IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 9, September 2016

A Literature Review on Agile Software Development

Anupama Kaushik

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

Abstract: Now-a-days the agile methodologies are the need of an hour for the software development firms. They are susubstituting the traditional methods of software development. Agile process is an iterative approach in which customer satisfaction is at highest priority as the customer has direct involvement in evaluating the software. In this paper a review of major work on agile software development from the year 2012 -2016 is done.

Keywords: Agile Manifesto; Agile software development.

I. INTRODUCTION

Agile is a software development methodology to build software incrementally so that the development is aligned with the changing business needs.

The name "agile" came about in 2001, when seventeen process methodologists held a meeting to discuss future trends in software development. They noticed that their methods had many characteristics in common so they decided to name these processes agile, meaning it is both light and sufficient. In consequence to this meeting, the "Agile Alliance" and its manifesto for agile software development emerged. The agile methods claim to place more emphasis on people, interaction, working software, customer collaboration, and change, rather than on processes, tools, contracts and plans [1].

In today's increasing volatility and uncertainty, talented people want to work in an organization in which they have more control over how they work and how they interact with peers, customers and management. Problems are changing, people are changing and ideas are changing. While there is still a need for plan driven style development and management in some situations the bigger growth lies in agile and flexible.

There are many agile approaches developed so far. The commonly used approaches are Extreme Programming, Scrum, Dynamic System Development Method, Feature Driven Development and Adaptive Software Development.

The Agile Manifesto gathered representatives from Extreme Programming (XP), Dynamics Systems Development Methods (DSDM), Adaptive Software Development (ASD), Scrum, Crystal Methods, Feature-Driven Development (FDD), and others who saw the need for an alternative to documentation driven, heavyweight Traditional software development processes. The manifesto reads as follows (Agile Alliance, 2001): "We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value [2]

- Individuals and interactions over Processes and tool
- Working software over Comprehensive documentation Customer collaboration over Contract negotiation
- Responding to change over following a plan

The previous four values have been further defined by twelve principles: [3]

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes tackle change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organising teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

DOI 10.17148/IJARCCE.2016.5971

II. MAJOR STUDIES FROM 2012-2016

Though agile methodologies are widely used and accepted by the software development firms. The software development community as a whole is still unfamiliar with the role of the requirements engineering practices in agile methods. The term ‘‘agile requirements engineering’’ is used to define the ‘‘agile way’’ of planning, executing and reasoning about requirements engineering activities.

Irum Inayat et al. [4] presented requirements engineering practices adopted and challenges faced by agile teams in order to understand how traditional requirements engineering issues are resolved using agile requirements engineering. They conducted a systematic review of literature published between 2002 and June 2013 and identified 21 papers, that discuss agile requirements engineering.

Maturity in software development is currently defined by models such as CMMI-DEV and ISO/IEC 15504. These models emphasize the need to manage, establish, measure and optimize processes. However, an increasing number of teams have been implementing agile software development methods that focus on people rather than processes.

What, then, is maturity for these agile teams that focus less on detailed, defined pro-cesses? Rafaela Mantovani Fontana et al. [5] addressed this issue. They asked agile practitioners about their perception of the maturity level of a number of practices and how they defined maturity in agile software development. They used cluster analysis to analyze quantitative data and triangulated the results with content analysis of the qualitative data and proposed a new definition for agile software development maturity.

Their findings showed that practitioners do not see maturity in agile soft-ware development as process definition or quantitative management capabilities. Rather, agile maturity means fostering more subjective capabilities, such as collaboration, communication, commitment, care, sharing and self-organization.

Taghi Javdani Gandomani and Mina Ziaei Nafchi [6] developed an empirical framework for Agile transition and adoption using “A Grounded Theory approach”. They have conducted a large-scale empirical research study using Grounded Theory approach with the participation of 49Agile experts from 13 different countries.

Their study inductively developed a substantive Agile transition and adoption frame work which appears to be simple and flexible. The main aim of their paper was to present the developed framework. The primary characteristics of their frame-work, including iterative, gradual, continuous, and value-based are inline with the Agile approach and show promise of being useful in software companies and organizations, regardless of size.

Amadeu Silveira Campanelli and Fernando Silva Parreiras [7] presented “Agile methods tailoring–A systematic literature review”. Their study seeks to evaluate, synthesize, and present aspects of research on agile methods tailoring including the method tailoring approaches adopted and the criteria used for agile practice selection. The method adopted was a Systematic Literature Review (SLR) on studies published from 2002 to 2014.

Mario Spundak [8] presented “Mixed agile/traditional project management methodology – reality or illusion?” They studied both the traditional and agile project management approach, and explored whether there exists a need to combine both approaches on a single project. Their paper covers thorough literature review and starts with the definition of the project management approach and of the project management methodology. The literature review shows what is considered as part of project management methodology in a wider or narrower sense, and what the main characteristics of a methodology are. The need for combining project management approaches is shown on the case of software development project.

Georgios Papadopoulos [9] studied the benefits from “Moving from traditional to agile software development methodologies also on large, distributed projects”. They provided evidence by the analysis of a case study that agile software development methodologies perform better than traditional methodologies in large, distributed projects. Improvements are observed on the quality and on the customer perception of the end product, while agile methodologies allow for requirement changes even late in the project. At the same time, they also build better communication and collaboration in the team which results in enhanced relations between team members and improved employee satisfaction metrics.

Aditi Panda et al. [10] presented Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points. In their study they calculated effort for agile projects and enhanced the prediction accuracy of agile software effort estimation process using Story Point Approach.

For doing this, they used different types of neural networks General Regression Neural Network (GRNN), Probabilistic Neural Network (PNN), Group Method of Data Handling (GMDH) Polynomial Neural Network and Cascade-Correlation Neural Network. Finally performance of the models generated using various neural networks are compared and analysed.

Sergio Galvana [11] presented A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process. In this paper they studied the issue of compliance of Agile Software Development Methodologies (SCRUM, XP, and UPEDU) and the new ISO/IEC 29110 standard.

DOI 10.17148/IJARCCE.2016.5971

III.CONCLUSION

Agile methodologies are gaining popularity in industry. In the past few years research on agile software development suggests that agile methods are effective and suitable for many situations and environments. This paper reviews some of the latest work on agile software development.

ACKNOWLEDGMENT

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

REFERENCES

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

[2] Highsmith, , “Agile software development: the business of innovation”, IEEE Software, 2001

[3] Beck, Kent; et al. (2001). "Principles behind the Agile Manifesto". Agile Alliance.

[4] Irum Inayat , Siti Salwah Salim , Sabrina Marczak , Maya Daneva , Shahaboddin Shamshirband, “A systematic literature review on agile requirements engineering practices and challenges”, Computers in Human Behavior, Vol. 51, pp. 915-929, 2015

[5] Rafaela Mantovani Fontanaa, Isabela Mantovani Fontanac, Paula Andrea da Rosa Garbuioa, Sheila Reinehra, Andreia Malucelliaa Pontifical, “Processes versus people: How should agile software development maturity be defined?”, The Journal of Systems and Software, Vol. 97, pp. 140-155, 2014.

[6] Taghi Javdani Gandomani ,Mina Ziaei Nafchi, “An empirically developed framework for Agile transition and adoption:A Grounded Theory approach”, The Journal of Systems and Software, Vol. 107, pp. 204-219, 2015.

[7] Amadeu Silveira Campanelli,,Fernando Silva Parreiras, “Agile methods tailoring–A systematic literature review”, The Journal of Systems and Software, Vol. 110, pp. 85-100, 2015.

[8] Mario Špundaka, “Mixed agile/traditional project management methodology – reality or illusion?”, Procedia - Social and Behavioral Sciences , Vol. 119 pp. 939 – 948, 2014

[9] Georgios Papadopoulosa,, “Moving from traditional to agile software development methodologies also on large, distributed projects”, Procedia - Social and Behavioral Sciences, Vol. 175, pp. 455 – 463, 2015.

[10] Aditi Panda, Shashank Mouli Satapathy, Santanu Kumar Rath, “Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points”, Procedia Computer Science Vol. 57 pp. 772 – 781, 2015

[11] Sergio Galvana, Manuel Morab, Rory V. O Connorc, Francisco Acostad, Francisco Alvareze, “A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process”, Procedia Computer Science, Vol. 64, pp. 188 – 195, 2015.

Copyright to IJARCCE

DOI 10.17148/IJARCCE.2016.5971