

# Review of Agile Systems Development

AgwiUche Celestine<sup>1</sup> and Ogwueleka Francisca Nonyelum<sup>2</sup>

<sup>1,2</sup> Department of Computer Science, Nigeria Defence Academy, Kaduna, Nigeria.  
Email: ucheworld2015@gmail.com

**ABSTRACT:** Software development methodology refers to one or more techniques for building software. Agile software development describes a set of principles and practices that support adaptive planning, evolutionary development, early delivery, and continuous improvement of the software product, which encourages rapid and flexible response to change. This paper is a review of agile software development methodologies. Several literature materials on software development methodologies were reviewed. The paper provides an overview of agile methodology, types, strengths, weaknesses, and applications of agile development methodologies in software development projects.

**Keywords:** Extreme programming (XP), Scrum model, Feature Driven Development (FDD), Adaptive Software Development (ASD), Dynamic System Development Method (DSDM)



## 1. INTRODUCTION

Software development methodology is a model, a formalized approach used to plan, design, test and control the processes for developing software and information system. It is comprised of one or more techniques for building software [1]. How to deliver an effective, more efficient and a cheaper software solution has been a subject of serious discussion in the software development circle for decades. These discussions have elicited several solutions such as standardization, measurement of the software processes, development of tools, techniques, and practices. This drive to improve software development processes has immensely

influenced how software is now been developed worldwide [2].

Agile software development methodology (ASDM) is an outcome of the discussion held by some experienced software practitioners. This group wrote the manifesto for agile software development which included twelve key points [3], [4]. These key points were also referred to as the principles of agile software development and were summarized as;

- Place the highest priority on satisfying 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,

-----

- AgwiUche Celestine<sup>1</sup> is currently pursuing Doctorate degree program in Computer Science in Nigeria Defence Academy, Nigeria, PH+2348065235687. E-mail: ucheworld2015@gmail.com
- Ogwueleka Francisca Nonyelum<sup>2</sup> is Professor, Lecturing in Nigeria Defence Academy, Nigeria, Department of Computer Science.

- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale,
- Ensure business people and developers 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,
- Use the most efficient and effective method of conveying information to and within a development team is face-to-face conversation,
- Ensure that working software is the primary measure of progress,
- Ensure that agile methodology promotes sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely,
- Give Continuous attention to technical excellence and bearing in mind that 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-organizing teams and

- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.[3].

Agile Methodology is a software development methodology that builds software iteratively and incrementally so that the development is aligned with the changing business needs. Anupama and Malek, et al., described agile methodology as a framework that promotes an iterative, lightweight and lean software design and development methodology[6], [7].

Malik and Siew stated that the early years of software development witnessed relatively stable users' requirements[8]. That development proceeded according to the plans without major changes. But now the business environment has become more dynamic, software needs and requirements keep changing constantly. Therefore, the need to develop software that meets requirements for more critical and dynamic industrial projects have aroused and has led to the emergence of new difficulties and challenges. Turk, et al highlighted several problems affecting software development which included changing customer requirements due to evolving business needs or legislative issues, lack of customer's involvement in the development process, inadequate budget and deadlines which are not

met, and miscommunications between developers and customers resulting to misunderstanding of requirements and needs[9]. The existence of such problems has made software development become more complex, therefore, could not be handled by the traditional software development methodologies. As a result, new software development methodologies such as agile methodologies were evolved.

Agile software development describes a set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams [10]. It advocates adaptive planning, evolutionary development, early delivery, and continuous improvement, which encourages rapid and flexible response to change[3]. This paper is focused on the review of agile software development methodologies. The paper will provide an overview of the agile methodology, types, strengths, weaknesses and its applications as well as develop the readers' understanding of agile development methodologies in software development projects.

## 2. REVIEW OF RELATED LITERATURE

Agile software development represents a major departure from traditional, plan-based

approaches to software engineering [4]. Several literature studies have been carried out on various issues pertaining agile software development. However, from the review, it was found that the various literature touches different aspect of agile software development. This is an indication that there is still room for further researches.

Mihaidid a comparative study of some software development methodologies. In this study, specific characteristics in software development methodologies were highlighted [11]. Current software development methodologies were presented with a graphic illustration in order to better show its structure. Software development was classified into heavyweight and lightweight development approaches. The heavyweight was identified to be easy to understand and implement because they are well structured and provide tangible deliverables. While the lightweight are better alternatives when it comes to handling projects where specifications are unclear and dynamic due to certain internal or external factors.

Tore and Torgeir carried out a systematic review of agile software development[4]. The review investigated the benefits and limitations of the agile methodologies. Kaushal and Anjuprovided a review of some different agile methodologies and how they are divergent from conventional process methods, the pros, and cons for

applying agile processes to the research projects as well as difficulties faced by enforcement of agile methodologies in the project[2]. It further recommended that future research should include a review of lean software development (LSD) and Dynamic System Development method.

Stavru[12] surveyed the use of agile methods through industrial survey studies published between 2011 and 2012. They determined the papers, which could be trusted and recommend that the quality level of research 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. The recommendations include examining the rate of agile method usage as compared to alternative methods; examine the rate of agile method usage at an organizational level. Also, conducting researches about using an agile method by academics besides industry in order to decrease the gap between industry and academia in addition to increase the trustworthiness in the widespread adoption of agile method usage. Additionally, providing highly detailed reports in the future; thus, raising the level of confidence and trustworthiness in the reported studies.

In a related research, [13] suggested that the differences between traditional methodologies and Agile Methodologies rely on two main assumptions: First, traditional methodologies assume that customers do not know their requirements, hence they need guidance from the developers, but Agile Methodologies assume that both customers and developers do not have full understanding of requirements when the project starts. Therefore, in traditional software development environments, developers want a detailed specification, whereas in Agile Methodologies customers and developers learn together about the system requirements as the development process evolves. Second, traditional methodologies assume that customers' ability to foresee their future requirements is limited, and as such developers have to build in extra functionalities to meet these future needs, often leading to an over designed system. On the other hand, Agile Methodologies emphasizes simplicity. The outline of Agile Methodologies was laid down by the Agile Manifesto, published by a group of software practitioners [3].

### 3. AGILE METHODOLOGIES

Juyun[14] described agile software development methodology as one that allows iterative and incremental development, adaptability throughout the systems development life cycle, minimal planning, light and fast development

cycles, people-centric development, customer collaboration, and frequent delivery. Agile methodologies for different types of application development include; Extreme Programming (XP), Scrum, Feature Driven Development (FDD), Adaptive Software Development (ASD), Dynamic System Development Method (DSDM), Crystal Clear, Lean software development etc. Each methodology is concerned with best practices in order to achieve maximum benefit.

### 3.1 Extreme Programming (XP)

Extreme programming (XP) was introduced by Kent Beck in 2000. Beck's idea was to develop a programming technique that will be rooted on some sets of values, principles, and practices. XP is an agile methodology which encourages iterative and incremental software development. Extreme programming emphasized practices which include simplicity of design, collaborative programming, continuous improvement of codes and continuous testing of software before final release[15]. The goal of agile methodology in general and XP, in particular, is to ensure that quality software is delivered to customers as quickly as possible and then evolving it to meet changing business needs [8]. Extreme programming comprises of four phases which include planning by creating user stories, project schedule and planning iterations; designing for simplicity and choosing a metaphor; coding

which involves translating plans and designs into system Units and continuous test of system units and integration.

XP encourages values such as communication between customers and among programmers, simplicity of design, continuous feedback from customer through customer participation in the software development process and courage [16].

XP is characterized by the quick release of working software, incorporation of additional requirements while development process is ongoing, user participation in the software development process, continuous testing, and suitability for small team size. However, it is limited by its non-suitability for large and complex projects such as mission critical software projects. Also, it is not suitable for large team of developers as communication among them becomes difficult [15].

### 3.2 Scrum model

Scrum methodology was introduced by Ken Schwaber in 1995 [15], [8]. Scrum is a software development process that quickly gets the software product to the customer. It is based on empirical control through inspection and adaptation [17]. The basic idea about SCRUM methodology is same as other agile methodologies. However, project management practices are incorporated into scrum methodology. These practices are useful in

helping the team find out the various tasks at each stage of development [8]. Scrum methodology adopts the principle in which small teams work cross functionally towards producing good results. Scrum is focused on improving revenue and quality of the software hence, it is said to be revenue-centric [2]. Scrum allows changing requirements to be incorporated into the software development process while development is ongoing and then releases a working software in short cycles called sprints. Three important factors in scrum are the product owner, scrum master and team. The product owner is one who specifies the various features of the software, sets priorities and release date of the software. He acts as the owner of the business. The scrum master makes ensures that the team is function and collaborate effectively and successfully. He supports the team members to solve problems when they appear while the team is comprised of developers, testers, and other roles. The team would make initial contact with customer and identify the need for a new product [8].

Scrum methodology is based on the sprints, daily scrum meetings and sprint planning [15]. Sprints are small functionality that is developed within 30 days [15]. The Daily Scrum meeting is initiated by the scrum master to reviews the work that is done regarding development. This meeting usually lasts for is a fifteen minutes

session [2]. The sprint planning meeting is between the customer and the team. The product owner prepares the product backlog which has a list of features of the product including functionality and technical architecture [19]. Scrum model resolves the problems and complexities of each team members in daily scrum meeting [15]. Scrum methodology has advantages which include Short time duration, daily scrum meeting, user involvement, review sprints, focuses on team communication and can work with any technology. However, it also has some disadvantages which include more suitable only for small sized application, employs skilled and experienced team [2]. 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 [8], [20].

### **3.3 Feature Driven Development (FDD)**

Kaushal and Anju stated that FDD is different from other agile development methodologies because it places emphasis more on planning and upfront designing[2]. They further stated that FDD approach is focused on the software features of the system which is the main driver of the entire Feature driven development methodology. In FDD features decomposition technique is used. Features are small client valued functions. A project is broken down into

small Units which are handled by a team. [15]. The feature is developed within few hours or days, but no longer than two weeks. Using FDD, development teams are divided according to design and to implement a particular feature. The development work is performed in parallel on all the features. Each team is headed by a feature owner, who is responsible for the code segment that implements those features [2]. The emphasis in FDD is on the activities that guarantee quality assurance hence inspections, reviews, walkthrough etc. are performed [15]. FDD has advantages such as; assurance of the product quality, costing as per feature, collaboration among team members, and involvement of the users in the development process. However, it is not suitable for projects that are small in size. It is complex to understand and implement. In addition, it requires a lot of time. [2], [15].

### **3.4 Adaptive Software Development (ASD)**

Adaptive Software Development (ASD) was created in 2000 by Jim Highsmith. It has evolved from the Rapid Application Development (RAD) methodology. The goal of ASD is to increase software organization's responsiveness while decreasing development overhead [21]. ASD is a risk driven approach suitable for handling complex projects. It starts risk assessment as early as possible so as to be able

to determine the chances of success of such project. ASD emphasis is on the human interaction and team self-organization [15]. The activities in ASD are classified into three phases; the speculation phase, the collaboration phase, and the learning phase. In speculation phase, planning for what to achieve in each development cycle is made. In the collaboration phase focus is on how customer and team members interact to share their knowledge. The technique allows team members to take certain decisions rather than going through the whole process of getting permission through formal method from the higher authorities. The learning phase emphasizes learning of individual team members in order to develop a project in more efficient manner. Learning phase concentrates on the focus group, technical reviews and project post-mortems. ASD has the advantages of encouraging risk driven development, team collaboration, its focus on individual team members learning, quality reviews and time boxing. However, its disadvantages include over dependence on inter human collaboration, not been scalable [15], [8].

### **3.5 Dynamic System Development Method (DSDM)**

The Dynamic Systems Development Method (DSDM) is a framework for developing software in an agile way [18]. DSDM was first created in 1995 by a consortium that wanted to explore



different ways to develop software. In 2008, the DSDM consortium released DSDM ATERN 4.2 which contains much of the original method but incorporates elements of modern software development [5]. DSDM subscribes to many agile principles like active user involvement, empowered teams and frequent delivery of products. However, more focus is on the techniques used in DSDM development which include MoSCoW (Must have, Should have, Could have and Want to have) prioritization, prototyping which has to do with creation of prototypes of the system under development at an early stage of the project, facilitated workshops which aims at bringing the different stakeholders of the project together to discuss requirements, functionalities and mutual understanding. In a workshop, the stakeholders come together and discuss the project and Time

boxing which is a set interval for software development iterations after which deliverables have to be presented, preferably in the form of working software. DSDM eliminates the problems of missing deadlines, going over budget, management not committed, frequent release, and time boxing [15].

#### 4. COMPARISON OF METHODOLOGIES

Table 1 compares five agile development methodologies, Extreme Programming, Scrum Model, Features Driven Development, Adaptive System Development and Dynamic system Development Methodology that were discussed in this paper. These five methodologies were compared against eight criteria [15].



Table 1: Comparison of some AgileMethodologies (Source: [15])

Methodology	Extreme Programming (XP)	Scrum	Feature Development Methodology (FDD)	Adaptive Software Development (ASD)	Dynamic System Development Method (DSDM)
<b>Application Size</b>	Suitable for Small-Medium	Suitable for Small- medium	Suitable for Medium - large	Suitable for Small- Large	Suitable for Small- Large
<b>Known Requirements Initially</b>	Involves requirements during the development	Involves requirements during the development	Involves requirements during the development	Involves requirements during the development	Involves requirements during the development
<b>Risk and Security</b>	Risk reduction via test driven development and reviews	Risk is reduced through sprint reviews	Risk removal is through milestones	Risk removal is through continues risk updating	Risk removal Is through continues risk updating and testing
<b>Complexity</b>	Risk removal Is through continues risk updating and testing	Resolves Complexity	Resolves Complexity	Resolves Complexity	Resolves Complexity
<b>Quality</b>	Good Quality	Good Quality	High quality	High quality	Good Quality
<b>Flexibility to Change</b>	More Flexible	Flexible	Flexible	More Flexible	More Flexible
<b>Time</b>	Delivers software product within Short Time Duration	Delivers software product within Short Time Duration	Takes a Long Time Duration to Delivers software product	Delivery time of software product is according to Application Size	Delivery time of software product is according to Application Size
<b>Cost</b>	Low Cost	Low Cost	According to Application size	According to Application needs	According to Application needs

#### 4.1 Limitations of Agile Methodology

Generally, agile development methodologies have brought lots benefit to software development. In spite of these enormous benefits, there are still some limitations to the general applicability of agile development methodology. Such limitations include;

- Agile development methodology has limited support for distributed development environments as a result of face-to-face communication which is highly emphasized in Agile Methodology
- Agile development methodology is most suitable for a small team, not more than twelve. Therefore, for a larger team agile support become limited.
- Agile methodologies have limited support for Safety-critical software. The safety-critical software is software in which failure can result in direct injury to humans or cause severe economic damage. The quality control mechanisms supported by current agile processes (e.g., informal reviews, pair programming) have not been proven to be adequate to assure users that the product is safe.
- Agile development methodology has limited support for developing large, complex software

#### 4.2 Benefits of Agile Methodology

Agile software development offer both the developers and the users of the software products a handful of benefits which include;

- i. the saving of time and money
- ii. less documentation required as the technique focuses more on application rather than documentation
- iii. regular feedback are gotten from the end user so that the same can be implemented as early as possible
- iv. With the transparency afforded by agile development projects, customers have witnessed stronger results and have benefitted from being provided with real-time updates on the status of development
- v. agile method, offer development teams a lightweight process that supports a focus on the rapid delivery of business value[22].

#### 5. RESULT OF THE REVIEW

Agile Software Development has brought many good things to software development. The most intuitive are the improved quality of products, improved efficiency of developers, fewer errors and timely delivery of a working software product. Agile is considered one of the most popular software design and development

methodologies. Agile methodologies include a set of software development approaches which have some variations but still share the same basic concepts. This study examined five agile methodologies were studied and compared based on certain criteria. This will help the developer decide which of these software

development methodologies should be suitable for that particular software application. This paper will in no doubt develop the readers' understanding of agile development methodologies in software development projects.

## REFERENCES

- [1] Rajagopalan, S. and Mathew, S. K. (2016) Choice of Agile Methodologies in Software Development: A Vendor Perspective Journal of International Technology and Information Management Vol. 25, No. 1, Pp 39 - 54
- [2] Kaushal P. and Anju S. (2013) "Review of Agile Software Development Methodologies" International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 2. Pp 270-276
- [3] Beck, K., Beedle, M., A. van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. and Thomas, D. (2001) "The Agile Manifesto," <http://www.agileAlliance.org>.
- [4] Tore D. and Torgeir D. (2008) "Empirical studies of agile software development: A systematic review" Elsevier. Information and Software Technology, Vol. 3, Issue 11, Pp. 833–859 [www.ijert.org](http://www.ijert.org)
- [5] DSDM Consortium. (2012). DSDM Atern: the Handbook. DSDM Consortium. London. Pp. 3-56
- [6] Anupama K (2016) "A Literature Review on Agile Software Development". International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified. Vol. 5, Issue 9. Pp 337-339
- [7] Malek, A., Mariam, B., Wael, E. and Adnan, S. (2017) "Agile Software Development Methodologies: Survey of Surveys". Journal of Computer and Communications. Vol 5, No?, Pp. 74-97 <http://www.scirp.org/journal/jcc>
- [8] Malik, H. and Siew, H. (2009) "Review Of Agile Methodologies in Software Development" International Journal of Research and Reviews in Applied Sciences Vol. 1, Issue 1, Pp 1-8
- [9] Turk, D., France, R. and Rumpe, B. (2002) Limitations of agile software processes. In

- Proceedings of the Third International Conference on Extreme Programming and Flexible Processes in Software Engineering.
- [10] Collier, K. W. (2011). Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing. Pearson Education. pp. 121
- [11] Mihai, L. (2014) "Comparative Study of Software development methodologies". Database Systems Journal. Vol. 5, No. 3, Pp 37
- [12] Stavru, S. (2014) "A Critical Examination of Recent Industrial Surveys on Agile Method Usage". Journal of Systems and Software, 94, pp. 87-97.  
<https://doi.org/10.1016/j.jss.2014.03.041>
- [13] Highsmith (2001), "Agile software development: the business of innovation", IEEE Explore Pp. 120 - 122
- [14] Juyun, J. C. (2010) "An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum". Thesis Utah State University (Unpublished)
- [15] Neetu, K. L. and Pankaj, D. (2014) "Software Development Process and Methodologies: A Review". International Journal of Engineering Research and Technology (IJERT) Vol. 3, Issue 11, Pp. 1338-1341
- [16] Ramya, K. T.S., Phani, K. C., Phani, K. C.V and Vamsi, K. T. V (2011) Survey on Extreme Programming in Software Engineering International Journal of Computer Trends and Technology. vol. 2 Issue 2 Pp. 21-24.  
<http://www.internationaljournalsr.org>
- [17] Advance development methods (2003) Scrum Methodology: incremental, Iterative, Software Development from Agile Process. Advance development methods, Inc Pp. 15-40
- [18] Aydal E.G. (2005). "Extreme programming and Refactoring for Building Secure Web based Applications and Web Services", MSc Project, University of York. (Unpublished)
- [19] Cao, D.B. (2006) "An Empirical Investigation of Critical Success Factors in Agile Software Development Projects", Ph.D. Thesis, Capella University, USA.(unpublished)
- [20] Singh M. (2008) "U-SCRUM: An Agile Methodology for Promoting Usability". In Ag. AGILE '08. Conference, Toronto, 555-560.
- [21] Maurer, F. and Martel, S. (2002) "Extreme programming. Rapid development for Web-based applications". In Internet Computing, IEEE, Vol.6 No.1, 86-90.
- [22] Radha, S., Renuka, P., Mantha, S. S. and Vinaya, B. (2012) "Agile Methodology Adoption: Benefits and Constraints" International Journal of Computer Applications. Vol. 58, No.15, Pp. 31-37