ISSN 2394-187(Online)
ISSN 2394-7179 (Print)

Empirical Studies of Agile Software Process Model

V. Meenakshi

Department of Computer Applications, S.A. Engineering College
meenakshisaras18@gmail.com

Abstract—Agile software is upcoming one and most of the Software Organizations can able to gain high profit and successful result by using agile software. The most important methods of agile software development are namely Scrum, Extreme Programming; Feature Driven Development etc. are mainly used to develop software. This paper presents the analysis and techniques of Dynamic Systems Development (DSD).And also presents the increased productivity in the world of business field by using dynamic systems development method as well as strengths and weakness .

Keywords— *Agile, Scrum, Dynamic Systems Development Method, DSDM functionalities, DSDM Techniques, Facilitated Workshops,*

1. Introduction

Agile is a software used in the business field. According to dictionary, the word agile has two meanings, mentally quick and physically quick. Agile is nothing but it responds to things which happen accidently. An agile process is main about flexibility. This presents the increased productivity in the business field by using dynamic systems development method. Dynamic Systems Development is used in the process of developing a project. Mainly, DSDM focuses on information systems project. DSDM is a straight forward framework based on best principles for implementing a new project structure. It is simple, extendible, but not calming to be the solution to all kinds of projects. Results of development are directly and promptly visible. Since the users are actively involved in the development of the system, they are more likely to embrace it and take it on. Basic functionality is delivered quickly, with more functionality being delivered at regular intervals. Eliminates bureaucracy and breaks down the communication barrier between interested parties. Because of constant feedback from the users, the system being developed is more likely to meet the need it was commissioned for.

2. Dynamic Systems Development Method

The Dynamic Systems Development Method (DSDM) is framework of agile software project delivery and it is primarily used as a software development method as shown in Fig. 1. It is a framework containing the current knowledge about project management. The DSDM framework can be implemented for agile and traditional development processes.

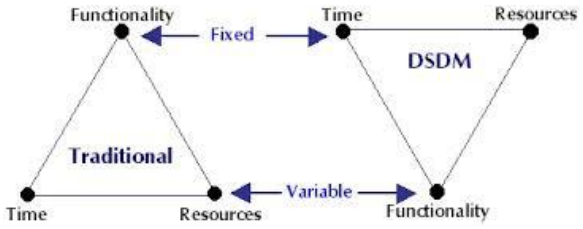

Fig 1: Functionality of dsdm

3. Techniques of DSDM

DSDM (Dynamic Systems Development Method)is a framework that delivers the right solution at the right time. DSDM has been for many years the leading, proven agile approach, providing the agility and flexibility demanded by organizations today.

The approach is the high point of practitioner's experience drawn from a wide range of public and private sector projects over a span of years as shown in Figure 2, DSDM deals with the use of several proven techniques, including:

- Facilitated Workshops
- Modeling and Iterative Development
- Moscow Prioritization
- Time boxing

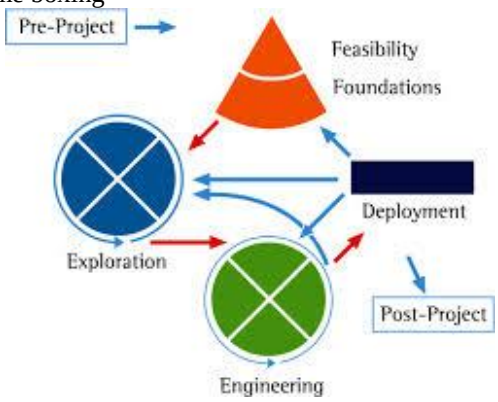

Fig. 2: Life-cycle of a DADM

ISSN 2394-187(Online)
ISSN 2394-7179 (Print)

3.1 Facilitated workshops

Facilitated Workshops are a specialized for a type of meeting, with a clear product, a set of people who are chosen and allow producing the product and a facilitator which is of an independent person can qualify the effective achievement of the objective.

Facilitated Workshops are a process of a neutral facilitator is nothing but a facilitator is someone who helps a group of people who understand their common objectives and assists them to plan how to achieve these objectives in doing so, the facilitator remains "neutral" meaning he/she does not take a particular position in a discussion, with no support in the outcome of the workshop, qualify a group to work together in order to achieve an goal can be of solving a problem, building a plan, making a decision or more requirements as shown in Fig. 3.

3.1.1 Benefits

There are direct and indirect benefits to a project as shown in Fig. 4 by using Facilitated Workshops.

3.1.1.1 Rapid, high quality decision-making

Facilitated Workshops can reduce the progress time required to achieve the identification, agreement and sign-off requirements which is nothing but the objectives. Because all relevant to a person with an interest or concern in something, especially business are present at the same time and able to communicate with each other then they will have much more confidence in the result.

Fig. 3: Areas of facilitation

3.1.1.2 Greater buy-in from all stakeholders

Facilitated Workshops leads to participants involved in the end results having an opportunity to participate in, and contribute to, both decisions that are made. This builds and helps to maintain eagerness throughout the project.

3.1.1.3 Building team spirit

Facilitated Workshops are a controlled way of building link across the community. The output of the workshop benefits from the participant's ideas and gaining a better understand ability of other viewpoints. A successful Workshop depends on high levels of synergy being achieved and it is a major part of the Workshop Facilitator role to ensure that this happens.

3.1.1.4 Ideas building consensus

The Facilitated Workshop provides an chance or opportunity for participants to discuss the information about the project that are the major issues, problems and the important decisions. If business procedures and practices are reviewed, participants can gain a greater understanding of the inputs and implications of their work. This can lead to improved efficiencies, led by the participants themselves, giving greater buy-in and commitment and therefore a greater chance of successful implementation.

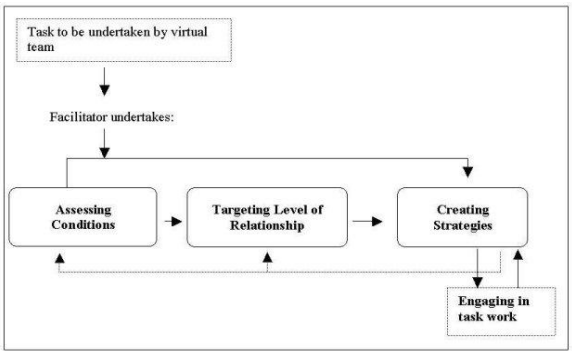

Fig. 4: Functions of facilitator

3.1.1.4 Clarification of issues

Workshops help to minimize the quality of being open to more than one interpretation and misunderstanding. The participants can learn about it and the model ideas, which simplify the review and sign-off the workshop deliverables.

ISSN 2394-187(Online)
ISSN 2394-7179 (Print)

3.2 Modeling and iterative development

An iterative life cycle model is not a specification of requirements. Instead, of the requirements the development begins with the implementation of the software as shown in Fig. 5, and then it can be reviewed in order to identify the requirements.

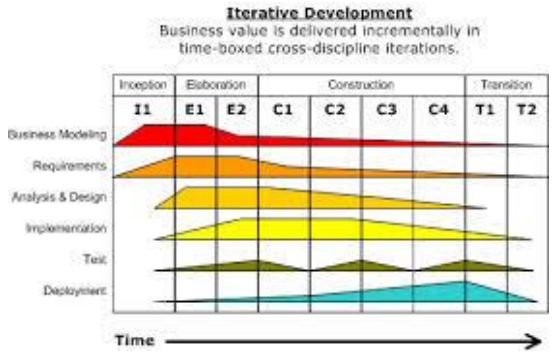

Fig. 5: Iterative development

As well as the process is repeated for introducing a new version of a software of each cycle of a model. Iterative is an adjective means repetitious. The business value is delivered incrementally in time-boxed cross-discipline iterations namely,

3.2.1 Inception phase

Inception is the smallest phase in the project, and ideally it should be quite short. If the Inception Phase is long then it may be an indication of excessive up-front specification, which is contrary to the spirit of the Unified Process.

3.2.2 Elaboration phase

During the Elaboration phase the project team is expected to capture a healthy majority of the system requirements.

3.2.3 Construction phase

Construction is the largest phase in the project. In this phase the remainder of the system is built on the foundation laid in Elaboration. System features are implemented in a series of short, time boxed iterations. Each iteration results in an executable release of the software.

3.2.4 Transition phase

The final project phase is Transition. In this phase the system is deployed to the target users. Feedback received from an initial release may result in further refinements to be incorporated over the course of several Transition phase iterations. It also includes system conversions and user training.

3.3 Moscow prioritization

In an Atern project where the time is fixed, so that it will be easy to understand the importance of making the progress and keeping the deadlines. Prioritization is a technique which can be applied to products, requirements, user stories and tests. Moscow is a technique which helps to understand priorities as shown in (Fig. 6). The letters stands for,

- Must Have
- Should Have
- Could Have
- Won't have this time

3.3.1 The Moscow Rules

There are different possible rules namely,

3.3.1.1 Must have

It provides a Minimum Usable Subset(MUS) are requirements in which the delivery of the project is guarantee. If there is some way round it, even if it is a manual workaround, then it will be a Should Have or a Could Have requirement.

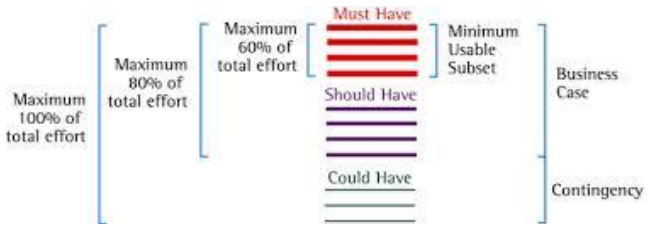

Fig. 6: Rules of moscow

3.3.1.2 Should have

Should Have is important, but it is not that much essential. It may be painful to leave out, but the solution is still possible. As well as it may need some kind of workaround, (e.g. management of expectations, some inefficiency, an existing solution, paperwork, etc).

3.3.1.3 Could have

It is desirable but it is not important. It is less important, compared with a Should Have it has less impact, if it is left out.

ISSN 2394-187(Online)
ISSN 2394-7179 (Print)

3.3.1.4 Won't have this time

These are requirements which the project team has agreed it will not deliver. They are recorded in the Prioritized requirements list where they help clarify the scope of the project and to avoid being reintroduced 'via the back door' at a later date.

3.4 Time boxing

Time Boxing is just setting short time periods for the development work. It is a process of controlling low-level products in an iterative fashion, with several important points to qualify the quality of those products and the efficiency of the delivery process. By managing on-time delivery at the lowest level, on-time delivery at the higher levels can be assured. Initial Moscow prioritization of the work within the Time box and continual re-assessment of what can be achieved in the agreed timeframe ensures that time boxes finish on time, every time.

3.4.1 Controlling a Time box

Every Time box can be considered as beginning with a Kick-off and ending with a Close-out meeting. The Time box itself divided into three main stages or Iterations – Investigation, Refinement, and Consolidation – each reflecting a pass through the Iterative Development cycle.

3.4.1.1 Time box Kick-off

The aim of the Kick-off is to ensure that it is still feasible to deliver in the Time box what was expected at the Foundation stage and to re-plan accordingly if not, Agree the acceptance criteria for each product to be delivered. Remember that commitment to delivery is based on a pre-agreed and fixed minimum resource levels as shown in figure 7.

3.4.1.2 Investigation Iteration

The aim of Investigation is to provide a firm foundation for the work to be carried out during Refinement. For Time boxes focused on Exploration activity, this will entail the Solution Developers and Business Ambassadors jointly investigating the detail of requirements and agreeing how these requirements will be met as part of the Evolving Solution.

3.4.1.3 Refinement Iteration

The aim of Refinement is to complete as much of the development work as possible including testing the deliverable(s). Development is carried out iteratively, with the primary objective being to meet the detailed acceptance criteria previously agreed. Refinement should start off with a quick and informal planning session, focused on determining which members of the team will be working on what products, in what order.

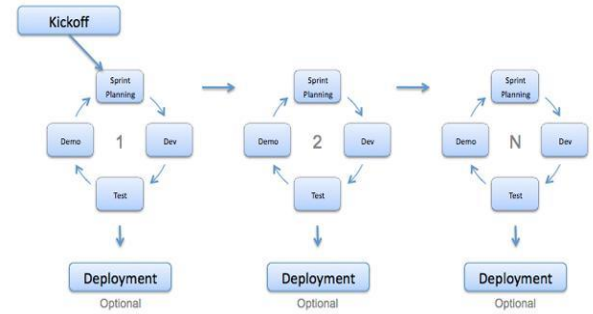

Fig. 7: Foundation stage of kick-off

3.4.1.4 Consolidation Iteration

During Consolidation, the actions agreed at the Refinement review are carried out, together with any work required to satisfy organizational or project standards.

3.4.1.5 Close-out

The primary aim of the Close-out session is to record formal sign-off or acceptance of all the products delivered by the Time box. An important secondary aim is to determine what is to be done about work that was initially part of the Time box but was not completed. Such work may be considered for the next Time box as shown in Fig. 8, scheduled for some point further into the future or even dropped from the increment or project completely.

Fig. 8: Steps of Time box

4. Conclusion

This paper will focus on what was behind the agile movement, how these methods are distinct from traditional approaches, and what are difficulties in implementation of these approaches. In addition, the paper reviews the agile methodologies. The proposed roles, skills, and ability of Dynamic Systems Development Method can be used to increase the productivity in the business field. It can be very useful in the future, while it is used in the projects to increase the productivity in the business world that is

ISSN 2394-187(Online)
ISSN 2394-7179 (Print)

considered within the current scope of objectives without the need for a formal change control process that reaches beyond the team.

References

[1] C. Larman and V. Basili, "Iterative and Incremental Development: A Brief History",• IEEE Computer, vol.36(6):47-56.
[2] Schwaber Ken, Beedle Mike. *Agile software development with Scrum*, 1st Ed, Prentice Hall, 2002.
[3] Rising, L.,Janoff, N.S.,"The Scrum software development for process small teams",Software, IEEE,Volume:17,Issue:4,pp.26-32.
[4] Jeff Sutherland's Scrum Handbook (September, 2012) [online]. Available: jeffsutherland.com/scrumhandbook.pdf
[5] Mikio Aoyama, "Agile Software Process and Its Experience", Proceedings of the 1998 International Conference on SoftwareEngineering,p.3-12.
[6] IlanOshri, Julia Kotlarsky, and Leslie Willcocks, "building social ties for global teamwork ", communications of the acm, vol.51,no.4,(2008),pp.78-81.
[7] Fatma Cemile Serce, Ferda Nur Alpaslan Kathleen Swigger, "Exploring Collaboration Patterns among Global Software Development Teams", Fourth IEEE International Conference on Global SoftwareEngineering,(2009).pp.61-70.
[8] Felix Kobler, Marilyn Tremaine, Jan Marco Leimeister, Helmut Krcmar, "non-optimized temporal structures as a failure factor in virtual teams", Technische University Munchen, Rutgers University &UniversityKassel,(2007).
[9] Beck, Kent; et al. (2001). "Manifesto for Agile Software Development". Agile Alliance. Retrieved 14 June 2010.
[10] Beck, Kent; et al. (2001). "Principles behind the Agile Manifesto". Agile Alliance. Archived from the original on14June2010.Retrieved6June2010.

V. Meenakshi is holding under graduation degree in B.Sc computer science from Bhaktavatsalam Memorial college for women and pursuing post graduation in computer applications from S.A. Engineering College. This paper is a part of curriculum covered under in (13MC022) "Technical Seminar And Report Writing".