Article

Performance Assessment of Agile Methods

Kardan Journal of Engineering and Technology
1 (1) 11–22

Kardan Publications
Kabul, Afghanistan
https://kardan.edu.af/Research/Kardan_journal_of_engineering_and_technology.aspx#

Muhammad Tahir
Jamshaid Khan
Zabiullah Atal

Abstract

With the passage of time, there are numerous agile software developed all around the globe contributing to the development of new and improved software methods. This study analyzed and compared three different agile software methods, which can be used by the management of different organizations. The arguments of different scholars have been used for presenting the analysis and overcoming the responsibilities of critical review. The discussion is on three different software agile methods, and software tools, which can be used by the management of organizations through comparative analysis. The study identifies the best software which can be used based on critical discussion.

Keywords: *Kanban, DSDM, FDD, Agile Methods, Performance Review of Agile Methods*

Mr. Muhammad Tahir, Mr. Jamshid Khan and Mr. Zabiullah Atal, *Assistant Professor, Faculty of Engineering and Technology, Kardan University, Kabul, Afghanistan.*

Introduction

The field of software development and agile development has significantly developed since the last decade, and a large number of various approaches related to software development have been observed. A recent study by Abrahamsson and his partners showed that in the last 25 years, a wide variety of different approaches related to software development have been introduced, with only a few of them surviving in today's world [1]. It is also argued that traditional information systems and their methods behind the development are primarily treated as a basic necessity for presenting an image of control towards symbolic status. However, these methods are too mechanistic and should not be used in details due to which this research tries to evaluate some of the best software agile management methods, which can be used by the organizations to provide the background and critical review of software agile methods.

1.1 Background

Agile is called as "the quality of being agile", by adopting to readiness, nimbleness, dexterity and activity in motion. The companies are not asking for more agile software methods which can answer the eager business community, who wants to automate their software development process [2]. According to the latest researchers, new agile methods have evolved rapidly and substantial amount of literature and debate is offered by the authors. However, academic research is still scarce on this topic and most of the publications are presented by practitioners or consultants. While Agile and Lean software development process remains scarce, this study aims to present the current software development process in different industries and draws important conclusions on that.

1.2 Research Questions

This research aims to answer the following question:

- What are the best agile software currently available in the market?
- Based on the assessment of new software agile methods, which one of them is best and should be used?

2 Literature Review

2.1 Agile Software Development

Agile software development is now used by the organisations as an alternative to plan-driven and often outclassing heavy weight machines. Agile methods allow sharing of common values and principles [3]. Agile methods are constantly developing with a rapid pace and there are various methodologies which can be used by the firms and developed to overcome the problems of driving high productivity rate. Within today's fast-paced

business communities, the businesses are constantly required to adapt to rising challenges. In a study, it was argued that companies that want to increase their productivity and producing the software should definitely turn towards the agile development practices in order to become competitive [4]. Agile processes in software organisations provide a platform to increase collaborations and iterations and these all result in rapid delivery of the software products. Among many examples, pair programming has shown slow speed among teams, while agile practices lead to significantly high productivity.

Agile software development provides a plan-driven approach and an alternative approach to heavyweight machines methods. According to the study of Brhel, and his partners, the agile software development method assists in sharing common goals and values. They allow interactions, working software methods, collaboration with customers and positively responding to the changes [2]. However, another study argued that agile development focuses on short development cycles, continuous face-to-face communication, frequent delivery systems and learning platform, which remains a challenge.

Some of the popular agile development methods in software domain include Scrum, Lean Software Development, Extreme Programming and Kanban with many others in the development [4]. Scrum is a widely popular method in agile software development framework and is primarily developed to induce complex systems and products in a workplace. According to the study of Cooper and his partners, Scrum is normally used for the development of an interactive approach in the workplace and to control risk by controlling predictability [5]. Arguing with the statements, Abrahamsson and his partners argued that mostly three people are involved in the development process and they are the product owners, scrum master, the project team and scrum team [1]. Therefore, it is important that the process and standards of scrum process are properly followed by the organisations.

There is a whole process involved in scrum process and it normally starts with the product owner. First, the software owner is required to collect data from customers, stakeholders etc. and compile a list of features according to the priorities of the project managers and owners [3]. However, this compilation should be in the form of a sequence and completely complying with business requirements, the available resources and keeping it alive for the product owners [6]. If the product owners do not constantly revise the backlog of product, the chances of losing competitive advantage after some time would significantly increase. From

the backlog of the product, the teams should be able to look and pull out Spring backlog, which happens for different types of the sprint. When a team of an organization find spring backlog, it should run these building blocks in a sequence of sprints [7]. However, this Spring backlog should continue for a fixed time period after mutual understanding of the team members and product owners.

In case some emergencies are faced during the project, the product owner can directly order the team to stop this scrum and start a new one. Most of the studies argued that scrum normally depends on 5-9 people, and this team works independently [8]. Nonetheless, this team should be carefully developed after critical evaluation of the project needs and objectives along with the self-organization and self-management to be in the form of a cross-functional approach [8]. They also added on the argument that teams must select what delivery they would make by day end, and at the Sprit. It is one of the crucial factors for team members to take part in the team activities equally regardless of the experience level.

In the scrum process, there are many success factors which contribute to project success or failure, one of which is project meetings. Dingsoyr and his partners argued that teams must compile a chart named burn down chart, which should highlight the tasks and the total hours left to complete that task [9]. However, Matharu argued that team meetings must be carried out daily for 15 minutes or less, where each member stands and report things he has done in the project [10]. In this project meeting he explains what he did yesterday, what he would be doing tomorrow and the blocks and obstacles he would face [10]. On this occasion, it is the responsibility of the scrum master to observe the challenges reported by that team member and provide him with the guidance.

It is the responsibility of the team leader to protect and lead his team, by protecting and serving them. In a study, it is argued that if scrum master support is not there, the team is at a high risk of failure and the aim of this team is to deliver 100% of what they committed and it should be potentially shippable for the project [1]. Each shippable component must be implemented and tested with no major deficiencies and 2 reviews should take place at the end of each phase. Adding on the argument, they argued that at the end of each sprint, a 'sprint review' meeting should take place, in which the product owner, scrum master and stakeholder should be involved [4].

Another type of meeting is called a Retrospective meeting, which is attended by team, scrum master and the product owner. According to the arguments of Anwer and his partners, this meeting takes place at the end of

each sprint to check their approach and adapt new ways if needed [6]. Sprint review is also a product review while Retrospective approach is a process review. A study was also found arguing that Extreme Programming is a popular method of agile practices for the development of high-quality software because it focusses and concentrates on leveraging the quality of software [1]. The methodology adopted during the agile software development practices is normally depending on addressing the engineering practices and taking them to the extreme levels. However, the research article of Brhel argued that code reviews during scrum process must be taken because it is a good practice and it can be reviewed continuously according to the practices of pair programming [2]. Therefore, it can be argued that most of the studies are willing to support scrum process, however, there is a critical need to identify the drawbacks of these software development methodologies.

3 Methodology

The research methodology is considered one of the most important parts of a research study and demands that an adequate amount of literature should be consulted for collecting data. In order to identify methodology, some important methods of data collection and analyses should be identified according to the requirements of the research paper.

3.1 Research Design

The research design is considered an important part of this study because there are many types of research designs, and identifying the best one of them is necessary. The design of this study if descriptive because this study is based on collecting the literature from Google Scholar website and presenting it in the form of critical arguments. The descriptive approach allows the understanding of the results in the form of critical review approach and presenting them in the form of arguments. This study can also be called the study of arguments because it critically evaluates the role of all the software development agile methodologies and presents them critically and individually.

3.2 Data Collection

Data collection methods allow a reader to look into the methods adopted for the collection of data, and to analyse the relationship between conceptual framework and its variables. In order to collect data, Google Scholar website was used by applying various keywords, and selecting the best research articles for data collection. Data was collected from the year range between 2015-2019 and it provided some scarce research papers. Out of those research articles, the one matching with the topic was selected

only. The data is collected and compared among the three software agile methods, which are:

1. Feature Drive Development
2. Dynamic software development method
3. Kanban

3.3 Data Analyses

A number of methods are available to the scholars for analyzing data, and collecting data according to the needs of the research. This research made use of critical review approach for collecting the data and used a high number of research journal articles to compare and contrast various agile methods of software development. The research only selected and analyzed the research articles, which were available in the PDF format and with full access.

3.4 Research Sample Size and Sampling Method

Many types of sampling methods are available, but an appropriate selection should be made when carrying out the research. The research sample is also called an important component of a research paper because it allows the understanding of sample selection method. The research sample is based on collecting the data from Google Scholar website, and because of that convenient sampling methodology was adopted. Only the research article addressing the research topic was selected and sampling method deployed was convenience sampling method.

3.5 Ethical Considerations

Ethical considerations are rated the most important aspect when conducting research because they can significantly contribute to the valid arguments and results in a research paper. One of the ethical considerations which is taken by the researcher in this investigation is selecting the research article from the Google Scholar website only, and the published journal articles only. In addition, the articles are selected based on the topic within the last five years' period.

3.6 Research Validity and Reliability

The validity of the research depends on several characteristics and approaches which are undertaken by the scholars. A method adopted in this study to increase research validity and reliability is by selecting the research articles from Google Scholar only, and reliability is increased by selecting the most appropriate articles within the last 5 years only. The reason to select the articles from the last five years is that it would increase the importance of this research and the latest developments in the agile methodologies.

4 Results and Discussion

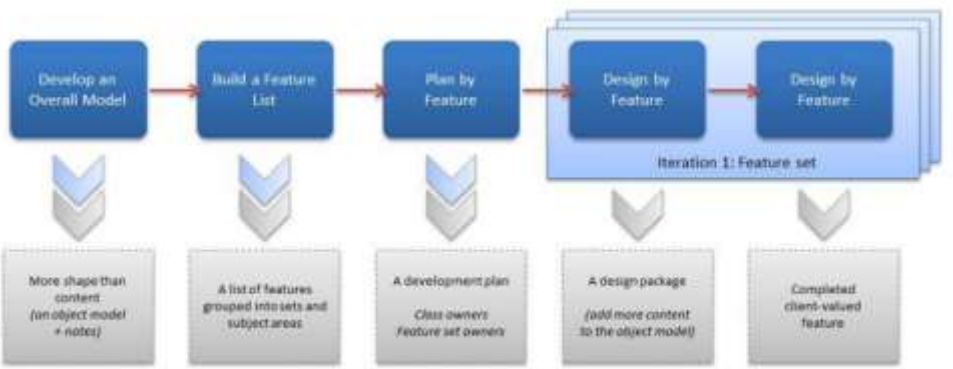
4.1 Overview

This section provides the discussion on several agile methodologies and concludes with the comparison chart of the three selected agile software development methodologies:

4.2 Feature Driven Development (FDD)

FDD is also an agile software development method, which is based on using a short-term iterative process and incremental processing. According to a study, this method is for those organisations that want to develop software through lightweight methods adoption [1]. Combination of the key benefits can be achieved by using this method of agile software development which is rated as the best method to meet the needs of larger projects and team. However, it is also argued that its features are small valued by the organisations, and it normally depends on a given time period which can be broken into small features [11]. The steps of this method can be found in the figure below:

Figure 1 Feature Driven Development Approach ([12]



The following steps should be followed by the teams comprising; project manager, chief architect, domain experts, chief programmers, Development manager and class owners:

Step 1: Come out with a model by mutual understanding of all the project team members [7].

Step 2: Compile a list of features which must be organized according to the hierarchy of business activity, and features to be included in software [6].

Step 3: Always plan by feature. Research argued that the construction of initial variables should take place, and responsibilities should be assigned to the team members [2].

Step 4: Designing by feature now should take place by critically evaluating the team features, collaboration to date and updating the artefacts for supporting the changes [8].

Step 5: Building feature should take place that should be followed by implementation of design features and mandating the code inspections [12].

One of the research articles was found arguing that FDD is not an ideal methodology when completing the smaller projects and for an individual software developer [1]. However, Papadopoulos argued that this method of software development allows the working of teams in parallel [3]. However, no one was found supporting the argument. This method of software development depends on the main developer, which means that the responsible team should be equipped with an ideal methodology of coordination to allow this methodology to work for this organisations.

4.3 Dynamic Software Development Method (DSDM)

This method of software development also comes under an agile project delivery method and is a RAD approach towards software development with increased customer involvement. In an investigation presented by Silva and his partners, the authors argued that DSDM became a platform for project development and a solution delivery method because it started allowing the organisations with industry standards to develop software tools [11]. However, the principals involved in DSDM are not easy to follow by a new organisations. Some recommended techniques by some studies when applying the DSDM approach includes MoSCoW, Prototyping and configuration management etc. [4]. Following are some of the principles, which must be considered by the organisations while using this approach:

1. Always focus on addressing customer needs.
2. Induce active involvement of the users
3. Emphasis on frequent releases rather than heavy focus on quality.
4. If there is a need to re-correct a solution, and always adopt the iterative development approach.
5. Perform testing on the product regularly.
6. Involve each stakeholder and communicate to meet their needs.

According to the research article of [6], if the following steps are not followed in this approach, chances of losing the valuable customers and poor software sales(**will**) significantly increase:

Figure 2: Steps in DSDM ([7]

Source: [7]

The first step is conducting a feasibility study followed by the modelling of function, designing and building iteration, and then implementing. Nonetheless, a study found that these steps are too generic in nature and demand a critical review of the methodologies which will be used for completing this project [3]. Supporting the argument, [10] wrote that most of the companies develop their own criteria when developing this software, which sometimes results in the development of inadequate software due to lack of depth. Hence, it can be argued that this method should be critically analyzed by the companies before using, otherwise chances to losing its validity would significantly increase.

4.4 Kanban

This method of agile software development is commonly used for managing the work and ensure timely delivery of software development. This method is supported by many studies because its development process is easy to follow and transparent [11] [13]. In addition, this method depends on an approach which is incremental and aims to bring revolutionary changes in the systems of an organisations. Nonetheless, [14] argued that this method uses work-in progress approach which is a big challenge for the organisations. Adding on the argument, visualization is required in this approach because it assists in understanding the workflow in software development.

Criticizing this approach, the study of [15] argued that this method lacks in providing a specific set of roles unlike DSDM. This method initiates with the deployment of activities based on existing roles followed by continuous incremental and evolutionary changes in software development methods

[9]. The Kanban method is also named as change management method in an organisations by some scholars because it demands evolutionary changes in a firm [16]. This method of agile software development is based on respecting the current procedures, roles and responsibilities which a company is facing, but old systems must be changed when using this approach.

In another research article, it is written that an organisations can use the Kanban method to resolve the issue of resistance faced by the workers, especially when it comes to retaining the workers for a long time period. Supporting the statement, [5] wrote that the Kanban method seeks to eliminate the initial fears from the workers involved in software development by providing them support in current roles and responsibilities. However, a research article wrote that Kanban method demands leadership inducted at all levels, coming from individuals to board members [2]. Therefore, it is important to include all the people when using this approach to ensure effective implementation of the software.

4.5 Comparison of All the Three Methodologies

The table below provides the comparison of all the agile software development methodologies in summary form:

Table 1: Comparison of All Three Methodologies (Self-Created)

Development Approaches	FDD	DSDM	Kanban
Time period required to perform an iteration	Greater than 2 weeks	Normally 2-4 days	Scheduled time based on the agreement.
Size of the team	Depends on the management	No size limit	Development of skilled or cross-functional team.
Size of the project	Complex projects	Each project	Small projects with LOC less than 3000.
Communication between the teams	Written and documented communication.	Written and documented communication.	Informal communication.
Involvement of customers	Reports formation	Very frequent	Communication through incremental release.

Source: Author compilation

5 Conclusion

This research investigated various methods behind agile software development and the ways most of the organisations utilize them. The research was based on following a critical review methodology and selecting the research articles from Google Scholar website. The method of

critical review approach was selected based on high level of available literature on this topic. After critical review carried out, it was found that there are three methods of agile software development, which can be used in different situations by the organisations. DSDM is recommended to be used by the management when the issue of change management is arising, while the methodology of FDD is used when a highly complex project is needed to be developed. In addition, Kanban can be used when small software is needed to be developed.

However, many scholars were found criticizing these approaches, because of their several drawbacks. For example, DSDM is not recommended when low interaction with the customers is required. It is also found that most of the studies have not evaluated both the benefits and drawbacks of these software methodologies and more investigation is required.

6 Future Scope

The future studies should investigate and compare more agile software development methodologies developing at a rapid pace. People using software packages should be investigated through the approach of interviews in future studies, especially in developed nations. The benefits and drawbacks should be investigated with all the stakeholders and to evaluate their performance.

References

[1] P. S. O. R. J. a. W. J. Abrahamsson, "Agile software development methods: Review and analysis," *arXiv preprint arXiv:1709.08439.*, pp. 1-9, 2017.

[2] M. e. a. Brhel, "Exploring principles of user-centered agile software development: A literature review.," *Information and software technology,* vol. 61, pp. 163-181, 2015.

[3] G. Papadopoulos, "Moving from traditional to agile software development methodologies also on large, distributed projects.," *Procedia-Social and Behavioral Sciences,* vol. 175, pp. 455-463, 2015.

[4] T. R. K. a. M. T. Suomalainen, "Continuous planning: an important aspect of agile and lean development," *International Journal of Agile Systems and Management* , vol. 8.2, pp. 132-162, 2015.

[5] R. G. Cooper, "Agile–Stage-Gate Hybrids: The Next Stage for Product Development Blending Agile and Stage-Gate methods can provide flexibility, speed, and improved communication in new-product development," *Research-Technology Management,* vol. 59, no. 1, p. 2, 2016.

[6] F. e. a. Anwer, "Agile Software Development Models TDD, FDD, DSDM, and Crystal Methods: A Survey.," *International journal of multidisciplinary sciences and engineering,* vol. 8, no. 2, pp. 1-10, 2017.

[7] M. a. R. R. Alqudah, "A review of scaling agile methods in large software development.," *International Journal on Advanced Science, Engineering and Information Technology,* vol. 6.6, pp. 828-837, 2016.

[8] J. M. Bass, "Artefacts and agile method tailoring in large-scale offshore software development programmes.," *Information and Software Technology* , vol. 75, pp. 1-16, 2016.

[9] T. e. a. Dingsøyr, "Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation," *Empirical Software Engineering,* vol. 23.1, pp. 490-520, 2018.

[10] G. S. e. a. Matharu, "Empirical study of agile software development methodologies: A comparative analysis," *ACM SIGSOFT Software Engineering Notes* , vol. 40, no. 1, pp. 1-6, 2015.

[11] F. S. e. a. Silva, "Using CMMI together with agile software development: A systematic review.," *Information and Software Technology* , vol. 58, pp. 20-43, 2015.

[12] T. a. C. L. Dingsøyr, "Emerging themes in agile software development: Introduction to the special section on continuous value delivery," *Information and Software Technology* , vol. 77, pp. 56-60, 2016.

[13] E. M. V. M. a. J. I. Kupiainen, "Using metrics in Agile and Lean Software Development–A systematic literature review of industrial studies," *Information and Software Technology* , vol. 62, pp. 143-165, 2015.

[14] John, "Systematic Lanugages," *Information Technology,* vol. 4, no. 3, pp. 2-9, 2019.

[15] V. T. e. a. Heikkilä, "A mapping study on requirements engineering in agile software development.," in *41st Euromicro conference on software engineering and advanced applications. IEEE*, 2015.

[16] T. e. a. Dingsøyr, "Coordination in multi-team programmes: An investigation of the group mode in large-scale agile software development.," *Procedia Computer Science* , vol. 121, pp. 123-128, 2017.

[17] R. V. a. M. D. Anand, "Popular agile methods in software development: Review and analysis.," *International Journal of Applied Engineering Research,* vol. 11.5, pp. 3433-3437, 2016.

[18] M. e. a. Al-Zewairi, "Agile software development methodologies: survey of surveys.," *Journal of Computer and Communications,* vol. 5.05, pp. 74-97, 2017.

[19] T. e. a. Dingsøyr, "Team performance in software development: research results versus agile principles.," *IEEE software,* vol. 33, no. 4, pp. 106-110, 2016.

[20] Y. e. a. Lindsjørn, "Teamwork quality and project success in software development: A survey of agile development teams.," *Journal of Systems and Software,* vol. 122, pp. 274-286, 2016.