-up-phase: Requirement displaying done through active participation of stakeholders in order to identify the initial requirement basic or high-level requirements for the system. The main objective is to understand the problem.

3) Construction: In this phase excellent working software is delivered incrementally, which meets the variable needs of the customer. Adjustment in the requirements is allowed to meet the exact needs of the customers. At the end of each progress iteration there is a fractional, working system which is shown to the customer and testing can be done on that like system integration testing.

4) Production: It handles all operations and provides support.s.

5) Selection: The goal of the construction phase is to keep systems useful and active even after the product has been deployed to the user.

6) Departure: It is also known as system deactivating phase.

The ASDMs have the potential to provide higher customer satisfaction, lower bug rate, shorter development cycles, and quicker adaptation to rapidly changing business requirements.

VI. Comparison Between Basic Models And Advanced Model

The comparison between Waterfall model (basic model) and Agile model (advanced model i.e. which comes into existence earlier) shows the periodically improvement towards software development life cycle models and it can be shown by following table-

TRADITONAL APPROACHES	AGILE APPROACHES
Measured and formal, direct ordering of steps, based on rules.	Developing, iterative and investigative, outside formal rules.
Everything is up front and large before starting.	The focus is on whether customer requirements are met in the current iteration
Documents and evaluation are needed to solve an issue	A short discussion can solve the problem .
Circumscribed access to architecture.	The whole team effects and understands the architecture and will be able to do a design presentation.
No communication within the team.	High level of communication and interaction, understanding groups, meetings
Extra time is spent on design so the product will be more maintainable.	There is no time for the what is unknown.
Modification necessities is difficult in later stages of the project	Can respond to customer requests and changes easier
Successive and synchronous process	Parallel and asynchronous process.
In this the environment is taken as steady and expected.	In this the environment is taken as tempestuous and difficult to predict

Table I: Comparison of Traditional Approaches with Agile Approaches

So, this was a old-fashioned approach versus agile approaches assessment, which tried to compare the basic variations between the two. An perfect model should combine the responsibility and accountability of a waterfall model, with the quickness and adaptableness of the agile model. Thus that can be an effective balance between the two principles that can create a more efficient software improvement model.

VII. Conclusion And Future Scope

A discussion on various enlargement models has been obtainable in this paper. Although many development models exist, this paper discusses different models out of those and the comparison includes the advantages and disadvantages of different models which can help to select specific model at specific situations depending on customer request and including business requirements.

It also describe about agile methodology, its various principles and steps. It also gives a comparison of agile and waterfall models, and also describes the benefits of agile over traditional methodology.

There are many limitations and parameters in various models. In future, our main focus is to conduct an interview from different industrialists, research scholars and calculate the results for estimation process. Currently our questionnaire related to future work is done and we do calculations using chi-square approach and scheming a model which will upsurge the performance of product and also compute cost and would be applicable in all kinds of software improvement process.

REFERENCES

[1] Steve Easterbrook, "Software Lifecycles", University of Toronto Department of Computer Science, 2001.

[2] Rashmi Popli1, Anita and Nareshchauhan “Mapping of Traditional Software Development Methods to Agile Methodologys”.

[3]M, Hue, 1. Varner, L. Zhu, and M, A. Babar, "Software quality and Agile Methods," Pros,28th Annual International Computer Software and Applications Conference (COMPSAC'04), pp., 520-525,2004.

[4] Guy Want, “Drowning in the Waterfall, The Benefits of Agile versus the Predominance of Waterfall “ Software Engineering CS 390:", October 29, 2008.

[5].Frye, Colleen. “Agile practitioners face challenges, but see process improvements.” Search Software Quality. 27 June 2008. 20 Oct 2008

[6] Lillian Doric and Elias Niemelae "A Survey on Software Architecture Analysis Methods”, IEEE Transaction on Software engineering, VOL. 28, NO. 7, JULY 2002

[7] Jeffrey A. Livermore Walsh College, “Factors that Impact Implementing an Agile Software Development Methodology”.

[8] Eduardo Malaga Chicano, “A comparative study if iterative protyping vs waterfall model processed by small and medium sized projects by system engineering”, 1996.

[9] Nabil Mohammed Ali Munassar, Dr. A. Govardhan, “Comparison study between Traditional and Object-5th Malaysian Conference in Software Engineering”,2011.

[10] S. Arshadi, S. Muhammadi and S. Shahzad,” Empirical Analysis of traditional and agile requirement process” Science .and Tech. and Dev., 32 (1): 44-47, 2013.

[11] Ashley Aitken ,Vishnu Lingo, “Comparative Analysis of Traditional Software Engineering and Agile Software Development”, 46th Hawaii International Conference on System Sciences,2013.

[12] David H. Olsen, Karina Hauser, "An Exploratory study on issues and challenges of Agile Software Development With SCRUM”, A dissertation submitted in partial fulfilment the requirements for the degree of DOCTOR OF PHILOSOPHY, 2010.

[13] Barbara Kitchen ham,” DESMET A method for evaluating Software Engineering methods and tools”, Technical Report TR96, 09August 1996.

[14] Gaurav Kumar, Pradeep Kumar Bhatia,” Impact of Agile Methodology on Software Development Process”,” ISSN 2249-6343International Journal of Computer Technology and Electronics Engineering (IJCTEE)Volume 2, Issue 4”,august 2012.