International Journal of Latest Trends in Engineering and Technology
Vol.(15)Issue(3), pp.012-015
DOI: http://dx.doi.org/10.21172/1.153.03
e-ISSN:2278-621X

A COMPARATIVE STUDY ON TRADITIONAL VS AGILE SYSTEMS DEVELOPMENT METHODS

Said A. ALMandhari[1], Emdad S. M. H[2], Mohamed Aissa[3], Mohammad A. K[4]

Abstract- The selection of System Development Methods is so essential in software engineering. The main idea of it is to keep track of the software processes in a structured and methodical way that meet organization's requirements and save a great amount of time and money and thus using an appropriate approach eliminates errors or issues prematurely. However, the limitations or weaknesses in every system development methodology make it difficult to get a certain method suitable to all different projects. This research paper presents an overview and comparative study on system development methods and highlights the limitations that are possible to be evolved. On this occasion, we will make a comparison between different methodologies to show the features and limitations of each model and we will link between methods challenges and the causes of project failures to come up with a hypothesis that can be suitable and adaptable to different projects.
Keywords – Agile method, method engineering, system development methods, system development life cycle

1. INTRODUCTION

Nowadays, organizations in different sectors rely on information systems as the most powerful factors that drive the process to become faster and smoother in achieving their goals [1]. This requires having an efficient System Development Methods (SDM) capable of evolving in order to meet changing requirements. SDM is a set of tools that assist system developers to implement an information system [2]. As just one of many available methodologies, SDM takes its place and importance as a research methodology.

The subject matter of this research paper is to give an overview of the different SDM and make a comparison between them in order to show the features and limitations of each model. It also shows the agile methods that allow teams to respond to the changing project requirements and to search for important factors for avoiding project failure by using agile methods.

The rest of the paper is organized as follows. Background and literature review are explained in section II. The methodologies are presented in section III. Analysis and results are presented in section IV. Concluding remarks are given in section V.

2. BACKGROUND

The increasing scientific of the technology industry today is closely linked to the very diverse knowledge progress in the world that necessitates the creation of mechanisms and methods adapted to the changes and expansion in the development of information systems programs [3]. Systems Development Life Cycle (SDLC) models are abstract descriptions of the structured methodical development and modification process typically showing the main stages in producing and maintaining executable software [4]. In that time, the goal was to keep systems in control and production of what the system is supposed to achieve and thus it was relatively easy to divide a project into a sequence of steps where each subsequent step builds on the outputs of previous steps. The most popular depiction of this SDLC is the waterfall model which is a sequential process of software development [5].

As the information systems evolved and unpredictable factors started to be introduced, it became unrealistic to design systems along a strict sequence of steps. A survey on a comparative study of SDM will be presented in this paper to show some essential aspects of a new approach to evolving an acceptable SDM. Since the working practices in the information system fields cover modelling techniques, we will include adaptation of modelling techniques and tools in this study as well as some aspects of the agile methods. Based on various technical, organizational, project and team considerations, each of the available methodologies has strengths and weaknesses [6]. A comparison study between different types of SDLC models in software engineering will be put in place in this paper that is looking at software development through the development models. This research paper leads us to study the different aspects of SDM that needs modifications in some circumstances and with a view to have of a new approach and how are adapted in future practices. The following discussion gives an overview of some traditional methods of SDLC and substantially focuses on the adaptation of modelling techniques.

2.1 Traditional Methodology

It is diagrammatic representation looks like a cascade flow of Waterfall and also known as classical life cycle model, it is one of the oldest process model defined in 1970 [6]. All phases of the waterfall model are independent of each other and developers must complete each phase before the next phase could begin.

[1,2,3,4] Department of Information Systems, University of Nizwa, Nizwa, Ad-Dakhiliyah Governorate, Oman

Traditional methodologies provide benefits for organizations through the clarity of tasks to be performed in the sequenced steps and through the documentation that is a basic part of the methodology which helps to reduce reliance on the original developers and thus contributing to the sustainability of the project and ongoing support [7].

2.1.1 Weaknesses of the Traditional Methodology:

Like any methodologies, Traditional SDLC accompanied by some weaknesses and limitations that should be handled carefully and can be summarized in the following points [8]:

- Inflexible, slow and cumbersome due to significant structure and tight controls.
- It depends upon early identification of requirements, yet users may not be able to provide early accurate requirements.
- Difficult to manage changes that occur later in the project.
- Promotes the gap between users and developers with clear division of responsibility.

In general, systems with little user interaction and the wise use of tools and strict involvement to the concepts and decisions can increase the probability of successful implementation of waterfall projects by identifying a manageable scope for the project and ensuring continued the project, even if key players leave the organization.

2.2 Prototyping Methodology

The prototyping model is an iterative framework evolution from the waterfall model, follows an approach handling selected portions of a larger (i.e. Incremental, Spiral, or Rapid Application Development) [9]. Some researchers found that SDLC models that adopted prototyping tend to be more dynamic and more responsive to client needs, as well as less risky and more efficient [8]. Prototyping model is mostly used when the desired system needs to have a lot of interaction with the end users. It allows feedback which may generate specifications or new features for a production application.

2.2.1 Weaknesses of the Prototyping Methodology:

In the prototyping methodology, the disadvantage is given in the following points [10]:

- Requirements may frequently change significantly.
- Designers may prototype too quickly, without sufficient up-front user needs analysis resulting in an inflexible design with narrow focus that limits future system potential.
- Unskilled designers may lead to poorly designed systems.
- Iterations add to project budgets and schedules, thus the added costs must be weighed against the potential benefits.

Generally, prototyping model is especially good for project wherein the requirements are not well understood and for large and critical projects. One of the most productive uses of rapid prototyping to date has been as a tool for iterative user requirements engineering and human-computer interface design.

2.3 The Agile Methodologies

At the end of the 20th century, besides the cited methodologies, Agile Development Methodologies in developing information systems appeared [11]. Searching for a solution to overcome the challenges of traditional methods, some software developers are adopting agile methods [12]. They handle carefully with the main project stakeholders groups and they are; the decision maker, the software users and the software developers. These methodologies mainly focus on real-time face-to-face communication in order to minimize the risk and to deal professionally with rapidly changing on customer demands [13]. The most common used agile methods are Scrum, Kanban, Extreme programming (XP) and new Agile methodologies are still being created like the Lean method that has appeared in 2011 [14]. The basic and primary values of the agile approach are:

- Individuality and interaction versus standard processes and tools: The agile methods claim to place more emphasis on people, interaction, working software and customer collaboration rather than on processes, tools and plans [15].
- Customer collaboration over contract negotiation: The agile methodologies base on frequent contacts with their users, which means that client collaboration, is core to make the project successful [16].
- Change adapting versus keeping up with the plans: The agile approach suggests that we should be flexible and adaptable to changes because the directions of project development cannot be predicted for the far future [17]. The business environments change constantly, as well as the users themselves who change their requirements in relation to the system that is functioning.

2.3.1 The choice factors of traditional or agile Methodologies:

Further, the choice of traditional or agile methods for a given project is largely contingent on five factors [17]:

- The size of the systems development project.
- The consequences of failure (i.e., criticality).
- The degree of environmental change.
- The competence of personnel and team.
- Compatibility with the organization culture.

3. METHODOLOGY

In order to gain the necessary information for this research, a case study method was used besides a questionnaire on Systems Development Methods of different parameters and concepts that will be one of the appropriate ways to obtain a satisfactory result. The case study is a qualitative research method used to examine real-life situations in order to identify factors that take into account the organization's behavior patterns. It is easy to administer, and it can be of particular use if clarification of certain questions is required or if there are some problems with the respondents. Also, it can be a supplement to the results of the literature reviewed and it gives a realistic picture of the reasons associated with the failure of the programming projects with the ISDMs used by the System Development Team in any organization.

The case study of the Warehouse Management System (WMS) was conducted in one of the higher education institutions in Oman in March 2019. It was supported by a specialized interview with all the hierarchies of the system development team, with the participants being the Head of Development Department playing the role as the decision maker, the System Development Team Leader, and Senior Developers. The WMS is used to administer the inventory system of that institution, enabling the storekeeper to easily maintain all consumables and fixed assets. Also, anyone who requests items from the store would use the system to submit their request. The development team started work on the WMS project on September 2012 and continued until July 2016. The team included a project team leader, four system analysts and developers, two representative staff members from each department, and three data entry staff.

The case study explained the existence of a failed project, despite the fact that the institution used professional systems developers, and despite the fact that they followed the appropriate methods for each project. One of the biggest systems, WMS, failed 100% after almost four years of effort. The main reason for the failure was due to ignoring the users' requirements, with the basic principles of the traditional method used to develop it, namely the Waterfall Model, where new requirements were made after the completion of the implementation phase and programmers had to return to the requirement phase which in turn destroyed the essence of the system. It was clear that the development team adapted their future work by trying to create their own way of developing systems in order to create successful projects.

Further, an online survey using Google Forms Application was conducted in March 2019, using 16 questions. The purpose of the survey was to collect the opinions from personal experience with software development practices about Systems Development Methods. The strength of the survey was inspired by expert participants, and 54 respondents from Oman participated in the survey, providing a solid foundation of data to analyze. Most have a focus on technology, with 25.9% working in program leadership, 27.8% in software development and 29.6% in Information Technology.

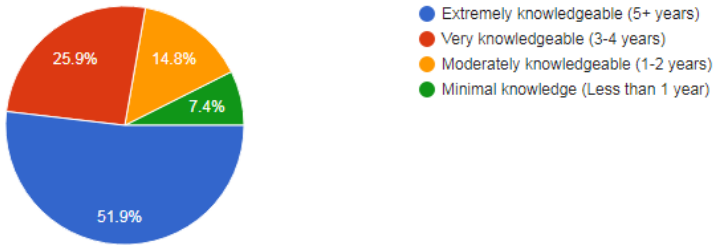

Figure 1. Level of Experience of participants

4. ANALYSIS AND RESULT

By studying and analysis related literatures it is clear that every system development methodology have some weaknesses and limitations resulting that leads to the need for method adaptation. Therefore we tried to highlight the possibility of adopting agile methods to be suitable, to some extent, with a certain information system in comparison with some traditional methods. But in the same time, there are a good opportunities to choose the traditional or prototyping methodologies if the organization and other factors fit with the advantages of those methodologies.

The table below summarizes the major differences between traditional and agile methodology.

Table -1Summary of Traditional and Agile Methodologies [7], [8], [12]

	Traditional approach	Agile approach
Organizational structure	Linear	Iterative
User requirements	Clearly defined before implementation	Interactive input
User involvement	Low: Get involved early in the project	High: Get involved throughout project lifecycle
Documentation	Basic part of all process	Minimal: Only as required
Bugs in the System	Not discovered until system testing	It can be discovered in early stages
respond to changes	Difficult to implement	Manageable

It is noticeable through the case study there was not a suitable methodology to be followed by the development team in order to avoid the projects failure. Therefore they sought to mix the available methods and, when necessary, to waive them in order to adapt to the users requirements in the organization.

From the survey, we see that the Traditional Waterfall Model is becoming a popular framework. In fact, more than 85% of the respondents report that the Traditional Waterfall Model is used more frequently than others. The survey uncovered some interesting key findings. 57.4% of respondents believe that there are limitations in Traditional Waterfall practices that cause projects to fail. In contrast, less than 26% believe that there are limitations in agile practices that cause projects to fail. In total we found two (2) major reasons for project failure, including decisional dilemmas. The identified causes are:

One of major causes of project failure is finding the correct methodology.

Even though the developers finalize the right method there is still a lack of understanding how to implement to method in the right direction.

However, to solve the dilemma and to solve those issues we conclude with number of recommendations which are also a clear reflection of our survey result. The recommendations are:

The authority must be flexible to adapt if needed; including both functional and non-functional requirements.

The mixed method can be one potential solution which also depends on the nature of the project.

To identify the nature of the project we must have an expert project leader with experience in a specific field.

5. CONCLUSION

This paper presents the study of different research papers related to a brief comparison between various process models in Information System Development Methods. It is found that every system development methodology has some weaknesses and limitations resulting in the need for method adaptation. Recently, an agile method could be the method of choice for information system development projects in most organizations. The main motivation for this method was to ensure to achieve substantial and adaptability product and process improvements in all projects. In this research we tried to highlight the possibility of adopting agile methods to be suitable, to some extent, with a certain information system in comparison with some traditional methods.

6. REFERENCES

[1] K. Ramzy, E. Helow, and A. M. Salem, “Exploiting the potential role of artificial intelligence in identifying the critical success factors for information centers in Alexandria Petroleum Company,” vol. 44, no. 1, pp. 12–22, 2020.

[2] J. Barnard, M. Huisman, and G. R. Drevin, “The development of a systems development methodology for location-based games,” Comput. Entertain., vol. 16, no. 3, 2018, doi: 10.1145/3236492.

[3] T. A. Jenkin, Y. E. Chan, and R. Sabherwal, “Mutual understanding in information systems development: Changes within and across projects,” MIS Q. Manag. Inf. Syst., vol. 43, no. 2, pp. 649–671, 2019, doi: 10.25300/MISQ/2019/13980.

[4] A. Saxena and &priya Upadhyay, “Waterfall vs. Prototype: Comparative Study of SDLC,” Imp. J. Interdiscip. Res., vol. 2, no. 6, pp. 2454–1362, 2016.

[5] L. K. Shinde, Y. S. Tangde, and R. P. Kulkarni, “Advances in Computational Research TRADITIONAL Vs . MODERN SOFTWARE ENGINEERING - AN OVERVIEW OF SIMILARITIES AND DIFFERENCES,” vol. 7, no. 1, pp. 187–190, 2015.

[6] P. Sharma and D. Singh, “Comparative Study of Various SDLC Models on Different Parameters,” Int. J. Eng. Res., vol. 4, no. 4, pp. 188–191, 2015, doi: 10.17950/ijer/v4s4/405.

[7] C. Tayntor, “Introduction to the Traditional SDLC,” Six Sigma Softw. Dev. Second Ed., pp. 209–217, 2007, doi: 10.1201/9781420044287.ch16.

[8] M. O. C. R. S. View, “A Comparison Between Five Models Of Software Engineering,” Int. J. Comput. Sci. Issues, vol. 7, no. 5, pp. 94–101, 2010.

[9] A. Kazim, “A Study of Software Development Life Cycle Models,” Int. J. Adv. Res. Comput. Sci., vol. 8, no. 1, pp. 15–23, 2017.

[10] I. Investigation, “Selecting A Development Approach,” Off. Inf. Serv. USA, pp. 1–10, 2008.

[11] N. Abbas, A. M. Gravell, and G. B. Wills, “Historical roots of agile methods: Where did ‘Agile thinking’ come from?,” Lect. Notes Bus. Inf. Process., vol. 9 LNBIP, pp. 94–103, 2008, doi: 10.1007/978-3-540-68255-4_10.

[12] C. M. Budoya, M. Kissaka, and J. Mtebe, “Instructional Design Enabled Agile Method Using ADDIE Model and Feature Driven Development Process.,” Int. J. Educ. Dev. Using Inf. Commun. Technol., vol. 15, no. 1, pp. 35–54, 2019.

[13] V. Dattatreya, “Agile Programming and Design Patterns in Web Development - A Case Study,” Int. J. Softw. Eng. Appl., vol. 3, no. 1, pp. 37–45, 2012, doi: 10.5121/ijsea.2012.3104.

[14] A. Henriksen, “Agile project management - A case study on agile practices,” ProQuest Diss. Theses, no. February, 2016, doi: 10.13140/RG.2.2.14048.33283.

[15] M. Gechman and M. Gechman, “Software Development Methodologies,” Proj. Manag. Large Software-Intensive Syst., pp. 49–66, 2019, doi: 10.1201/9780429027932-4.

[16] S. R. Mohammed and A. J. Jasim, “Examining the Values and Principles of Agile Construction Management in Iraqi Construction Projects,” J. Eng., vol. 24, no. 7, p. 114, 2018, doi: 10.31026/j.eng.2018.07.08.

[17] M. Raffai and K. Kovács, “I3P : the IT Professionalism Program,” 2008.