

Agile Survey : Indian Organizations

Uma Kumari

Jagannath University, Jaipur

Abstract

The software market is becoming more dynamic which can be seen in frequently changing customer needs. Software companies need to be able to quickly respond to these changes. For software companies this means that they have to become agile with the objective of developing features with very short lead-time and of high quality. In agile the goal is to understand the effect of migrating from a plan-driven to an agile development approach. Agile methods have proven to be beneficial in small projects, and there has also been growing interest in using these methods in large projects. This paper analyzes what agility and agile development are and their study in Indian organizations.

1. Introduction

During the past forty years, new software development approaches were introduced to fit the new cultures of the software development companies [1]. Most software companies nowadays aim to produce valuable software in short time period with minimal costs, and within unstable, changing environments. Agile methodologies were thus introduced to meet the new requirements of the software development companies [1].

1.1 Key features of agile methodology

The key features are as follows

- *Adaption*
The teams of developers are self-organized based on the daily meetings. Developers and customers self-organize at the end of every increment to guide the project and create the greatest value.
- *Emergence*
The architecture, team structure, and requirements emerge during the course of the project rather than being determined at its outset.
- *Modularity*
Modules are designed to work for a specific task. Agile methods break the whole task into small parts. Every part is solved individually

with a very little planning. These tasks do not directly involve future planning. This means the whole concentration is on present module and its requirement at that time. The future scope of that module and what other modules will contain is not of concern at this stage. They are just containing one part and others are included at latter stages.

- *Incremental development*
The life cycle of project is divided into small iterations. Each iteration last from two to four weeks. Each iteration undergoes the whole process of software development life cycle. This includes requirement analysis, planning, design, implementation, unit testing and acceptance testing. After acceptance testing of software one iteration is over. Once it is released it is verified by the stakeholders and corrections are made. In incremental development, there are small software released with rapid cycles. So the whole application is built in small steps.
- *Team composition*
Team consists of experienced developers. It is because there is no as such documentation for the project. Everything depends on how the stakeholder gives the definition of a particular module. The developer should understand the view point of the stakeholder. This understanding requires a lot of experience, so almost negligible place for fresh candidates.
- *Meetings*
Meetings provide an internal status of the project.
- *People oriented*
In agile methodology, instead of the processes and technology, people are favored. Whatever method of agile is followed, each agile team consists of one customer representative. Customer representative is the one who is involved as an end user of the project. This person is appointed by the stakeholders (customer representative who will give money). He acts on their behalf and can make any commitment required to carry on with the development of the project. He is also responsible for answering for questions raised

by the developer during any mid –iterations i.e. during development. At the end of the iteration this customer representative together with the stakeholders, review the progress of the project and re-evaluate the priorities of the modules. It is done in order to satisfy the company's current needs and goals.

- *Contrast with cowboy coding*

Cowboy coding is the absence of a defined method: team members do whatever they feel is right. Agile development's frequent re-evaluation of plans, emphasis on face-to-face communication, and relatively sparse use of documents sometimes causes people to confuse it with cowboy coding. Agile teams, however, do follow defined (and often very disciplined and rigorous) processes. As with all methodologies, the skill and experience of the users define the degree of success and/or abuse of such activity. The more rigid controls systematically embedded within a process offer stronger levels of accountability of the users. The degradation of well-intended procedures can lead to activities often categorized as cowboy coding.

- *Cooperative*

In agile, the customers and the developers work together with close communication. Most agile implementation used to have a daily meeting among the team members. Together with the developers, the customer representative or one of the stakeholders also attends the meeting. This meeting helps everyone to know the actual progress of the project. In this session each team that is handling individual modules is required to report what they have done last day and what they intend to do today. Also they discuss the difficulty they are facing. This face to face communication helps them to get the solution to their problems.

- *Collaborative and communicating working style*

When the whole of the team working on a single project is in same location, instead of written documents, agile method to a great extent emphasize on face to face communication between the team members. So, the cost of moving the information between the people is reduced. This communication is facilitated through a single open office of agile team known as bullpen. The team size is typically small, ranging from 5 to 9 people per team. This simplifies the communication and the cooperation among the

team members can be increased. It is not necessary that one team work on one module, multiple teams can work on one module. So, more development efforts can be there if they are needed. A prioritization across teams is done in order to achieve things in right manner and at right time. When team works at different locations, then contact is maintained through video conferencing, e-mails etc.

- *Lightness*

Lightness signifies the ability to adapt to the changes. The numbers of processes that are followed during the development of project are very less. Also, the documentation part of the project takes a back seat. So, it is not necessary to provide a heavy documentation to the project at the beginning of the development itself. The documentation regarding the project is made available to the developers as and when required. Moreover the documentation are usually replaced the conversation between the people. The product that is released after iteration contains many bugs but after many iterations the product has full functionalities.

In past main methodology was heavyweight methodology which consists of heavy documentation, comprehensive planning and extensive design. New methodologies such as lightweight methodologies emphasis is on working software rather than documentation and responding to change over following a specific plan

Due to all these problems a study of lot of agile methodologies has been worked out and their analysis has been done.

2. Analysis

To identify which methodologies software practitioners use in different sized companies in India for different sizes of project development, three different methods were followed. Information regarding opinions on agile methodology and heavyweight methodology was collected using questionnaire, web based survey and interviews.

2.1 Questionnaire

It was divided into two sections- general questions and software based question

2.2 Web based survey

Survey studies a phenomenon for a population by surveying a sample that is representative for that population. Online questionnaires are preferred as they allow reaching a larger sample and are less time consuming from the researchers point of view, and are thus more efficient than interviews with regard to the number of data points that could be collected. Having collected the data of the sample statistical inference is used to draw conclusions for the overall population. Web based survey was offered over a period of three months (June, July, August) in the year 2013. In web based survey respondents were asked a total of 37 questions[2].

2.3 Interview analysis

Interviews are conversations guided by an interview protocol and are considered one of the most important resources for data when conducting case studies. The interview protocol can vary in the degree of structure, ranging from very structured (interviewee has to stick with research questions) over semi-structured (interviewee has a guide, but can change the course of the interview to follow interesting directions) to unstructured (rough definitions of topics to be covered). Very structured interviews were followed. Unstructured interviews were not considered as interviews. The data was primarily collected from a total of 11 interviews conducted at different companies in India.

3. Agile v/s different traditional models

Here comparison of agile with different models has been specified.

3.1 Agile v/s Waterfall model

Adoption of waterfall has helped to drive down the failure rate of software development projects, but even with rigorous project management and processes, a full 70 percent of software projects using this methodology fail to meet their objectives [3]. Organizations tried to cut the failure rate by insisting on more detail in the requirements and design phases. This process of requiring extensive, even exhaustive, documentation culminated in 1988 with the publication of the Department of Defense Standard for software development[3].

Why are people abandoning waterfall and moving to agile?[4]

- *Agile is Adaptive*
- *WYSIWYG (What You See Is What You Get) Development*
- *Shorter Time to Market*
- *Greater Employee Satisfaction*
- *Higher Quality*
- *Higher ROI*

It's sometimes hard to understand why waterfall is still practiced. But the truth is, adopting agile takes a paradigm shift in thinking that is not easy for individuals, much less organizations, can make it. It also takes experience not only in practicing agile, but also in managing organizational change.

3.2 Agile v/s Iterative enhancement model

Iterative model has the same phases as the waterfall model, but with some restrictions. The phases occur in the same order as in waterfall model, but these may be conducted in several cycles, with each release providing additional functionality [5].

Some of the major differences between iterative enhancement model and agile are

- Each agile team consists of one customer representative. Customer representative is the one who is involved as an end user of the project or he can be product distributor or he can be the representative of the company who will use product. But iterative enhancement model does not consist of customer representative.
- Agile does requires less documentation but iterative method require a lot of documentation.
- Agile methodology requirements can change till the end of the project but in iterative enhancement method once requirements are made they do not change.
- Agile is based on iterative and incremental method both but an iterative method is only based on iterations.
- In agile at the end of each iteration, the iteration does not consist of all functionalities but consists some functionality but they are bug free. In iterative enhancement model all functionalities are provided and if some bug remains then they are removed in further iterations.
- Iterative is predictive whereas agile is adaptive.

- Time required in agile is in weeks whereas in iterative enhancement it is in months.

3.3 Agile v/s Spiral model

Spiral model is one of the popular process model used by industry. It is based on evolutionary approach which couples the iterative nature of prototyping with the controlled and systematic aspects of linear sequential model. The project is reviewed and the next phase of the spiral is planned [6]. It was proposed by Bhoem in 1988 and is a popular model used for large size projects.

Key difference between agile and spiral method are:

- Agile methodology is for smaller projects which consist of smaller teams but spiral model is for larger projects involving larger teams.
- Agile consists of experienced team members but spiral can consist of fresher's as well.
- There is less documentation in agile but spiral consists of heavy documentation.

3.4 Agile v/s V-Shape model

V-shape model was developed to relate the analysis and design activities with the testing activities and thus focuses on verification and validation activities of the product. Main differences between V-shape model and agile methodology are:

- V-shape model does not support iterations of phases and dynamic changes in requirements throughout the lifecycle but agile does support.
- It does not take into account risk analysis but agile does support this.
- Testing cycle time of agile is relatively short compared to V-model, because testing is done parallel to development in agile.
- Agile model delivers a working version of the product very early compared to V-model. As more features are delivered incrementally, customer can realize some of the benefits early on.
- Agile is a proactive model (due to its very short cycles) compared to the much more reactive V-model [7].
- V-model is very rigid and relatively less flexible than agile model.
- V-model does not provide a clear path for problems found during testing phases.

3.5 Agile v/s Prototyping model

A prototype is made first and based on it final product is developed. A prototype is a model or a program which is not based on strict planning, but is an early approximation of the final product or software system. A prototype acts as a sample to test the process. From this sample we learn and try to build a better final product. This prototype may or may not be completely different from the final system we are trying to develop. This type of System Development Method (SDM) is employed when it is very difficult to obtain exact requirements from the customer. While making the model, user keeps giving feedbacks from time to time and based on it, a prototype is made. Completely built sample model is shown to user and based on his feedback; the SRS (System Requirements Specifications) document is prepared [8]. There are some differences between agile and prototyping model such as

- Prototyping is usually done at the cost of the developer. So it should be done using minimal resources. It can be done using Rapid Application Development (RAD) tools. Sometimes the start-up cost of building the development team, focused on making prototype, is high [8].
- It is sometimes called as "Throw-away" prototype.
- Prototyping is a slow process [8].

3.6 Agile v/s Rapid Application Development (RAD) model

Rapid Application Development (RAD) model is incremental software development process model that emphasizes short development cycles. The Rapid Application Development model was proposed by IBM in 1980's and later on was introduced to software community by James Martin through his book Rapid Application Development. There are some differences between RAD and agile such as:

- One of the agile methodologies DSDM uses RAD model. DSDM grew to provide an industry standard project delivery framework for what was referred to as Rapid Application Development (RAD).
- RAD was extremely popular in the early 1990's, the RAD approach to software delivery evolved in a fairly unstructured manner. As a result, the DSDM Consortium was created.

4. Detailed findings

- *Organization size*
As far as organization size is considered, most of the organizations were of medium size targeting around 40.5% and 26% were of small size and 33.33% of large size. Most of people who currently use agile are in Information technology field.
- *Adaptation*
From an adoption standpoint it was observed that people are using agile in India or they want to adopt agile methodology in near future. Most of the persons have average knowledge of agile methodologies and average knowledge of traditional methods.
- *ASD methodologies*
There are several ASD methodologies available today. Scrum is most liked software development technology that is used by maximum users for small scale software development. The technology changes as the size of software to be developed increases.
- *Most preferred heavyweight methodology*
When it comes to use of heavyweight methodology then waterfall model is the most well known model used by individuals for small scale projects. If we concern medium and large sized projects then spiral model is mostly used.
- *Improvement in technology when agile is used*
As far as questions related to quality, productivity, business stakeholder's satisfaction and cost of system development is concerned, maximum responded that cost of system development reduce on using agile. In case of productivity it becomes better. Maximum responded that their quality was better on using agile. When it comes to business stakeholder's satisfaction then business stakeholders are more satisfied on using agile as compared to heavyweight methodologies.
- *Problems faced while practicing agile*
According to finding the most common problem while practicing agile methodologies for small and medium scale projects is lack of skilled people, who can follow agile methodologies but for large scale projects it is project size or complexity.
- *Agile or heavyweight which technology is more suitable*
It has been found that agile is more suitable technology for small scale and medium scale

projects but heavyweight methodology is more suitable for large scale projects. This happens because small scale projects can be managed easily but large scale projects management becomes tough because there are lot of teams and managing large teams is a difficult task without control of management.

- *Which aspect of agile methodology appeal most*

This question tells as that why people are moving towards agile. Maximum responded that changes can be given by stakeholders at any stage in case of agile. But in traditional methods a specific plan is followed. So maximum responded that "Respond to change verses following a plan" appeal them most to adopt agile for different type of software development.

- *Which aspect of methodology do people dislike the most*

It was found that people dislike "low planning" in case of agile most but in case of heavyweight methods they dislike "lack of project structure" strategy.

References

- [1] Malik Hanief, Siew Hock Ow, Review of Agile methodologies in software development, International journal of research and reviews in applied sciences, Vol I(issue I),Oct 2009
- [2] <http://www.surveyact.com/s/Szjjuw1AZ5DJuii>
- [3] <http://www.serena.com/docs/repository/solutions/intro-to-agile-devel.pdf>
- [4] <http://www.cedarpointconsulting.com/deliver/articles/departing-waterfall-next-stop-agile>
- [5] <http://www.projectsjugaad.com/Software-Engineering-Projects/Software-Life-Cycle-Models.html>
- [6] Ian Sommerville, Software Engineering (7th Edition), Pearson Education Ltd., 2001
- [7] <http://vspages.com/agile-vs-v-methodologies-model-7142/>
- [8] <http://www.ianswer4u.com/2011/11/prototype-model-advantages-and.html#axzz2nFYxMSJc>