

Current State of Agile Methodologies and its extended practices in Software Development - A Review

Mrs. Shivaleela S¹, Dr. L .Manjunatha Rao²

Asst. Professor, Department of MCA, Dr. Ambedkar Institute of Technology, Bangalore-56, India ¹

Professor and Head, Department of MCA, Dr. Ambedkar Institute of Technology, Bangalore-56, India ²

Abstract: In fast changing world agile has become successful in software because of its acceptance of changing requirement and fast delivery of the software product. Once an organization decides to adopt an agile development regimen, there is still plenty of difficult research and decision-making to be done for the project architect /manager. There have been many studies and suggestions in improving the software project development process .The aim of this paper is to document the view on introduction of various agile methodologies, current state of agile adoption, reasons, benefits and cautions of adopting agile methodology and its extended practices. The research method follows grounded theory of qualitative research method, in the first phase systematic literature survey of papers has been done. Structured analysis techniques have been applied to develop well-grounded theories. The basis of this survey will serve as foundation to my research work.

Keywords: Agile Methodologies, Extreme Programming, Lean, DevOps, challenges

I. INTRODUCTION

Agile Development is an umbrella term for several iterative and incremental software development methodologies. That represents a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change. The most popular agile methodologies include Extreme Programming (XP), Scrum, Crystal, Dynamic Systems Development Method (DSDM), Lean Development, and Feature-Driven Development (FDD). The outline of Agile Methodologies was laid down by the Agile Manifesto, published by a group of software practitioners (Beck et. al, 2001).^[1] Agile development methodologies provides flexibility to handle changing requirements, improved communication and coordination mechanisms, and improved quality while enhancing the time-to market speed^[3] This paper categorized as related work, various agile methodologies, Current state of agile methodologies ,benefits and challenges of adopting agile methodology.

II. RELATED WORK

A large number of studies have been conducted on agile methodologies with a focus on agile transformation; agile adaption, benefits and challenges of applying agile methodologies but there is a lack of studies which include all agile practices and its extended practices like Devops. Some of the following papers discussed about Agile methodologies: Abdul Rauf^[5] performed an empirical survey of benefits and challenges to accumulate the experience of industry practitioners in order to better understand the benefits and challenges of applying agile practices in projects. The study highlighted benefits and challenges of agile by considering only XP and SCRUM practices. Surbhi R. Khare ^[2] insights into the general state of research on agile ISD in terms of research approaches, the results show that the state of research on agile ISD is still nascent because there is an imbalance in terms of the employed research methods towards interview-based case studies. Those qualitative research designs are essential for providing first evidence on important factors and relationships

III. DIFFERENT AGILE METHODOLOGIES

Following section introduces the various agile methodologies at high level:

A. Extreme Programming: (XP) is a methodology for creating software within a very unstable environment. It allows flexibility within the modelling process. The main goal of XP is to lower the cost of change in software

requirements. This means that the cost of changing the requirements at a later stage in the project - something that is very common in the real-world can be very high. Extreme programming (XP) which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints at which new customer requirements can be adopted.^[2] Addresses requirements through user stories and onsite customer .User Stories of two components: written card & conversations after the written card. Written cards are just "promises for conversation". Cards need not be complete or clearly stated. Story cards destroyed after implementation^[11]

B. Scrum: This is one of the light weight method victimize for the development of software. Its principle lies in the fact that small teams working cross functionally produce good results. Scrum is more revenue centric with attention on improving revenue and quality of the software. Since being lightweight it can adapt to changing requirements and releases the software in small release cycles called sprints. Scrum has three roles Product Owner, Scrum Master and team member. Scrum is an iterative and incremental agile software development framework for managing product development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal", challenges assumptions of the "traditional, sequential approach" to product development, and enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members, as well as daily face-to-face communication among all team members and disciplines in the project.^[2]

On a high level, the development is constructed from multiple subsequent sprints, where an increment of the software is developed. Sprints are planned by selecting items from a backlog and estimating the effort needed to complete each item selected for the sprint. During sprints, the team groups up every day for a daily scrum meeting, where the status of the tasks is tracked. At the end of the sprint, a sprint review and demo is organized. Learning is emphasized in every sprint with a sprint retrospective meeting^[6]. The most widely used agile method SCRUM, on the contrary, provides good management techniques. Because this work focuses on software engineering, rather than software management,^[8]

Scrum is a framework with roots from Control Theory, which has been used for years to address complex adaptive problems. As any framework, one is free to tailor it, in order to fit the specific purposes, employing various processes, techniques, and best practices.^[9]

Weak points or limitations present in the Agile SCRUM: 1. Unstructured process of working 2. Unsuitability for large-scale organizations 3 Lack of accurate documentation and artefacts 4 Absence of problem solving phase 5 Not architecture-based 6. No activity of building prototypes or conceptual model of the product^[10]

Strengths or merits of the Agile SCRUM: 1. Adaptive nature, SCRUM is very flexible to adapt or combine the new knowledge and can give better performance. 2. SCRUM can evaluate the all iterative deliverables by using the Sprint retrospective phase. 3. SCRUM is simple framework and follows the required agile principles and became a famous^[10]

Requirement management description: Also addresses requirements through user stories. Thus discussion of user stories which defines actual requirements. So, product owner plays the lead role in the development of the software.^[11]

C. Feature-Driven Development (FDD): The FDD approach focuses on the software features of the system. They are the main driver of the entire development process. It differs significantly from the other agile processes because they put a strong emphasis on planning and upfront design. More emphasis on quality, frequent and tangible deliveries and accurate monitoring of project progress. It is also iterative and incremental software development process. FDD blends a number of industry-recognized best practices into a cohesive whole. These practices are all driven from a client-valued functionality (feature) perspective. Its main purpose is to deliver tangible, working software repeatedly in a timely manner.^[2]

It uses the notion of feature to express functional requirements (Palmer and Felsing, 2001). A feature is typically expressed as: <action> <result> <object>; e.g., 'check the availability of seats on a flight'. Due to their object oriented nature, features should be elicited after identifying the problem domain classes. Each feature belongs to one feature-set (activity), expressed as: <action>-ing a(n) <object>; e.g., 'reserving a seat'. Each feature-set belongs to one area, expressed as: <object> management; e.g., 'ticket management'.^[7]

FDD gathers user requirements & represents in a UML diagram with a list of features .Feature list manage functional requirements and development tasks. Solution requirements analysis begins with a high level examination of the scope of the system and its context. The team assesses the domain in detail for each modelling area. Small groups composes a model for each domain and present the model for peer review.^[11]

D. Crystal: Crystal Methods is an agile software development methodology developed by Alistair Cockburn in which people are more emphasized in software development rather than tools or processes^[15, 16]. The Crystal methodology is one of the most lightweight, adaptable approaches to software development. Crystal is comprised of a family of agile methodologies such as Crystal Clear, Crystal Yellow, Crystal Orange and others, whose unique characteristics are driven by several factors such as team size, system criticality, and project priorities.

Crystal methods are a toolkit of methodology elements to suit individual projects. Large or safety critical projects require more methodology elements than small non-critical projects. With Crystal Methods, organizations only develop and use as much methodology as their business needs demand. Basically this is used for small teams and small projects that are not life critical.^[14]

E. Adaptive Software Development: Adaptive software development (ASD) is a management to control software project for changeable requirement and short period project. ASD Promotes adaptive paradigm, derives principles from radical software development (phases support in SDLC) Requirements, design, code, unit test, integration test, system test, acceptance test. It is more about concepts and culture rather than in practice. Adaptive software development (ASD) is a software development process that grew out of rapid application development work by Jim High smith and Sam Bayer. It embodies the principle that continuous adaptation of the process to the work at hand is the normal state of affairs. This dynamic cycle provides for continuous learning and adaptation to the emergent state of the project. The characteristics of an ASD life cycle are that it is mission focused, feature based, iterative, time boxed, risk driven, and change to learnt^[2]

Requirements gathering are done in speculative phase. First, setting the project's mission and objectives, understanding constraints, establishing project organization, identifying & outlining requirements, making initial scope estimates & identifying key project risks Project initiation data is gathered in a preliminary JAD sessions^[11]

F. Lean Development: Lean software development is a software development philosophy that aims to streamline the production of applications and software products, and to make them more efficient. One of the key ideas in Lean Production - eliminate waste - is drawn from TPS Toyota Production System.^[17]User requirements gathering are done by presenting screens to the end-users & getting their input. Just in time production ideology applied to recognize specific requirements & environment. At the beginning customer provides the needed input presented in small cards or stories. Developers estimate the time needed for the implementation of each card. Work organization changes into self-pulling system, each morning during stand-up meetings^[11]

G. Agile Modelling: Agile modelling (AM) is a methodology for modelling and documenting software systems based on best practices. It is a collection of values and principles that can be applied on an (agile) software development project. This methodology is more flexible than traditional modeling methods, making it a better fit in a fast changing environment. It is part of the Agile software development tool kit. Agile modeling is a supplement to other agile methodologies such as Scrum, extreme programming (XP), and Rational Unified Process (RUP). It is explicitly included as part of the disciplined agile delivery (DAD) framework. As per 2011 stats, agile modeling accounted for 1% of all agile software development.^[2]

H. Agile Unified Process (AUP): Agile Unified Process (AUP) is a simplified version of the Rational Unified Process (RUP) developed by Scott Ambler. It describes a simple, easy to understand approach to developing business application software using agile techniques and concepts yet still remaining true to the RUP. The AUP applies agile techniques including test-driven development (TDD), Agile Modeling (AM), agile change management, and database refactoring to improve productivity^[2]

Requirement phase includes identifying the stakeholders, understanding the user's problem, establishing a basis of estimation and defining user interface for the system. Activities occur during the Inception phase & Elaboration phases but continue through the phases to improve the unfolding design. The deliverables are the business use case model. In construction phase, user stories implemented & iteratively reworked to reflect understanding of problem domain as the project progresses^[11]

I. Dynamic systems development method (DSDM): DSDM is a project delivery framework, primarily used as an agile. DSDM originally sought to provide some discipline to the rapid application development (RAD) method.^[3] It is a generic approach to project management and solution delivery. DSDM is an iterative and incremental approach that embraces principles of agile development, including continuous user/customer involvement. DSDM fixes cost, quality and time at the outset and uses the MoSCoW prioritisation of scope into musts, should, could and won't have to adjust the project deliverable to meet the stated time constraint.^[2]

In DSDM there are four requirement management phase. Feasibility phase: Requirements for a particular project is gathered & checked for feasibility & prioritization^[11]

J. Kanban: Kanban is neither a software development lifecycle methodology nor a project management approach. Instead, Kanban can be applied to incrementally change and improve some underlying, existing process [1]. It is an evolutionary process model that allows each Kanban implementation to be different, suited for each context. On the other hand, LeanSD describes Kanban as part of one of its tools: “Pull Systems”^[17].

Kanban, too, defines certain principles; for example, Kanban systems are always pull systems. Work is pulled to development only when there is capacity, compared with some other systems where work is pushed to development.^[6] User stories will be used to understand what the actual goals of a sprint were. A sprint contains one story card. The tasks divides a user story into smaller pieces. A story is divided into client side & server-side task. The tasks were divided into sub-tasks. Developers minimize the amount of items within a sprint to maintain time of the project^[11]

K. Devops Extended Practices: DevOps can be interpreted as an outgrowth of Agile—agile software development prescribes close collaboration of customers, product management, developers, and (sometimes) QA to fill in the gaps and rapidly iterate towards a better product ... [DevOps acknowledges that] service delivery and how the app and systems interact are a fundamental part of the value proposition to the client as well, and so the product team needs to include those concerns as a top-level item. From this perspective, DevOps is simply extending Agile principles beyond the boundaries of the code to the entire delivered service.^[19]

DevOps (a portmanteau of “development” and “operations”) is a software development method that extends the agile philosophy to rapidly produce software products and services and to improve operations performance and quality assurance. It was born to accelerate the delivery of Web-based systems and quickly bring new value to users. Many Web-based systems evolve according to usage trends without a clear long-term goal.^[12] DevOps approach includes an automated software deployment mechanism focusing on the rapid and repeatable release of software changes and automated management of operational infrastructure.

IV. CURRENT STATE OF AGILE METHODOLOGIES

In account with the 12th state of agile report survey key benefits and challenges of adopting agile methodologies is shown below.^[18]

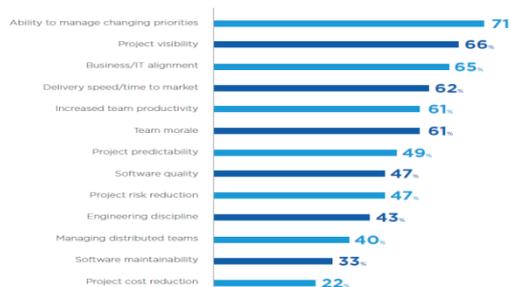


Fig.1. Benefits of agile from 12th state of agile report

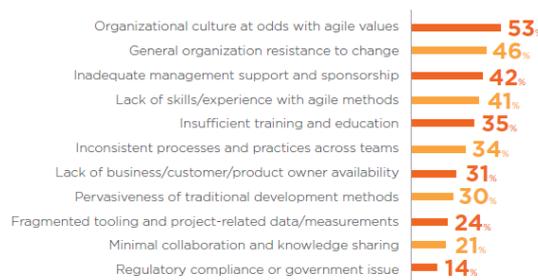


Fig.2. Key challenges experienced adopting agile from 12th state of agile report

V. BENEFITS OF AGILE METHODOLOGIES

Software companies do not follow any specific agile method completely; rather they use a mix of agile practices. Hence it is useful to capture the views of industry practitioners behind using the specific practices and abandoning the other ones. The experience with agile practices varies from project to project and team to team, hence it is worth

accumulating the benefits and challenges associated with individual practice usage. This research motivation leads us to the following research question: What are challenges and benefits of agile practices in state of practice? ^[5]

The industries who have adopted agile practices are already leveraging and seeing the advantages for their respective projects, some of the key benefits and challenges described from our detail introduction of all agile methods and from survey papers of agile methodologies to develop software projects. And are summarized as following

a. Iterative and adaptive to the changing environment: In agile software development method, software is developed over several iterations Each iteration is runs through its own life cycle of analysis, design, and implementation and testing after each iteration the mini project is delivered to the customer for their use and feedback. Any adhoc changes that upgrade the software are welcome from the customer at any stage of development and that changes are implemented.

b. customer satisfaction being the key: This agile methodology requires active client or customer involvement throughout the development cycle. The deliverables developed after each iteration is given to the user for use and improvement is done based on the customer feedback only. So at the end what we get as the final product is of high quality and it ensures the customer satisfaction as the entire software is developed based on the requirements taken from customer.

c. Light process overheads and enhanced communication: The process overheads are very light weight and focused towards delivering more meaningful deliverables over sprints, for example documentation in agile methodology is short and to the point though it depends on the agile team. Generally they don't make documentation on internal design of the software. The main things which should be on the documentation are product features list, duration for each iteration and date. Short and frequent meeting helps countdown too much of documentation and reduces unambiguity of remote or virtual means of interaction leading better communication. The unwanted process cut down saves time of development and delivers the project in least possible time.

d. creating visibility and transparency throughout development cycle : Agile mandates incremented mini software development cycle involving short meaningful deliverables to the customers after every sprint and regular feedbacks are taken from the customers, it warns developers about the upcoming problems which may occur at the later stages of development. It also helps to discover errors quickly and they are fixed immediately.

VI. CHALLENGES OF ADOPTING AGILE METHODOLOGIES

Although there are increasing adoption rates in the agile methodologies, and companies have started seeing the relative benefits over other existing models, there exists challenges and on the field cautions and concerns the teams are going through surrounding the new models. Thus demanding a need for continuous efforts for the system and community as a whole to evolve, and help mature the processes and practices in dealing with the challenges.

a. Continuous customer involvement and feedback is crucial to agile success: Regular customer feedback to the development progress and releases is key to agile practice. It's very important the client representative is clear on the product requirements and has active involvement in clearing delivery milestones.

b. Iterative development cycle's introduces frequent unplanned changes: Agile gives the flexibility to introduce change requirements at any phase leading to unprecedented design changes in the software, this may introduce potential challenges in maintaining a fixed design documents and change logs, although Agile mandates minimal or least documentation that saves development time as an advantage of agile method, on the other hand it is a big disadvantage for developer. Here the internal design is getting changed again and again depending on user requirements after every iteration; this poses a major challenge to maintain the detail documentation of design and implementation because of project deadline. Due to this frequent change and less documentation, leading to new members spending more time in catching up the project details and getting productive with less dependency

c. Continuous feedback involves high rate of rejection amounting rework: Early customer feedback involvement builds in the confidence in the project deliverables and the opportunity to validate the product getting developed, but as well leads to high early rejection rates involving lot of time wasted for the investment made in delivering the components and as well redesign and rework of the entire component or project in extreme cases leading to low morale.

d. Involves and demands a huge mindset change: The agile methodologies demands a huge mindset change and needs efforts from companies to break the barriers of big companies who have been used to and adapted the older practices and tied to lot of perceptions around adopting something new.

VII. CONCLUSION

Agile, with an expanding and self-sustaining pace, has reached out to nearly every nook and corner of the world where IT lives. It is no more considered as an alternative to IT process, but a must-have to break the ground rules of traditional development models. Since no process is without flaws; agile too has its own negatives of being highly successful only in co-located small teams and performing weakly in distributed environments. However, the advantages of going the agile way may outweigh the cons that can definitely be overcome using strategic development practices. In this paper the survey clearly brings out the advantages of adopting agile methodologies. As enterprises are clearly seeing the benefits of adopting the agile, they also see clearly challenges while execution on the ground. This calls out for improvising and customizing the practices suitable for the unique nature of the project while keeping the fundamental practices in tact as Further detailed study taking into consideration real projects metrics executed for enterprises to bring out the advantages and challenges in adopting the existing methodologies and customizations unique to the project and organization culture.

REFERENCES

- [1] Năftănăilă Ionel, Piața Romană “Agile Software Development Methodologies: An Overview Of The Current State Of Research “ JEL code:M10
- [2] Surbhi R. Khare1, Ritesh Shrivastava2 ” Literature Survey on Agile Information Systems Development” International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 Volume: 02 Issue: 08 | Nov-2015 p-ISSN: 2395-0072
- [3] P. Abrahamsson , J. Warsta, M.T. Siponen , J. Ronkainen, “New Directions on Agile Methods: A Comparative Analysis” . Proceedings of 25th International Conference on Software Engineering, pp. 244 –254, 2003
- [4] Abrahamsson, P., Salo, O., Ronkainen, J., 2002. “Agile Software Development Methods – Review and Analysis”. VTT Publications 478. <http://www.inf.vtt.fi/pdf/publications/2002/P478.pdf>> (accessed December 2005).
- [5] Abdul Rauf, Mohammed AlGhaffees “Gap Analysis between State of Practice & State of Art Practices in Agile Software Development “2015 Agile Conference 978-1-4673-7153-7/15 \$31.00 © 2015 IEEE DOI 10.1109/Agile.2015.21
- [6] Kupiainen, E., Mäntylä M. V., Itkonen J., "Using Metrics in Agile and Lean Software Development - A Systematic Literature Review of Industrial Studies", Information and Software Technology, Preprint submitted to Information and Software Technology February 10, 2015.
- [7] Rezvan Mahdavi-Hezave and Raman Ramsin In Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE-2015), pages 229-237 ISBN: 978-989-758-100-7
- [8] Jason Jen-Yen Chen* and Mike Mu-Zhe Wu “Integrating Extreme Programming with Software Engineering Education” MIPRO 2015, 25-29 May 2015, Opatija, Croatia
- [9] Rene Esteves Mariafile: “Applying Scrum in an Interdisciplinary Project using Big Data, Internet of Things, and Credit Cards” 2015 12th International Conference on Information Technology - New Generations 978-1-4799-8828-0/15 \$31.00 © 2015 IEEE DOI 10.1109/ITNG.2015.17
- [10] Gaurav Raj Asst. Komal Yadav ,Arunima Jaiswal “Emphasis on Testing Assimilation Using Cloud Computing for Improvised Agile SCRUM Framework” 2015 1st International conference on futuristic trend in computational analysis and knowledge management (ABLAZE 2015) 978-1-4799-8433-6/15/\$31.00©2015 IEEE
- [11] Nomi Baruah “Requirement Management in Agile Software Environment” The 2015 International Conference on Soft Computing and Software Engineering (SCSE 2015) Procedia Computer Science 62 (2015) 81 – 83
- [12] The case for DevOps in scientific applications Maximilien de Bayser, Leonardo G. Azevedo Renato Cerqueira 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)
- [13] Sriram Rajagopalan, Saji K. Mathew ”Choice of Agile Methodologies in Software Development: A Vendor Perspective” Journal of International Technology and Information Management Volume 25, Number 1 2016© International Information Management Association, Inc. 2016 , ISSN: 1543-5962-Printed Copy ISSN: 1941-6679-On-line Copy
- [14] Gourav kumar ,Pradep Kumar Bhatia “Impact of Agile Methodology on Software Development Process” SSN 2249-6343 International Journal of Computer Technology and Electronics Engineering (IJCTEE) Volume 2, Issue 4, August 2012
- [15] Jeffrey A. Livermore, “Factors that impact implementing an Agile Software Development Methodology”, pp. 82-85, IEEE 2007.
- [16] A. Ahmed, S. Ahmad, Dr. N. Ehsan, E. Mirza, S.Z. Sarwar, “Agile Software Development:Impact on Productivity and Quality” , pp. 287-290, IEEE 2010
- [17] 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
- [18] <https://stateofagile.versionone.com>
- [19] <https://newrelic.com/devops>