

Ground Axioms to Achieve Movables: Methodology

Sahil Aggarwal

Amity School of Engineering and Technology
Noida

Saru Dhir

Amity School of Engineering and Technology
Noida

ABSTRACT

To develop any software there is a need to go through particular phases which software should follow during its lifetime that is known as software development life cycle (SDLC). In SDLC both organizational standards and customer needs are fulfilled. Organization standards help in making quality of software and to meet market deadlines. The demand of current time is that the development system should be efficient enough to deliver, when the customers need and it should welcome changing in requirements at any phase of development.

This paper reviews the classical methodology and agile methodology of developing software and also explains that how the working of agile methodology is different from the classical methodology and it is the need of the day. This paper also suggests the conditions that are needed in both the methodologies and improvements of agile methodology so that this methodology of development can be implemented in the organization effectively.

Keywords

Agile Methodologies, Classical Methodologies, Return On Investment (ROI), Software Development Life Cycle (SDLC), Validation, Verification.

1. INTRODUCTION

In today's world, everything is growing fast; customers need the product on right time at minimal cost. Requirements are also changing frequently. So to remain in the market and to deliver the goods, time and cost have become very critical factor for every organization.

In software development method, as customers changing requirements frequently, produces problem to deliver the software at right time with the all given requirement specification. So escalation time is needed to complete the project hence unpredictability of the completing the project increases. So in the current scenario there is a need to resolve chaos and unpredictability [1] in the project which would be cost effective, and time effective. There are mainly two major methodologies for developing the software, which are known as classical or traditional development and agile methodology. In different conditions different types of methodologies are followed, some are new to the market and some are traditional one. New methodologies take time to be followed in an organization effectively. Frequent changing requirements by the customers and environment make it more difficult for the organizations to follow a particular methodology. Current situations need a methodology in which rules are flexible and can be changed according to the conditions and it will be better if there are no particular rules to implement software. But in this situation some ethics and principles need to be followed, which will be followed by every individual related to the development of software.

In section 2 the execution plans of developing software in traditional methodologies are shown with the difficulties in these approaches. In section 3 and 4 the main focus is that, how principles and ethics work in agile development

environment and how agile methodology is considered as different and new from the current classical methodology and section 5 focuses on the issues of applying methodology in the organization and how it helps to develop the software quickly. In section 6 methods and paths are given to improve organizations under agile development environment.

2. CLASSICAL DEVELOPMENT APPROACHES

In this section those approaches of software development life cycle are considered as a classical approaches in which plan of execution has already been made before the starting of the project. These approaches are generally known as predictive and predefined approaches. Waterfall, spiral evolutionary and prototype are the main example of these types of development cycles. In these approaches customer has to give all the requirements before the starting of the project or a contract has been made at the starting of the project to seize the requirements. In this type of development approaches, requirements are generally taken from Subject matter experts and business users and captured by an experienced person from the developer side known as Business Analyst who do query and clear the doubts regarding the project with the help of questionnaires, interview sheets, modeling tools etc and finally requirements are gathered in requirement specification templates according to organizational framework.

But at the starting, customers give too much requirements than they needed, because customers feared that if the requirements are not given at the starting then their demands will not be entertained at the time of development. There are many reasons for the customers to think like that because customers know that if management has changed then their requirements will not be added and they think that some features can be used after sometimes and this is one of the biggest reason why documentation is treated as a product which takes too much effort and time to be completed. Documentation is done from the beginning till end. Lots of money is spent to make the documentation. With the help of documentation developers also want to ensure that what the customer wants exactly in his software and accordingly a fully describes plan is made. Plan-driven methods work best when developers can determine the requirements in advance and when the requirements remain relatively stable, with change rates on the order of one percent per month [2]. One solution fit for all, is a mind set in these development approaches. Finally customers find a product in which many features will never be used and some features will be used rarely and there will be a small part which will be exactly needed. Because these approaches take too much time to implement any project, during that time customer requirement may be changed or new technology has already been evolved.

Testing of software is an essential part to ensure that software is functionally and structurally correct. For this reason there is separate phase of testing defined after the development phase in the classical development life cycle. However verification part of the requirements is done at the starting but there are also a chance of errors and faults which cannot be captured by

verification process. So Validation testing after the development phase is also needed. So success of a project which is approached in this way relies on knowing all of the requirements before development begin and means that implementing change during the development lifecycle can be somewhat problematic. However, it also makes it easier to determine the costs of the project, set a schedule and allocate resources accordingly [3]. Figure1 depicts the theme used in every traditional approach.

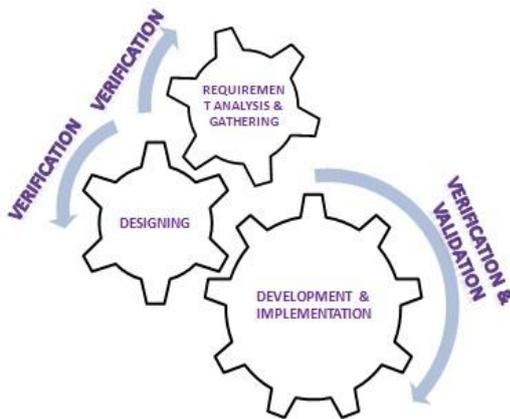


Fig 1: Theme of classical development approaches

3. AGILE DEVELOPMENT APPROACHES

Agile Methodology which is used to convert the classical development models to a quickly and easily moveable process. Agile is known as an adaptive and empirical process which works on following principles (Agile Manifesto) [4]:

- Individual and interaction over process and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

Agile methodology welcomes changing in requirement during the time of development. The main focus of agilest is to continuously deliver the software or it can be stated that Agile Methodology promotes sustainable development. The main requirement of this methodology is technical excellence and good design. In this approach, information is conveyed face to face and requirements are captured quickly because at the time of development customer is also involved.

Those customers who are involved in development process are known as active stakeholders and give their requirement directly to the developers and set the priority according to the need. Then developers estimate the efforts of making the particular phases. White board, video meeting modeling tools and face to face communication are the way of interaction between developers and stakeholders. But personnel resources are not unlimited, there is also some part of the software that may go untested, or may be verified to a lesser degree. The basic, and most difficult, aspect of system verification is to decide what must be tested, and what can be left untested, or partially tested [5].

The main part of agile methodology is that, it does not work on certain rules but it works on philosophies and principles. But if you are satisfied with how your project is managed, then agile is not for you. Team work is an important factor for making software quickly and correctly. Every project is

treated as a unique one and a sweet spot is necessary so that every person related to the project may work together in effective way. There is no specific rule of implementation in agile methodology; working is decided according to the situation of the organization and market as well. Work is done in incremental way. In this increment approach, the duration of implementation should not be more than one month and scope is decided according to that. Agile approaches look alike evolutionary and iterative models of classical development but it is very much different from the classical ones. The main difference is the duration of the increment. In classical one, the duration of increment are two to three months but in agile it is generally set to one month at max. There are currently six methods that are identified as agile development methods, which are Crystal methodologies, dynamic software development method, feature-driven development, lean software development, scrum, and extreme programming [6]. Figure2 depicts the theme of Agile Methodologies.



Fig 2: Theme of agile development

4. ANOMALY

In this section the main focus is that, why agile methodology is considered as a different and new approach from the classical methodology. In classical methodology development team size is large and work locally as well as remotely but in agile methodology team is relatively lesser (6-8 person) and try to work locally. The main deviation from the classical one is the way of communication. In classical methodology documents are the only way of communication among the team, since team size is larger without documents it will not be easy to track the work but it takes time to resolve any problem with the help of documents and the backbones of any development cycle, data administrators, architects and developers interact with documents only and their technical terms of working are different from each other which creates problem to understand the requirement correctly but in agile methodology, face to face communication is followed so that it may take less time to resolve any problem in the small team. During development in the classical methodology tracking of the progress is not feasible because the development cycle of this methodology only allows the full development of software then deliver hence cost is given by the customers based on the contract of whole software but the work in agile

methodology can be observed with the help of backlog charts and burn down charts because software is developed in small increments and priorities of delivery is also set by the customer. Cost is given or set according to the functionality delivered to the customer that is known as pay per use.

There are same practices are followed to develop any software in classical methodologies, in which testing efforts are estimated separately. In it validation and acceptance testing are the main focuses and in agile methodology plan of development and execution is based on the customer's need and the situations in which acceptance testing is done during development. Hence less effort are needed to test after the development.

5. DISCUSSION

Now it is cleared that what were the difficulties in the current methodology (Classical methodologies) and how these difficulties can be overcome with the help of agile methodologies. In agile there is no guidelines to follow, but have certain principles to follow strictly. We know if we use the Agile Methodology, we can give the better result and delivery of the working software would be on time. But the problem starts from here that how would we implement the agile technique within the organization. Although agile methodologies triumph traditional methodologies in many aspects, there exist several difficulties in putting it into practice. One among these is How to convert project management office to agile office in the organization [7].

To explore the data for the related project from the whole organizational data is a big task, when new methodology has to be implemented in the organization. In an organization, it is hard to convince the stakeholders to implement agile methodology. Quality of particular software is main aspect when there are no specific guidelines to follow. At the starting there is need to know that how the project would be mapped to the organizational standards.

To set the releases is also a point of discussion, it should be decided when to seize the customer's requirement for the particular increments. In agile methodology documents are not treated as a product so we need to specify the areas where documentation is necessary. For a new methodology training is main part and at which phases training is necessary should be decide.

Dedicated developers, experienced developers, small collocated team, and automated regression test, easy access to users, short increment and frequent delivery to real users are the six sweet spots to work better in agile approach [8]. In agile methodology after development and testing phases all the members in the team, share their problems and needs and try to find better way to execute next time. One person will note down the important aspects of the meeting, these aspects will help to adopt the methodology in an organization. One of the main points of discussion in agile methodology is that it has no certain steps to develop software. Every organization has their way of accepting and conducting this agile methodology. Generally it works for small teams because of good communication and understanding purpose. It is also highly possible for stakeholders to maximize their business return on investment by practicing agile methods in system development. The direction of the development is always readily changeable and the cost of change is low, as the stakeholders are given the opportunity to revise the business factors at the beginning of iteration to include additional features into the system according to business ROI [9]. Figure3 depicts the primary requirements from any methodology.

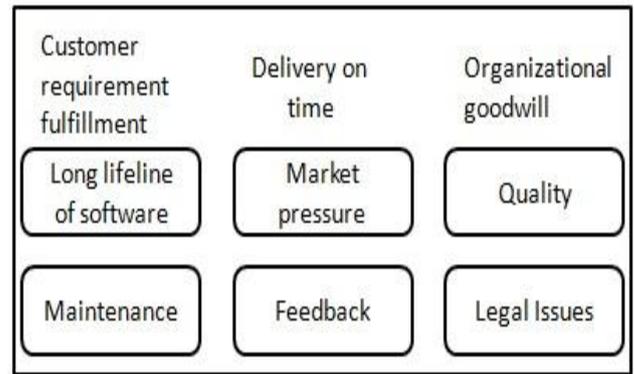


Fig 3: Demand from methodology

6. IMPROVEMENT

To start implementation under agile methodology (AM) within the organization, synchronization between project management office and agile office is necessary because it takes time to be implemented within the whole organization.

To improve the agile methodology a method known as The Agile Data Method [10] can be very useful in which database administration, application developer, enterprise administrator, and enterprise architect work together to make the data available to implement the software under agile methodology.

The belief on the Agile Methodology and the effective communication within the organization is the only way to convince stakeholders. In requirement issues we communicate and collaborate with the customers and accordingly acceptance tests are made by the testers and the developer. Three types of cases are used which are use cases, loose cases and choose case and for requirement NORMAP [11] (non functional requirement modeling for agile process) technology is used for relating non functional and functional requirements. Training is necessary in the beginning, for setting up the methodology as well as time to time training is necessary for effectiveness. Grooming the artifacts from the documents is also helpful to improve the methodology in an organization and few points should also be noted for developers to practice agile development.

First of all, they should make incremental change to the requirement, project plan system, and the resulting artifacts to enable agility They should also strive for feedback to ensure the project meets the needs of all participants and stakeholders. In addition, only those tasks that add value to business processes supported by the system should be performed and lastly, processes and artifacts that do not add enduring value to the working software system should be discarded [12].

7. CONCLUSION

If all the stakeholders are able to express all their requirements at the starting with no market competition and they have enough time to wait with enough money to spend, then classical methodologies are the best for them. So classical methodology helps to provide quality in context of the customers and organization context as well.

But in unpredictable and requirement changing environment a methodology is needed which can maintain the speed and overcome the issues of unpredictability. A methodology which will be beneficial for both developers and customers side in term of cost. Market deadlines should be met in order

to remain in the competition. So all these aspect are fulfilled by agile methodology but we need to implement it according to requirements and the situations, so that it may be able to meet the requirement of the customers and the market. But customers will also have to take the responsibility to make the right product during all the phases of development.

8. REFERENCES

- [1] Jack Milunsky, How Agile methods resolve chaos and unpredictability, January 2009.
- [2] B.Boehm, Get Ready for Agile Methods, with Care, IEEE Computer, vol. 35, no. 1, pp. 64-69, 2002.
- [3] IBM, Rational Unified Process: Best practice for software development teams, <http://www.ibm.com/developerworks/rational/library/253.html>, 2003 .
- [4] Agile Manifesto, Manifesto for Agile Software Development, <http://agilemanifesto.org>, December 2007
- [5] M. Vouk and A. T. Rivers, Construction of Reliable Software in Resource-Constrained Environments, W.R.Blischke and D.N.P Murthy, Eds.Hoboken, NJ; Wiley-Interscience, JohnWiley and sons, pp 205-231, 2003.
- [6] Dyba, Tore, Empirical studies of agile software development: A systematic review, 24 January 2008.
- [7] Vijayarathy, Leo R, Agile Software Development: A survey of early adopters, Journal of Information Technology.
- [8] Mark C. Paulk, Agile Methodologies and Process Discipline, Carnegie Mellon University.
- [9] Rico, David F, What is The ROI of Agile vs. Traditional Methods, 2008.
- [10] Scott W. Ambler, The Agile Data Method, <http://agiledata.org>.
- [11] Weam M. Farid, Frank J. Mitropoulos, NORMATIC: A Visual Tool for Modeling Non-Functional Requirements in Agile Process, 2012.
- [12] Carayannis, Agile Project Management for IT project, Greenwood Press, Quorum Books,2002.