IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

Role of agile methodology for software product development

Jyoti Kharade[1*], Sneha Prajapati[2], Dakshata Narkar [3] , Dhanaji S. Kharade [4]

[1*] Bharati Vidyapeeth's Institute of Management and Information Technology, Navi Mumbai, India
[2] Bharati Vidyapeeth's Institute of Management and Information Technology, Navi Mumbai, India
[3] Bharati Vidyapeeth's Institute of Management and Information Technology, Navi Mumbai, India
[4]Bharati Vidyapeeth's College of Engineering, Navi Mumbai, India

[*]Corresponding Author: kharadejyoti09@gmail.com, Tel.: +91-9082604102

Available online at: www.ijcseonline.org

Accepted: 21/May/2018, Published: 21/May/20182018

Abstract**—** Increasing complexities in the requirements set on programming solution due to change in the business activities has lead to need of modern approaches, also known as agile methodologies also known as lightweight, which claim to provide solution to above said problem. Knowledge Management (KM) can be easily accepted into agile software development (ASD) environments. Agile software development processes include some practices that support Knowledge Management (KM). KM is about learning, and ASD set up an environment that supports learning processes. In this paper, attempt is made to study the role of Agile methodology in software development, find barriers in applying agile methodologies in software development and to propose the role of agile methodologies for enhancing software development.

Keywords**—**Agile Methodology, Software Development, Knowledge Management

I. Introduction

Ever since software started to be created in the 1950's, it has had an unprecedented track record of amazing impact on business and society, of being the source of incredible wealth and disastrous failure, of triumphant success and deep frustration. The more important software became from a business perspective, the stronger became the desire to make the process of creating software more manageable, more reliable, more "engineering"-like or more manufacturing-like. So it reflected more wishful thinking than reality when the term "software engineering" was coined in 1968 or the term "software factory" in the 1980's [1]. And there are good arguments why these terms do still not describe reality [2,3].
Heavyweight methodologies, commonly known for its traditional ways to develop software put emphasis on comprehensive planning, detailed documentation, and expansive design. Unlike traditional methods, agile methodologies employ short iterative cycles, and rely on tacit knowledge within a team. In today's competitive and complex global market, companies are required to manage their intellectual resources as well as their financial resources. Therefore, KM is accepted as a genuine management practice that helps organizations to distribute the right knowledge to the right people at the right time [4].

II. Related Work

Methodologies help in imposing a closely controlled process upon software development with a goal to make a software development process more efficient and predictable [5]. Agile is among the one methodology which means being able to deliver quickly, change quickly, and change often With an agile approach, one can deliver business-oriented results rapidly and effectively[6]. Differentiating from the working point of view of traditional methods, agile methodologies use short iterative cycles, and rely on tacit knowledge within a team. The name lightweight or agile can be defined as "1) marked by ready ability to move with quick easy grace or 2) having a quick resourceful and adaptable character" [7]. The name "agile" came about in 2001, when seventeen process methodologists held a meeting to talk about the future trends in software development. The outcome to this meeting was the formation of "Agile Alliance" and its manifesto for agile software development [8]. Agile in essence is an iterative, lightweight and lean software design and development methodology that was born in the late 1990s to be highly compatible with the rapid development of the WWW (World Wide Web) [9]. An agile method generally encourages incremental development and delivery of software product. This process should be able to allow changes occurring during the development phase and should be adaptive in nature [10].

© 2018, IJCSE All Rights Reserved

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 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. Unlike other methods, agile methods rely on feedback as control mechanism which ensures greater customer satisfaction [11, 12].

Agile techniques may differ in practices (XP, ASD, DSDM, SCRUM, CRYSTAL etc.) but, they contribute to common characteristics, including iterative development, and a focus on interaction and communication. Agile methods is not the practices they use, but their recognition of people as the main driving force which can lead to project success [13]. Many surveys have claimed XP and Scrum methodologies are widely used in the industry and are the most popular [14, 15, 16]).

III. To find the barriers /applying agile methodologies in software development

KM and Agile Software Development (ASD) are two organizational processes that face common barriers when introduced and applied in software development. The main barrier in launching the product development in agile software development and applying knowledge management into software organizations is the need to deal with the conceptual change, mainly the organizational cultural change that ASD and KM brings when introduced. Many studies have revealed that the introduction of KM and ASD processes have increased productivity, shortened time-to-market and resulted in higher product quality [17, 18]).

The common patterns which emerge between the two studies suggest that barriers to learning can be classified into four main areas: Multiple Goals (Projects), Excessive Iteration Pressure, Level of Customer Involvement, and Organizational Culture. As individual, team, and customer/organizational learning are each high probability outcomes of the utilization of agile methods, the emergence of barriers to this learning may be counter intuitive, given the inherent propensity for learning as a result of adoption [19].

Under a survey to adopting agile practices while developing medical device software the following barriers were identified[20]

- Lack of Documentation;
- Traceability Issues;
- Regulatory Compliance;
- Lack of Up-Front planning;
- Managing Multiple Releases.

Agility is ability to respond to unpredictable changes with quick response and profitability [21]. Agile Development is a type of development that ―encourages customer satisfaction and early incremental delivery of the software; small, highly motivated project teams; informal methods; minimal software engineering work products; and overall development simplicity. It puts an emphasis on delivery and communication with clients [22]. In fact, the clients are a part of the 3 stakeholders in development, along with the users and the developers. There are various methods of agile development, with two of the most popular being extreme Programming and Scrum. There are 12 principles of Agile Engineering [23].

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

IV. To propose the role of agile methodologies for enhanching software development

Agile – denoting “the quality of being agile; readiness for motion; nimbleness, activity, dexterity in motion” – software development methods are attempting to offer once again an answer to the eager business community asking for lighter weight along with faster and nimbler software development processes. This is especially the case with the rapidly growing and volatile Internet software industry as well as for the emerging mobile application environment.

First, the agile movement emphasizes the relationship and communality of software developers and the human role reflected in the contracts, as opposed to institutionalized processes and development tools. In the existing agile practices, this manifests itself in close team relationships, close working environment arrangements, and other procedures boosting team spirit. Second, the vital objective of the software team is to continuously turn out tested working software. New releases are produced at frequent intervals, in some approaches even hourly or daily, but more usually bi-monthly or monthly. The developers are urged to keep the code simple, straightforward, and technically as advanced as possible, thus lessening the documentation burden to an appropriate level. Third, the relationship and cooperation between the developers and the clients is given the preference over strict contracts, although the importance of well drafted contracts does grow at the same pace as the size of the software project. The negotiation process itself should be seen as a means of achieving and maintaining a viable relationship. From a business point of view, agile development is focused on delivering business value immediately as the project starts, thus reducing the risks of non-fulfilment regarding the contract. Fourth, the development group, comprising both software developers and customers representatives, should be well-informed, competent and authorized to consider possible adjustment needs emerging during the development process life-cycle. This means that the participants are prepared to make changes and that also the existing contracts are formed with tools that support and allow these enhancements to be made [24].

Agile methods, such as eXtreme Programming (XP) and Scrum, promise practices for improved collaboration, communication and project management. This is because, in agile software development, the planning is made more frequently than in so called ‘plan-driven’ software development. The constant planning enables these planning driven teams to respond to changes quickly.These are some of the reasons why agile methods have been increasingly attractive to software intensive companies [25].

The usefulness of XP and Scrum practices, however, vary between organizations and projects. This means that at the same time when some of the XP and Scrum practices such as (1) collective code ownership, (2) 40 h week, (3) coding standards and (4) simple design, (5) product backlogs and (6) sprint planning meetings seem to be very useful for companies, some other XP and Scrum practices such as (1) pair programming, 12 (2) TDD, (3) On-site customers and (4) product backlogs have been discovered to have negative effects (i.e. were not seen useful) on the software development organizations [26].

V. Conclusion

If organizations focus on agile methodologies and make sure they happen, they can scale the ability to deliver. Agile methodologies has been proven to be advantageous when it comes to software development, resulting in high quality software produced in a cost-effective manner. There is no doubt that it will soon be used by more organizations involved in product development, aside from software development.

References

[1] Kittlaus, H.-B. 2003. Software Engineering und Software Fabrik - vom Nutzen und Schaden der Metapher in der Informatik, In: Informatik-Spektrum 26: p. 8 - 12, and p. 291 – 292

[2] Davis, M. 2011. “Will Software Engineering Ever Be Engineering?”, in: CACM 54 (11), p. 32-34

[3] White, J., Simons, B. 2002. ACM’s Position on the Licencing of Software Engineers, CACM 45 (11), p. 91

[4] Spek, R. Kruizinga, E Annelies Kleijsen, 2009 Strengthening lateral relations in organisations through knowledge management, Journal of Knowledge Management, Vol. 13 Iss: 3, pp.3 – 12

[5] Awad, M. A. 2005 Comparison between Agile and Traditional Software Development Methodologies, Honours Programme Thesis, University of Western Australia.

[6] Highsmith, J. Orr, K. and Cockburn, A. 2000 Extreme Programming, E- Business Application Delivery, pp. 417

[7] Job Trends www.indeed.com/jobtrends?q=agile%2C+scrum%2C+%22extreme+programming%22%2C+%22test+driven%22 &l=,

[8] AmitojSingh ,Kawaljeet Singh , Neeraj Sharma “Managing Knowledge in Agile Software Development” International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT2012) Proceedings published in International Journal of Computer Applications® (IJCA)

[9] 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

[10] Tudor, D.2006 An update on Agile methods, ITadviser, Issue 56.

[11] El-Haik, B.S. and Shaout, A. (2010) Software Design for Six Sigma: A Roadmap for Excellence. Wiley, Hoboken. https://file.scirp.org/pdf/JCC_2017033115471602.pdf

[12] Highsmith, J. and Cockburn, A. (2001) Agile Software Development: The Business of Innovation. Computer, 34, 120-127. https://doi.org/10.1109/2.947100

[13] Highsmith, J and Cockburn, A. 2001 Agile Software Development: The Business of Innovation, Computer, Vol. 34,No. 9, pp. 120-122.

[14] Version one, 2008 3rd Annual Survey: The State of Agile Development, www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf.

[15] Version one, 2007 Agile development: Result Delivered, http://www.versionone.net/pdf/AgileDevelopment_ResultsDelivered.pdf.

[16] Versiion One, 2009 4th Annual Survey 2009, www.versionone.com/agilesurvey

[17] Bennet, D. and Bennet, A. 2003 The Rise of the Knowledge Organisation, chapter 1 in Holsapple, C.W. (ed.), Handbook on Knowledge Management,Vol 1, Springer, Berlin, pp. 5–20.

[18] Reifer, D. J. 2002 How to Get the Most out of Extreme Programming/Agile Methods , In proceeding Proc. Extreme Programming and Agile Methods - XP/Agile Universe

[19] Jeffry S. Babb1 , RashinaHoda 2, and Jacob Nørbjerg3 "Barriers to Learning in Agile Software Development Projects" Department of Computer Information and Decision Management, West Texas A&M University, 2403 Russell Long Blvd. Canyon, Texas USA, 79016 jbabb@mail.wtamu.edu

[20] Martin McHugh, Fergal McCaffery, and Valentine Casey "Barriers to Adopting Agile Practices when Developing Medical Device Software" Regulated Software Research Group, Department of Computing and Mathematics, Dundalk Institute of Technology &Lero, Dundalk Co. Louth, Ireland {martin.mchugh, fergal.mccaffery, val.casey}@dkit.ie

[21] Erande, Ameya S., and Alok K. Verma. "Measuring agility of organizations-a comprehensive agility measurement tool (CAMT)." International Journal of Applied Management and Technology 6.3 (2008).

[22] Pressman R. S. ―Software Engineering: a Practitioner's Approach‖ (2010) 7th ed.(McGraw-Hill, New York).

[23] Arun Kumar Kamepally ,TejaswiniNalamothu "Agile Methodologies in Software Engineering and Web Engineering" Department of Computer Science, Kennesaw State University, Kennesaw, Georgia, USA

[24] PekkaAbrahamsson, OutiSalo, JussiRonkainen and JuhaniWarsta "Agile Software Development Methods: Review and Analysis" https://arxiv.org/ftp/arxiv/papers/1709/1709.08439.pdf

[25] Agile Manifesto, 2001. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. Manifesto for Agile Software Development. http://AgileManifesto.org.

[26] Abrahamsson, P., Salo, O., Ronkainen, J. &Warsta, J. 2002. Agile Software Development Methods: Review and Analysis. Espoo. 408. VTT Publications 478. 107 p. http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf

Authors Profile

Dr Jyoti . Kharade, Bachelor of Science,Master of Computer Application from Shivaji University, M.Phil from Bharati Vidyappeth deemed University and Ph.D from SNDT University. She is currently working as Associate Professor in Bharati Vidyapeeth's Institute of Management and Information Technology, University of Mumbai, since 2004. She is a member of CSI. She has published more than 27 research papers in reputed international journals including conferences. Her main research work focuses on e-Governance, Data Mining. She has 16 years of teaching experience.

Ms. Sneha Prajapati pursed Bachelor of Science (Information Technology) from University of Mumbai, Mumbai, India in 2015. He is currently pursuing Masters Of Computer Applications from University of Mumbai, Mumbai, India.

Ms. Dakshata Navkar pursed Bachelor of Science (Information Technology) from University of Mumbai, Mumbai, India in 2015. He is currently pursuing Masters Of Computer Applications from University of Mumbai, Mumbai, India.

Dhanaji S. Kharade, Bachelor of Engineering from Shivaji University. He is currently working as Assistant Professor in Bharati Vidyapeeth's College of Engineering, University of Mumbai, since 1999. He is a member of ISTE. He has 21 years of teaching experience.