ISSN: 2321-7782 (Online)

Volume 3, Issue 4, April 2015

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Role of Agile Methodologies in Software Engineering

Sumera Kousar[1]
Departments of Computer Science & IT
University of Sargodha Women Campus.
Faisalabad, Pakistan

Salman Ali[2]
Departments of Computer Science & IT
University of Sargodha Women Campus.
Faisalabad, Pakistan

Abstract: Agile software development approaches is very important. This is due to the fastness that agile approaches take in the life cycle of software development. Agile development methodology is used for handling the change and we have discussed various model execution environment. Here we have used the concept of Extreme programming, Scrum and Agile modeling. A numerical measure of the insignificant supremacy of XP in excess of the design-led approach. Agile software development methods, without rationalization, cover different phases of the software development life-cycle and the majority of them do not suggest Inadequate support for project management. This paper explore the intra- organizational applicability and implication of open improvement in agile systems development. As well agile methodology is very important in the field of education. Agile Manufacturing processes based on (AM) are characterize by supplier, customer and incorporated process for product design, marketing, manufacturing, and support services. Based on this concept, a methodology for achieving agility in industrialized organizations is developed. Most software companies nowadays aim to produce valuable software in short time period with minimal costs, and within unstable, changing environments. Agile Methodologies were thus introduced to meet the new requirements of the software development companies. This paper presents a review of three agile approaches including Extreme Programming, Agile Modeling, and SCRUM, describes the differences between them and recommends when to use them.

Keywords: Agile software, Agile methods, Applications of agile methods, Extreme programming(XP), Scrum, Agile Manufacturing, Design-led approach, Support services,

I. Introduction

Programming advancement is a sorted out flourishes to convey items in speedier, better and less expensive ways. Recently, this investment has cleared path to another programming improvement system called Agile Software Development. To beat the quick changing hierarchical business needs utilizing conventional routines deft systems were presented. At whatever point another necessity comes into the picture it takes part of exertion regarding time and expense for examination and usage. Changing necessities from clients is making it significantly more difficult. Old programming improvement methodologies are not ready to fulfill the new prerequisites of the business sector in the most ideal way, any longer. Accordingly, new programming advancement methodologies are developed, as coordinated techniques, basically to take care of such issue. The new strategies incorporate alterations to programming improvement techniques, to make them more beneficial and adaptable. Exact Software Engineering (ESE) expects to give a thorough test methodology to examining these issues, however it is getting to be obvious that ESE itself is liable to the same weights brought on by fast advancement. Poaches, deft techniques are more adaptable regarding the matter of thinking seriously about prerequisites' progressions in all periods of programming improvement. Rather, light-footed methodologies ceaselessly include the client in the advancement process, probably prompting the improvement of a more creative and consequently more significant data framework. In lean manufacturing, the accentuation is on expense cutting. The prerequisite for associations and offices to wind up more adaptable and receptive to clients prompted the idea of the spry assembling as a separation from the &lean' organization a various scope of exploration has concentrated on the quick pattern of changes happening in the assembling field and the need of utilizing new dreams, and returning to the conventional rationalities and attitude about assembling business. Our paper is centered around the Agile programming advancement, nimble strategies focused on present practices in industry. Most normally utilized strategies will inspected from the plot of their pertinence, qualities and shortcomings and their reception in industry. So as to examine and break down, there is a need to think about the issues in the writing, exploration studies and industry. This will lead us to discover advantages, constraints and challenges on the move from conventional to dexterous programming improvement.

The Agile Techniques: Probably the most regularly utilized deft procedures are talked about as a part of this segment. There are a few parameters connected with the decision of these Agile systems, some of them are group size, emphasis length and backing for conveyed environment. These parameters are likewise talked about for these most regularly utilized coordinated methods here:

a) Extreme Programming (XP)

Five key standards of great Programming (XP) are correspondence, effortlessness, criticism, fearlessness, and quality work. Compelling writing computer programs is a decent light-footed methodology when the group size is by and large little i.e. from 2 to 10.iteration length is by and large short around 2 weeks. XP is not suitable for dispersed groups. XP gives a rundown of basic, particular, and apparently gullible standards and values that guide the product improvement prepare all through the fundamental four periods of programming advancement: arranging, coding, designing, and testing (Figure 1). The principle object is to convey what the client needs, at the time it is required. In expansion to this, one of the primary reasons of its prosperity is its capacity to acknowledge changes at whenever amid the improvement. XP additionally accentuates collaboration; encounters from all stakeholders are utilized to meet the particular objectives, and inside the given requirements.

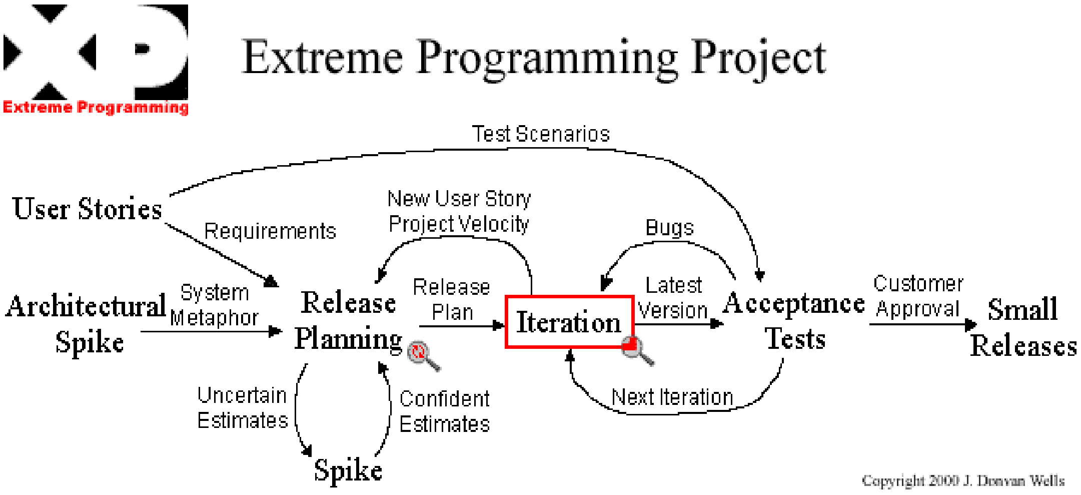

Fig. 1

This Agile approach enhances programming improvement in four ways. Firstly, high interchanges between the designers and their clients. Second, the effortlessness of outline. Third, the consistent change agreeing client's criticism, by conveying countless programming to the client. Fourth, empowering the client and designers to get to the right necessities by the create criticism cycle. Those four things are called XP values. There are some programming standards that are supported by XP. One of them is that, straightforwardness and adaptability will lessen upkeep expenses of the product later on. An alternate programming practice is the serious and strong testing component, which will lessen the quantity of bugs reported later by the client in the wake of conveying the last form of the application. Grasping changes is likewise supported by Xp since clients are destined to perceive new open doors for enhancing the framework to meet their goals, while still in the improvement process. In addition, XP supports making top notch code. Many different manages and practices are given by XP, for all the periods of programming advancement. These include: making client stories; making little continuous discharges; isolating the task into cycles; creating, spike answers for diminish hazard; accessibility of the client; composing code to concurred measures; pair programming; leaving enhancement until last; all the code is unit tried; making an experiment for each one bug discovered; just to name a couple.

b) Scrum

Scrum along with XP, is one of the more widely used Agile Methods. Scrum projects are split into iterations (sprints).Development team is divided up to 7 persons team. Iteration length varies from 1 week to 6 weeks. Project may consist of multiple teams that could be distributed. SCRUM methodology was initiated by Ken Swaber in 1995. It was practiced before the announcement of Agile Manifesto. Later, it was included into agile methodology since it has the same underlying concepts and rules of agile development. SCRUM has been used with the objective of simplifying project control through simple processes, easy to update documentation and higher team iteration over exhaustive documentation.

SCRUM shares the basic concepts and practices with the other agile methodologies, but it comprises project management as part of its practices. These practices guide the development team to find out the tasks at each development iteration. In addition to the practices defined for agility, one main mechanism recommended by SCRUM is to build a backlog. A backlog is a place where one can see all requirements pending for a project, sized based on complexity, days or some other unit of measure the team decides. Inside a product backlog, there is a simple sentence for each requirement; something that will be used by the team to start discussions and putting details of what is needed to be implemented by the team for that requirement. For the team of SCRUM, three main roles are defined as shown in Fig 2. The first role is the product owner, who mainly would be the voice of business. The second role is the SCRUM team which comprises developers, testers, and other roles. This team would make initial contact with customer and identify the need for a new product. SCRUM master, the third role, is responsible for keeping the team focused on the specific goals, and help the team members to solve problems when they appear.

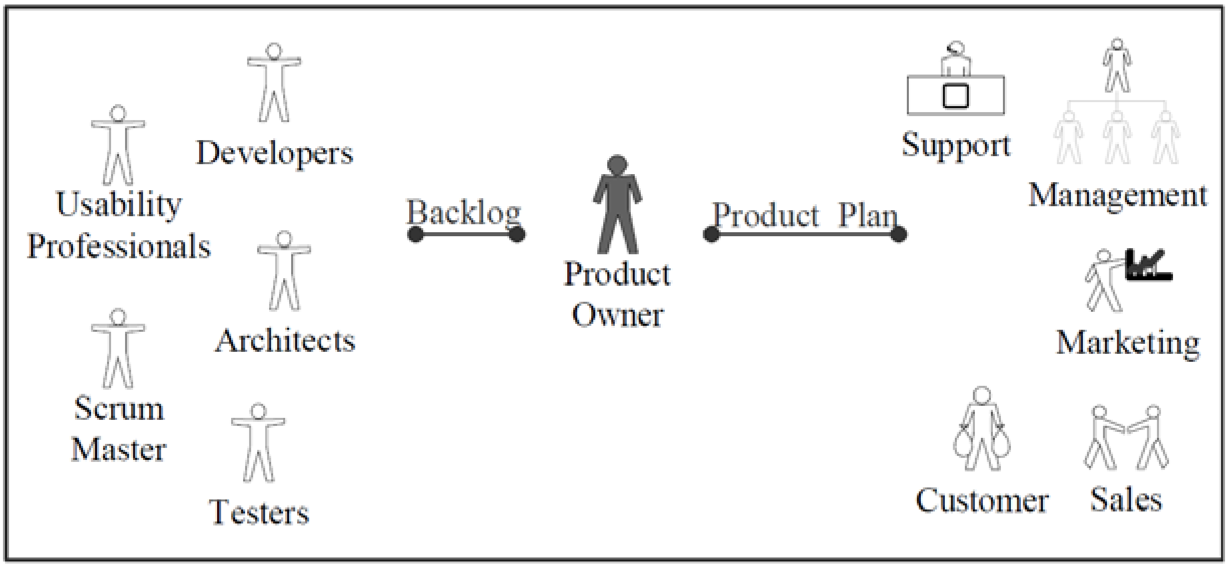

Fig. 2

The process of development using SCRUM divides the project into phases. In each phase, one feature is fully developed, tested, and become ready to go to production. The team does not move to a new phase until the current phase is completed. Whether what is being done adds value to the process or not, is the main concern of each phase. Current studies on traditional SCRUM development have shown that despites its advantages, it is not best suited for products where the focus is on usability. It fails to address usability needs of the user, because product owners keep their focus mainly on business issues and forget about usability. Since product owners usually come from business background, they lack the experience, skills, and motivation to design for user experiences. Moreover, traditional agile methodologies are not concerned about the user experience vision, which drives the architecture and is essential for ensuring a coherent set of user experiences. According to Mona Singh, U-SCRUM, is an agile methodology for promoting usability. Briefly, SCRUM is considered an iterative, incremental methodology of software development. It was proposed for software development projects, and at the same time, it can be used as a program management approach.

c) Crystal family of Methodology

Gem strategies are focused around the standard that how to attain to a greatest degree by which a composed correspondence or records correspondence can be lessened to a verbal correspondence for quicker advancement. All Crystal techniques start with a center set of parts, work items, strategies, and documentations. There is no restriction on group measure in precious stone routines.

d) Feature Driven Development

Characteristic Driven Development is the peculiarity arranged advancement approach that came to be known as FDD. FDD methodology is the most oversimplified improvement transform this works in three steps –

- Develop an advancement model
- Assemble a peculiarity list and
- Arrange by peculiarity. The group size changes according to the gimmicks unpredictability. Cycles lengths are dependent upon 2 weeks however it don't have help for dispersed frameworks.

e) Agile Methodologies

At the early years of programming improvement, the vast majority of the clients' prerequisites were genuinely steady, and advancement took after the arrangements without real changes. Notwithstanding, as programming advancement included more basic and element mechanical activities, new challenges developed by development of organizations.

These troubles include:

- Evolving prerequisites: client necessities are changing because of developing business needs or administrative issues. The majority of the clients don't have an acceptable vision about the determinations of their necessities at the early stages. A few clients acknowledge what their actual prerequisites are just when they utilize an application that does not so much address their needs. An alternate wellspring of progress originates from encounters increased amid the advancement.
- Customer inclusion: absence of client contribution prompts higher possibilities of undertaking failure. Many organizations generally don't assign any exertion for client association.
- Deadlines and plans: frequently, clients don't acknowledge disappointment. Then again, organizations typically offer low plans, tight due dates, while in the meantime, obliging levels of popularity, and the majority of this is a direct result of rivalry in the business sectors.
- Miscommunications: one reason for the misconception of necessities is the Miscommunication in the middle of designers and clients. Case in point, each one gathering uses its own particular language, and this prompts misconception of client's nece.

f) Agile Modeling (AM)

Demonstrating is an essential venture in programming advancement. It empowers programming engineers to consider complex issues before tending to them in programming. Deft Modeling (AM) was made by Scott Ambler in 2002. It is an accumulation of qualities, standards, and practices for demonstrating programming that can be connected on a product improvement extend in a powerful and light-weight way. Light-footed Modeling was constructed to be adjusted to, and utilized with existing philosophies, as XP and RUP, planning to permit a designer to construct a product framework that really meets the client's needs. The estimations of AM, which are thought to be an augmentation to the estimations of XP include: communication, simplicity, criticism, fearlessness, and modesty. Quietude intends to concede that you may not know everything; others may know things that you don't have a clue, and along these lines, they may give valuable commitment to the project. Again, the standards of AM are truly like those of XP, for example, expecting effortlessness, grasping changes, incremental change of the framework, and fast input. Notwithstanding these standards, AM standards incorporate the learning of the reason for demonstrating; having various powerful models; the Substance is more vital than the representation; keeping open and genuine correspondence between gatherings included in the advancement procedure; lastly, to spotlight on the nature of the work. The practices of AM have a few shared traits with those of XP, as well. A nimble modeler needs to take after these practices to make an effective model for the framework. AM practices highlight on dynamic stakeholder support; concentrate on gathering work to make the suitable models; apply the proper ancient rarity as UML charts; check the accuracy of the model, execute it and demonstrate the ensuing interface to the user;model in little additions; make a few models in parallel; apply displaying benchmarks; and different practices.

g) Agile Model Driven Development (AMDD)

AMDD is the nimble rendition of model driven improvement. To apply AMDD, a general abnormal state model for the entire framework is made at the early phase of the undertaking. Amid the improvement cycles, the displaying is executed as arranged every emphasis. Typically, AM is connected alongside different procedures, for example, Test Driven Development (TDD), and Extreme Programming (XP), to get the best results.

h) Other Agile Methodologies

In this paper, three fundamental coordinated approaches that have been generally utilized as a part of programming improvement are examined. Other than these three approaches, there are some other programming improvement philosophies lay under the deft umbrella. They incorporate Crystal approaches, Feature-Driven Development, and Versatile Software Development. Precious stone Methodologies were secured by Alistair Cockburn in 2000. They focus on proficiency and tenability as segments of venture security. Each of the Crystal approaches requires certain roles, policy gauges, items and apparatuses to be received. Clear as can be, which one of the Crystal philosophies is, can be connected to advancement groups of six to eight parts, chipping away at non-life basic frameworks. It concentrates on individuals, not techniques of artifacts. Feature Driven Development (FDD) was established by Jeff De Luca and Peter Coad. It consolidates a few practices perceived in the business into one system. These practices are all decided from a client valued usefulness (characteristic) perspective. As of other dexterous systems, its key objective is to convey tangible, working programming more than once in an auspicious way. Versatile Software Development (ASD) was made in 2000 by Jim High smith. It has become out of the Rapid Application Development (RAD). Like other light-footed procedures, ASD plans to expand a product association's responsiveness while diminishing improvement overhead. It encapsulates the conviction that persistent adjustment of the methodology to the work within reach is the ordinary condition of undertakings.

i) Limitations of Agile Methodologies

Dexterous improvement means to help early and brisk advancement of working code that addresses the needs of the client. Agile supporters assert that code is the main deliverable that matters, though, nimble adversaries found that accentuation on code will prompt memory misfortune, on the grounds that the measure of documentation and demonstrating done is not enough. There are a few restrictions to apply spry approaches. Initial one is that light-footed techniques are not suitable for green-field designing and not suitable for support, following there will be very little documentation for the frameworks. The second restriction is that spry approaches depend vigorously on the client association, and subsequently, the achievement of the undertaking will rely on upon the collaboration and correspondence of the client. An alternate restriction is that light-footed procedures focus work quality on the aptitudes and practices of the designers, as the configuration of the modules and submodules are made for the most part by single engineer. At the point when creating programming to be reusable, then agile strategies won't give the most ideal way. This is on account of the spotlight on building frameworks that tackle particular issues, and not the general ones.

Nimble approaches work best for groups with moderately little number of parts, and subsequently, they won't function admirably for groups with expansive number of parts. To get the favorable circumstances of applying spry strategies in the advancement, there is a situated of suspicions that are thought to be valid. To say some are: collaboration and up close and personal connection between the clients and the advancement group; developing and changing necessities of the undertaking; designers having great individual abilities and encounters; notwithstanding numerous more. If these suppositions don't have any significant bearing to a product advancement venture, then it is ideal to search for different approaches to request the improvement process, so as to improve results.

j) Identification of Agility Providers

A rundown of business practices, routines, instruments, and procedures, for the most part alluded to as deftness suppliers that could realize dexterity abilities for fabricating organizations are figured (Sharifi and Zhang, 1998b).

These incorporate both demonstrated instruments and practices that are as of now accessible to fabricating associations and in addition those which are as yet being produced by the exploration group. The essentialness of these dexterity suppliers (or the seen vitality of those suppliers still a work in progress) to different capacities is spoken to by an alternate system relationship where the association weight between a capacity and a supplier compares to the vitality of the supplier to the ability. The system takes as inputs the results from the ``capability system'' depicted before and produces a set of yields speaking to the imperativeness of individual deftness suppliers to an organization.

k) Implementation of the methodology

Despite the fact that the procedure was created focused around a survey of the writing relating to the subject and two experimental studies, it was seen as important to approve its pragmatic applicability. However, acceptance of the strategy is not an insignificant errand because of the accompanying impediments: It was not functional to completely inspect the strategy, considering the time period included in actualizing the proposed instruments and practices with the accessible assets for the exploration. It was genuinely hard to discover teammates for partaking in this stage, indeed in a shallow manner. The procedure in functional territories still needs upgrades, particularly in characterizing the relationship between readiness capacities and deftness rehearses.

II. Conclusion

Agile techniques incorporate a set of programming improvement approaches. They have a few varieties, yet at the same time they have the same essential ideas. The primary spry systems that are being utilized incorporate XP, Agile Modeling, and SCRUM. XP is the coding of what the client details, and the testing of that code.

Agile Modeling characterizes a gathering of qualities, standards, and practices which depict how to streamline displaying and documentation endeavors. SCRM, then again, underpins administration part in programming advancement. Spry systems are not best suited for all undertakings. At the point when correspondence between the engineer and the client is troublesome, or when the improvement group incorporates for the most part tenderfoots, deft techniques won't give the best results. These procedures display ideal results when there is a solid correspondence between the engineer and the client, and the advancement group bargains talented colleagues. At the point when there is a huge chance for misconception the precise client's necessities, or when the due dates and plans are tight, then Agile strategies are among the best programming advancement methodologies to apply.

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. Romain Robbes and Michele Lanza, "Change-based Approach to Software Evolution", Electronic Notes in Theoretical Computer Science (ENTCS) archive, Vol. 166 , Pages 93-109, (2007).
3. Agile Alliance. Manifesto for Agile Software Development. [Online] Retrieved 16th March 2009. Available at: http://www.agilemanifesto.org .
4. Agile Modeling Home Page. Effictive Practices for Modeling and Documentation. [Online] Retrieved 17th March 2009. Available at: www.agilemodeling.com .
5. K. Beck, and C. Andres, Extreme Programming Explained.Vol.159, Pages 122-230.
6. Ipate, F., Holcombe, M.: Generating test sets from non-deterministic stream X-machines. Formal Aspects of Computing 12.6 (2000) 443–458
7. Abrahamsson, P., Conboy, K. and Wang, X. "Lots Done, More To Do: The Current State of Agile Systems Development Research," European Journal of Information Systems 2009, 18, pp. 1-7.
8. Alleman, G. "Agile Project Management Methods for IT Projects." In the Story of Managing Projects: A Global, Cross-Disciplinary Collection of Perspectives, Greenwood Press, Berkeley, CA, (2002).
9. Abran, A., Moore, J. W., Bourque, P., & Dupuis, R. (2004). Guide to the software engineering body of knowledge. Los Alamitos, CA: IEEE Computer Society.
10. P. Bunce, P. Gould, From Lean to Agile Manufacturing, IEE Colloquium (Digest) (278) (1996) 3/1}3/5.
11. R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.
12. Somaieh Alavi, Dzuraidah Abd. Wahab, Norhamidi Muhamad, Behrooz Arbab Shirani. 2014. Organic structure and organizational learning as the main antecedents of workforce agility. International Journal of Production Research 52, 6273-6295.
13. Computer Programming. [Online] Retrieved 5th April 2009. Available at:http://en.wikipedia.org/wiki/Programming.

AUTHOR(S) PROFILE

Sumera Kousar, received the M.Sc degree in Computer Science from University of Agriculture Faisalabad in 2013.