Review on Agile Software Development Methodology

Neeta Chudasama
Assistant Professor
Department of Computer Engineering
K. J. Institute of Engineering & Technology, Savli, Vadodara, India

Abstract— Agile software development has rapidly gained a lot of interest in the field of software engineering. Agile software development, despite its novelty, is an important domain of research within software engineering discipline. In the last 15 years, an excessive number of research studies have been conducted on agile methods, a great number of notable methods have been proposed and various surveys have been presented by many researchers. The various agile Scrum methodologies share much of the same philosophy, as well as many of the same characteristics and practices. In this study, the authors intend to conduct a literature survey study of the surveys of the different agile methodologies ranging from January 2000 to December 2017 using an intuitive research methodology called "Compare and Review" (CR). Agile software development (ASD) methodologies [2] have been gaining acceptance among mainstream software developers since the late 1990s, when they were first postulated in the forms of Scrum [14], Crystal [8], Extreme Programming [5] and other methodologies.

Key words: Agile, Software Methods, Survey, Compare and Review, Research Methodology

I. INTRODUCTION

Agile in essence is an iterative, lightweight and lean software design and development An Agile software design and development methodologies have been gaining rigorous attention in the software engineering research community since their early introduction in the mid-nineties in addition to being highly adopted by the software development industry. Methodology that was born in the late 1990s to be highly compatible with the rapid development of the Software engineering has developed 40 years and emerged many milestones one after another. More and more new software architectures and software development methodologies come out, which transform software development from "arts", "technique" and "individual behavior" into "engineering" and "multi-group cooperation"

We can view agile methods such as Extreme Programming (XP) and Scrum as a reaction to plan-based or traditional methods, which emphasize a "rationalized, engineering-based approach, incorporating extensive planning, codified processes, and rigorous reuse.

Similar to climbing a well-designed ladder where length of all steps and distance between each step is equivalent, agile methods divides a task into small-length iterations that have the same interval size and distance making the transition between iterations much smoother with much higher pace. Agile methodologies try to find How to cite this paper: an equilibrium point between no process and too much process, allowing it to survive in dynamic environments where requirements frequently change while striving high quality software product [3].

Agile encompasses various methodologies, including: Adaptive Software Development (ASD) [4], Agile Unified Process (AUP) [9], Crystal Methods [6],eXtreme Programming (XP) [8], Feature Driven Development (FDD) [9], Kanban [10], Lean Software Development [11], Scrum [12], Scrumban [Ladas 2009 and several variant methods of agile]. The agile methodology is based on the "iterative enhancement" [13] technique [14]. As a iteration based methodology, each iteration in the agile methodology represents a small scale and self-contained Software Development Life Cycle (SDLC) by itself [1]. Unlike the Spiral model [15], agile methods assume simplicity in all practices [14]. In this research, the authors identify the following contributions: 1) a new research methodology called Compare and Review (CR) is used in this paper; 2) A survey of the surveys on agile methodologies were conducted, in which the survey papers were classified into four categories: "Agile Requirements Engineering", "Agile Methods", "Hybrid Agile Methods" and "Miscellaneous"; 3) Several new agile methods that have not been surveyed yet were reviewed and compared in terms of the changes that they proposed on the SDLC.

II. RESEARCH METHODOLOGY

The authors followed a two-stage research methodology called "Compare and Review" (CR), where the first stage aims to compare the survey studies on agile methodologies. While, the second stage intends to review the most recent research studies on XP, Scrum and FDD agile methods that have not been addressed in any previous literature reviews. In the following subsections, the research methodology will be explained.

A. Research Requirements

- Formulate a basic understanding of the different Agile Software Development Methodologies.
- Formulate a comprehensive knowledge of the XP, Scrum and FDD agile methodologies.
- Formulate a comprehensive comparison between the newly proposed agile methods in terms of the changes that they might have made in SDLC phases.

III. SURVEY OF SURVEYS ON AGILE METHODS

In this section, the authors will explore the different surveys that have addressed the agile methodologies in the past 15 years starting from January 2000 to December 2017. The surveys were classified into four categories as illustrated "Agile Requirements Engineering", "Agile Methods", "Hybrid Agile Methods" and "Miscellaneous".

A. Agile Requirements Engineering

One the most attractive features of agile methodology is that it accepts changes to requirements during any phase of the SDLC, making it more flexible and highly adaptable to dynamic environments where requirements change frequently. In this subsection, the authors review the literature

reviews on Agile Requirements Engineering (RE). Baruah presented in [16] a comparative study about the different ways each of the agile methodologies manage requirements. Comparison between the different requirements management techniques under agile methods.

1) XP

User Stories as both written cards and conversations. Written cards are not mandatory for implementation and are only considered "promises for conversation". Requirements are not supposed to be complete or clearly stated. User stories are destroyed after implementation is completed.

a) Customer Role

On-site customer is required to participate in requirements definition, estimation and prioritize.

2) Scrum

User stories are used to represent requirements. The actual requirements are defied based on the discussion of user stories between software owner and software developers.

a) Customer Role

Software owner plays the lead role in defining the requirements.

3) FDD

Requirements are represented using UML diagrams. List of features are used to manage the functional requirements. Requirements are first represented in a high-level context. For each modeling area, the requirements are modeled per domain. After requirements are modeled, it should be peer reviewed.

a) Customer Role

Not specified.

4) Lean Software Development

Just-In-Time methodology is applied in requirements gathering. User stories (cards) are also used by the customer to specify initial requirements and sample screens by the developers. Developers then provide a time estimate for each card.

a) Customer Role

Provide input on sample screens and initial user stories.

5) ASD

Requirements gathering is part of the speculation phase.

a) Customer Role

Not specified.

6) Kanban

User stories are used to define each sprint main goal. Each sprint handles a single user story. Each story is divided into server-side and client-side task. Each task is further divided into subtasks.

a) Customer Role

Not specified.

7) AUP

Requirement phase consists of the following activities: (1) identifying stakeholders, (2) understanding problem, (3) establishing a basis of estimation, and (4) defining user interface. User stories are used in the construction phase. Requirements are presented as Business Use Case Model.

a) Customer Role

Not specified

B. Agile Methods

They determined the papers, which could be trusted and recommend that the quality level of researches could be improved. The author concluded that the majority of the surveyed studies were incomprehensive in addition to being not trustworthy. On the other hand, the authors provided some recommendations in order to raise the research quality

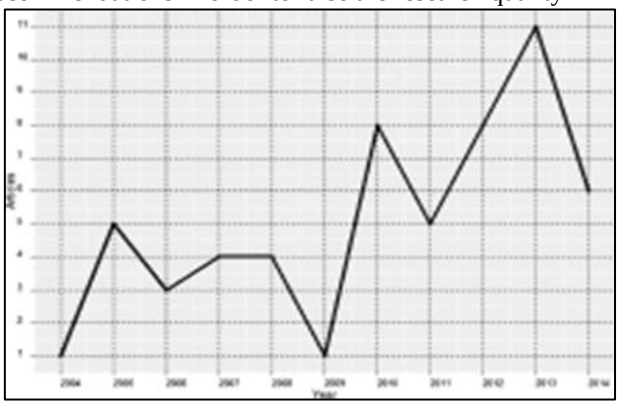

Fig. 1: Number of Papers on Agile Methods per Year
Adopted from [4]

C. Hybrid Approach

If a pure Scrum approach doesn't work for your project, you can also try a hybrid model. There are several methodologies that combine the principles of Agile or Scrum and adapt the framework to scale more effectively.

For example, Disciplined Agile Delivery (DAD) builds on the practices of Agile, Scrum, and Lean to provide a solid foundation from which to scale. DAD was developed to provide a more cohesive approach to Agile, taking strategies from Scrum, Kanban, Extreme Programming, and others. Rather than taking the time to learn one of these existing frameworks and cobble them together as needed, DAD already combines all relevant techniques.

Other hybrid methods include Large-Scale Scrum (LeSS), which extends Scrum with scaling rules and guidelines, and Scaled Agile Framework (SaFE), based on underlying Lean and Agile princi

D. Miscellaneous

A web-based survey conducted by Begel and Nagappan of employees who are working on the software production processes [7]. The survey investigated how Microsoft employees use agile software development methods and how they penetrate of agile software development practices and their perceptions of why agile works well or poorly on their software teams. The employee's responses indicate that around one-third of the respondents use agile, and SCRUM is the most popular method with 65% of the respondents were using it in their team, and most of agile users have a positive opinion about it.

IV. EVALUATION OF THE SURVEY

In this survey, the authors study several surveys related to agile software development. Figure 2 presents the ratio of published papers per publisher. It shows that Elsevier has the highest percentage of 38%, while Wiley has the lowest percentage of 8%.

Notably, "Hybrid Agile Methods" category was the most of surveyed in the literature showing the importance of agile method compared to the other development method.

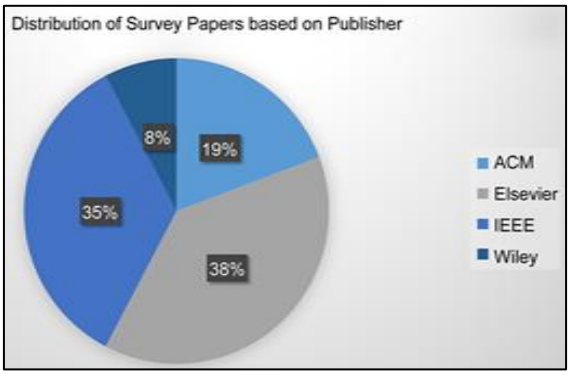

Fig. 2: The Ratio of Published Papers per Publisher

V. CONCLUSION

Agile is considered one of the most popular software design and development methodologies. In this study, a literature survey study of the surveys of the different agile methodologies ranging from January 2000 and December 2015 has been conducted. In this survey studies were selected for review and evaluation using a new proposed research methodology called "Compare and Review". The surveyed studies classified into four categories: "Agile Requirements Engineering", "Agile Methods", "Hybrid Agile Methods" and "Miscellaneous". Moreover, four newly proposed agile methodologies were reviewed, analyzed and compared. The evaluation shows that most of surveys were proposed and published in 2017, and the most surveyed category were the Hybrid Agile methods.

REFERENCES

[1] Williams, L. (2010) Agile Software Development Methodologies and Practices. Advances in Computers, 80, 1-44. https://doi.org/10.1016/S0065-2458(10)80001-4
[2] Andrew Begel, Nachiappan Nagappan, "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study", Microsoft Research One Microsoft Way Redmond WA 98052.
[3] Al-Zewairi1, M., Biltawi, M., Etaiwi, W. and Shaout, A. (2017) Agile Software Development Methodologies: Survey of Surveys. Journal of Computer and Communications, 5, 74-97. https://doi.org/10.4236/jcc.2017.
[4] ampanelli, A.S. and Parreiras, F.S. (2015) Agile Methods Tailoring—A Systematic Literature Review. Journal of Systems and Software, 110, 85-100. https://doi.org/10.1016/j.jss.2015.08.035
[5] Highsmith, J. (1997) Messy, Exciting, and Anxiety-Ridden: Adaptive Software Development. American Programmer, 10, 23-29.
[6] Cockburn, A. (2004) Crystal Clear a Human-Powered Methodology for Small Teams. Addison-Wesley, Reading.
[7] Begel, A. and Nagappan, N. (2007) Usage and Perceptions of Agile Software Development in an Industrial Madrid, 20-21 September 2007, 255-264. https://doi.org/10.1109/ESEM.2007.12
[8] Anderson, A., Beattie, R. and Beck, K. (1998) Chrysler Goes to Extremes. Disruted Computers, 24-28.
[9] Ambler, S. (2006) The Agile Unified Process (AUP) http://www.ambysoft.com/unifiedprocess/agileUP.html
[10] Ladas, C. (2009) Scrumban-Essays on Kanban Systems for Lean Software Development. Modus Cooperandi Press, Seattle.
[11] Poppendieck, M. and Poppendieck, T. (2003) Lean Software Development: An Agile Toolkit. Addison-Wesley, Boston.
[12] Schwaber, K. (1996) Controlled Chaos: Living on the Edge.
[13] Basili, V.R. and Turner, A.J. (1975) Iterative Enhancement: A Practical Technique for Software Development. IEEE Transactions on Software Engineering, SE-1, 390-396. https://doi.org/10.1109/TSE.1975.6312870
[14] Cohen, D., Lindvall, M. and Costa, P. (2004) an Introduction to Agile Methods. Advances in Computers, 62, 1-66. https://doi.org/10.1016/S0065-2458(03)62001-2
[15] Boehm, B. (1986) A Spiral Model of Software Development and Enhancement. Software Engineering Notes, 11, 14-24. https://doi.org/10.1145/12944.12948 https://ars.els-cdn.com/content/image/1-s2.0-S0164121215001843-gr4.sml

