IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 3, Issue 9, September 2016

DSDM and ASD Agile Methodologies

Anupama Kaushik

Assistant Professor, Department of IT, Maharaja Surajmal Institute of Technology, New Delhi, India

Abstract: Now-a-days software development firms are adopting agile methodologies. Agile represents a group of software engineering methodologies which promise to deliver increased productivity, quality and project success rate overall in software development projects. The outline of Agile Methodologies was laid down by the Agile Manifesto, published by a group of software practitioners. This paper discusses two of the important agile methodologies Dynamic System Development Method (DSDM) and Adaptive Software Development (ASD).

Keywords: Agile Manifesto; Agile software development; Dynamic System Development Method (DSDM) and Adaptive Software Development (ASD).

I. INTRODUCTION

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. According to [1], customers found it difficult to define their needs because of the fast changing technology and the companies using them in products. New methods, now called agile methods are designed to define the changing requirements in software environments. Traditional methods refer to the older and commonly used methods like the waterfall methods [2].

Agility is the ability to sense and response to business prospects in order to stay inventive and aggressive in an unstable and rapidly shifting business environment (Highsmith, 2002), [3]. The agile approach to development is about agility of the development process, development teams and their environment (Boehm & Turner, 2004), [4]. This approach incorporates shared ideals of various stakeholders, and a philosophy of regular providing the customers with product features in short time-frames (Southwell, 2002), [5]. This frequent and regular feature delivery is achieved by team based approach (Coram & Bohner, 2005), [6]. Agile teams consist of multi-skilled individuals (Fowler, 2002), [7]. The development teams also have on-site customers with substantial domain knowledge to help them better understand the requirements (Abrahamsson, Solo, Ronkainen, & Warsta, 2002), [8]. Multiple short development cycles also enable teams to accommodate request for change and provide the opportunity to discover emerging requirements (Highsmith, 2002), [3]. The agile approach promotes micro-project plans to help determine more accurate scheduling delivery commitments (Smits, 2006), [9]. There are many agile software development methodologies. The most commonly used are extreme programming (XP),Dynamic Software Development Method (DSDM), Scrum , Crystal, Feature Driven Development (FDD)and Adaptive Software Development.

In this paper two agile methodologies Dynamic System Development Method (DSDM) and Adaptive Software Development (ASD) is described. Section II describes Dynamic System Development Method (DSDM) and section III describes Adaptive Software Development (ASD).

II. DYNAMIC SYSTEM DEVELOPMENT METHOD

The DSDM [10], Dynamic System Development Method, was developed in the United Kingdom in the mid-1990. It is a blend of, and extension to, rapid application development and Iterative development practices [11]. Martin Fowler, one of the writers of Agile Manifesto, believes, “DSDM is notable for having much of the infrastructure of more mature traditional methodologies, while following the principles of the agile methods approach” [12]. The fundamental idea behind DSDM is to fix time and resources, and then adjust the amount of functionality accordingly rather than fixing the amount of functionality in a product, and then adjusting time and resources to reach that functionality [13]. DSDM consists of five phases [10] as shown in Fig. 1:

- Feasibility Study – In this phase a decision is made whether to use DSDM or not. This is determined by judging the type of project and, organizational and people issues. In addition, two work products are produced; a feasibility report and an outline plan for development.
- Business Study – A workshop is organized to help understand the business domain of the project. The key outputs of this section are System architecture definition and an Outline prototype plan.
- Functional Model Iteration – This is the first iterative phase. This phase involves analysis, coding and prototypes. The results gained from these prototypes are used in improving the analysis models. The key output is a functional model which consists of the prototype code and analysis models.
- Design and Build Iteration – The system is mainly built in this phase. The design and functional prototypes are reviewed by the users and further development is based on the users’ comments.

Copyright to IARJSET
DOI 10.17148/IARJSET.2016.3936

- Implementation – In this final phase the system is handed over to the users. Training is also provided. User Manuals and a Project Review Report are given to users. However, the DSDM iterative and incremental nature means that maintenance can be viewed as continuing development. Instead of finishing the project in one cycle, the project can return to any of the phases, Design and Build phase, Functional Model Iteration, or even Feasibility phase so that previous steps can be refined.

DSDM approaches iterations as short time-boxed cycles of between two and six weeks.

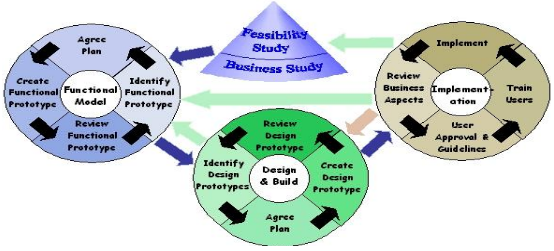

Fig. 1: DSDM Process Diagram

III. ADAPTIVE SOFTWARE DEVELOPMENT (ASD)

Adaptive Software Development (ASD) [10], developed by James A. Highsmith, offers an agile and adaptive approach to high-speed and high-change software projects [14]. It is not possible to plan successfully in a fast moving and unpredictable business environment. In ASD, the static plan-design life cycle is replaced by a dynamic speculate-collaborate-learn life cycle.

ASD focusses on three non-linear and overlapping phases as shown in Fig. 2 [10] [15]:

- Speculate - To define the project mission, make clear the realization about what is unclear.
- Collaborate – Highlights the importance of teamwork for developing high-change systems
- Learn – This phase stresses the need to admit and react to mistakes, and that requirements may well change during development.

ASD focuses more on results and their quality than the tasks or the process used for producing the results. In an unpredictable environment you need people to collaborate in a certain manner to deal with the uncertainty. Management is more about encouraging communication rather than telling people what to do, so that more creative answers are delivered. In traditional predictive environments, designs are followed the same way they were laid out, therefore learning is discouraged. Highsmith [14] points out, “In an adaptive environment, learning challenges all stakeholders - developers and their customers - to examine their assumptions and to use the results of each development cycle to adapt the next” [15]. As such learning is a continuous and important feature, one that assumes that plans and designs must change as development proceeds [15].

ASD does not have detailed principles like XP, but rather it provides a framework on how to encourage collaboration and learning within the project. ASD is not presented as a methodology for doing software projects but rather it is an approach or an attitude that must be adopted by an organization when applying agile processes [15].

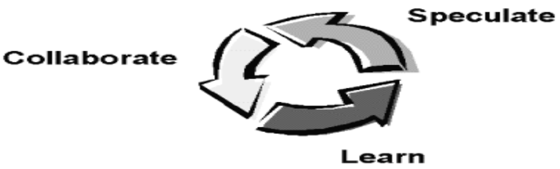

Fig. 2: ASD Lifecycle

IV. CONCLUSION

Agile methodologies are now-a-days widely used by the software development firms due to their added advantage over traditional methods. In this paper two of the most common agile methodologies Dynamic System Development Method (DSDM) and Adaptive Software Development (ASD) were discussed.

ACKNOWLEDGMENT

The author is thankful to all the authors whose references are included.

REFERENCES

[1] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Humt, A., Je®ries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thom, D.: Manifesto for agile software development. Website (2001) http://agilemanifesto.org/.

[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, April, 2011, pp. 35-45

[3] Highsmith, J. A. (2002). Agile software development ecosystems. Addison-Wesley Professional.

[4] Boehm, B., & Turner, R. (2004, May). Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on(pp. 718-719). IEEE.

[5] Southwell, K. (2002). Agile process improvement. TickIT International Journal, 3-14.

[6] Coram, M., & Bohner, S. (2005, April). The impact of agile methods on software project management. In Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the (pp. 363-370). IEEE.

[7] Boehm, B. (2002). Get ready for agile methods, with care. Computer, 35(1), 64-69.

[8] Abrahamsson, P., Solo, O., Ronkainen, J., & Warsta, J. Agile Software Development Methods. 2002. VTT technical Research Centre of Finland.

[9] Smits, H. (2006). Levels of Agile Planning: From Enterprise Product Vision to Team Stand-up. Rally Software Development Corporation Whitepaper.

[10] M.A. Awad, “A Comparision between Agile and Traditional Software Development Methodologies”.

[11] J. Stapleton, Dynamic systems development method – The method in practice. Addison Wesley 1997.

[12] M.Fowler,“TheNewMethodology,”http://www.martinfowler.com/articles/newMethodology.html

[13] Dynamic System Development Method Consortium, “DSDM Tour”, http://www.dsdm.org

[14] J. A. Highsmith, Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Addison Wesley 2000.

[15] J. Stapleton, Dynamic systems development method – The method in practice. Addison Wesley 1997.

DOI 10.17148/IARJSET.2016.3936