A Comparative Study on Hybrid IT Project Managment

Using Traditional Project Management and Agile Approach

Mojdeh Rahmanian
Department of Computer Engineering
Jahrom University
Jahrom, Iran
Email: mrahmanian {at} jahrom.ac.ir

Abstract—**Traditional Project Management Methodologies (TPMM) aim to prevent change by extensively planning and documenting as much as possible before the system is developed while change is inevitable and that it is not to be avoided. Additionally, Traditional Project Management rely more on processes, sequential software development, like waterfall. Consequently, there is a demand for system development methodologies and project management methodologies with the ability to adapt to a changing project and business environment. The aim of this study is to investigate hybrid IT project management that flexibly combining the traditional and the agile method; it emphasizes on two method of blending Scrum, an agile method, into traditional plan-driven project development and management such as waterfall or into structured and widely accepted project management methodologies such as Project Management Body of Knowledge (PMBOK). A comparison is then done of the two selected different hybrid methods.**

Keywords- Project Management, waterfall, Agile Method, PMBOK, Scrum Model;

I. Introduction

Information technology (IT) projects are very difficult projects to manage, as they have the tendency to change owing to elements of uncertainty such as project time and budget instability, constantly changing business and user requirements, and the team's ability to respond to new expectations. Because of the evolving business environment, the requirements set by business and users change frequently and unexpectedly. Consequently, there is a demand for system development methodologies and project management methodologies. [17] The answer to these demands is agile project management (APM). Agile project management is an outgrowth of the agile software development movement. APM is a highly iterative and incremental process, where developers and project stakeholders actively work together to understand the domain, identify what needs to be built, and priorities functionality.

Many researchers believe that using an agile approach to managing projects is totally different from a more traditional approach, such as PMBOK or waterfall. The plan-driven approach is more suited for large-scale project with heavy constraints, large risks and clear up-front requirements and agile approach can only be applied to small-scale projects with less rigid constraints, smaller risks, unclear requirements. In this paper we study and compare two hybrid method. First, we discuss one hybrid method using an agile approach and the PMBOK® Guide. Next, we investigate another hybrid model that apply an agile method, Scrum into the traditional software development process and IT project management such as waterfall

The reminder of this paper is organized as follows. In Section II, agile methods in IT project development and management and PMBOK are reviewed, then the hybrid model using agile and the PMBOK is discussed. Section III reviews the waterfall model. Then, another hybrid IT project development and management model is proposed, which is to blend the Scrum method into the traditional plan-driven software development process such as waterfall. Section V compare hybrid models and demonstrates the benefits of the new mixed models. Finally Section VII is the concluding remarks.

II. Project Managment Using Agile and the PMBOK® Guide

This method is a paradigm shift from the traditional plan-then-execute-project paradigm that embraces the fundamentals of the normal five-stage (initiate, plan, execute, control, close-out) project life-cycle phases in the PMBOK to a new five-phase (envision, speculate, explore, adapt, close) project life cycle in an agile approach. This section will introduces the agile method and PMBOK briefly, next investigate the hybrid model.

A. *Overview of agile project management*

Several agile leaders were called together to create the Agile Software Development Manifesto in 2001[1]. The primary goal of any Agile Software Development is to make an organization agile, in other words, give the organization the ability to adapt to change. It is characterized by the ability to handle changing business requirements, incremental and iterative development, and continuous code integration [2].

Scrum management that has been extensively practiced in various sized organizations is one of the most often used model of agile. Scrum is based on the following: three roles:

- **Product owner**, is defined as the person responsible for the product backlog and he/she represents the expectations, constraints, and interests of the stakeholders [17].
- **Scrum master**, is responsible for the overall project's success, including the delivery of products of good quality, facilitation of communication, removal of impediments, and the process as a whole [13].
- **Team**, a cross-functional team of five to nine people who organize themselves and the work to produce the desired results for each sprint.

Three documents and processes:

- **Product backlog**, contains the body of work required during the entire project. This includes requirements gained from software developers, clients and experts. All requirements are prioritized according to descending order of importance. Owing to the dynamic environment, the product backlog must be constantly updated and prioritized as new requirements are identified.
- **Sprint backlog**, is the starting point for every sprint, which contains all the tasks and requirements that ought to be completed during the current sprint.[12]
- **Burndown chart**, shows the cumulative work remaining in a sprint on a day by day basis. [14]

And three meetings:

- **Sprint or pre-sprint planning**, entails identifying the tasks necessary to reach the defined sprint goal. These identified requirements and tasks are moved from product backlog to sprint backlog to be completed during the next sprint after they have been prioritized [12].
- **Daily scrum**, every morning, a short meeting of approximately fifteen minutes is held to keep track of the development process. During each meeting, the Scrum team specifies what has been done since the last meeting, and discusses what should be done before the next meeting takes place. During these meetings, problems are identified and solutions are suggested to keep the team focused on the goal. Furthermore, the meeting enhances communication by keeping the stakeholders and team members involved and up to date [3].
- **Post-sprint or sprint review meeting**, after the first sprint, a post-sprint meeting is held during which a decision is made regarding whether the team should continue with the project. If this is agreed upon, a pre-sprint meeting is held to identify tasks to be completed during the next sprint. [3]

B. Overview of Project Managment Body of Knowledge (PMBOK)

The first edition of A Guide to the Project Management Body of Knowledge was published in 1996 followed by the second edition in 2000, third edition in 2004 and fourth edition in 2008. The fifth and current edition became available late December 2012. The current PMBOK Guide (fifth edition) recognizes 47 processes that fall into five basic process groups and ten knowledge areas. The five process groups are:

- Initiating
- Planning
- Executing
- Monitoring and Controlling
- Closing. [4,17]

Each project management process group is characterized by certain tasks that must be completed will be discussed in this section briefly.

Initiating In order to initiate a project or a project phase, a project manager must be assigned to manage the project, the requirements of the business must be defined, and a sponsor for the project must be obtained [5]. It is important to note that the initiation process is a part of all phases of the project, as before something can be planned or executed it must be initiated [6].

Planning The planning process entails the development and maintenance of a workable scheme to ensure that the project satisfies and addresses the business's requirements [17].

Executing This process entails the organization and coordination of resources to execute the various plans, actions and tasks to produce the deliverables, goals, products, services or results during the different phases of the project [7].

Monitoring and Controlling This process entails the continuous monitoring, measuring and controlling of project-related activities to ensure that the project team meets the objectives of the project.

Closing This process entails the formalized acceptance of the project's phases or the project as a whole and ending it efficiently and effectively [7].

C. Agile Project Management using PMBOK

This hybrid model called Agile Project Management (APM) consists of five phases, namely: envision, speculate, explore, adapt and close.[17,14] The APM delivery framework moves away from the traditional phase names such as initiate, plan, execute, monitor, close. This has a great significance, as it indicates that APM embraces change and changing as often as required.

Envision During the envision phase, the project scope, project community and stakeholders, product vision, and the manner in which the team works together are determined. Firstly, the team must envision what must be delivered. In order to do this, the project objectives, constraints, boundaries and vision must be defined and understood. Secondly, the team has to identify which stakeholders and members of the business community will be involved. Lastly, the team must decide on the manner in which they will work together in order to deliver the expected and defined vision of the project [17].

Speculate APM is more than just planning and doing, it is about creating a vision and exploring it because only some information is available and this must be examined to determine our course of action in the next iteration [8]

Explore During the explore phase, features are planned, developed, tested and delivered in short iterations while the objective is to constantly reduce the uncertainty and risk of the project.

Adopt adapt implies modification or change rather than success or failure. In order to adapt, there must be an understanding of the risks, changing requirements, project processes and the market.

Close By compiling a closeout report, the project team summarizes what has been completed successfully and finally agrees that the expectations of the client have been met.

III. Project Management Using Agile and Scrum

Most of the companies are struggling with how to use the agile practices into their IT software project development and management. Most of organizations already have developed the traditional plan-driven method such as waterfall for the IT project management. The Agile method brings a big change in the way how the teams are managed during the projects. This section briefly explains Waterfall model then investigates the hybrid model.

A. Overview of waterfall model

The Waterfall model is a sequential software development process in which progress is regarded as flowing increasingly downwards (similar to a waterfall) through a list of phases that must be executed in order to successfully build a computer software. Originally, the Waterfall model was proposed by Winston W. Royce in 1970 to describe a possible software engineering practice [15]. The Waterfall model defines several consecutive phases that must be completed one after the other and moving to the next phase only when its preceding phase is completely done. For this reason, the Waterfall model is recursive in that each phase can be endlessly repeated until it is perfected. Figure. 1 depicts the different phases of the Waterfall model.

Essentially, the Waterfall model comprises five phases: analysis, design, implementation, verification, and maintenance [9].

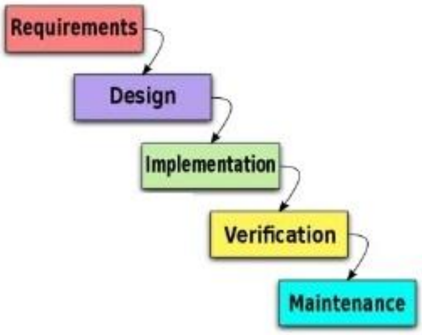

Figure 1. Waterfall Model

Analysis Phase: Often known as *Software Requirements Specification* (SRS) is a complete and comprehensive description of the behavior of the software to be developed. It implicates system and business analysts to define both functional and non-functional requirements. Usually, functional requirements are defined by means of use cases which describe the users' interactions with the software. They include such requirements as purpose, scope, perspective, functions, software attributes, user characteristics, functionalities specifications, interface requirements, and database requirements.

In contrast, the non-functional requirements refer to the various criteria, constraints, limitations, and requirements imposed on the design and operation of the software rather than on particular behaviors. It includes such properties as reliability, scalability, testability, availability, maintainability, performance, and quality standards.

Design Phase: It is the process of planning and problem solving for a software solution. It implicates software developers and designers to define the plan for a solution which includes algorithm design, software architecture design, database conceptual schema and logical diagram design, concept design, graphical user interface design, and data structure definition.

Implementation Phase: It refers to the realization of business requirements and design specifications into a concrete executable program, database, website, or software component through programming and deployment. This phase is where the real code is written and compiled into an operational application, and where the database and text files are created. In other words, it is the process of converting the whole requirements and blueprints into a production environment.

Testing Phase: It is also known as *verification* and *validation* which is a process for checking that a software solution meets the original requirements and specifications and that it accomplishes its intended purpose. In fact, verification is the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase; while, validation is the process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements. Moreover, the testing phase is the outlet to perform debugging in which bugs and system glitches are found, corrected, and refined accordingly.

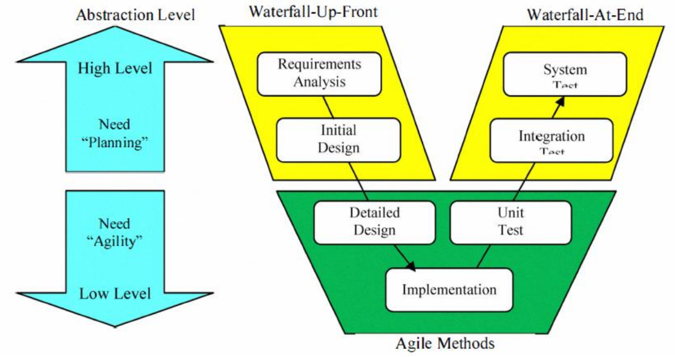

Figure 2 A Hybrid Model for Software Development and Project Management[10]

Maintenance Phase: It is the process of modifying a software solution after delivery and deployment to refine output, correct errors, and improve performance and quality. Additional maintenance activities can be performed in this phase including adapting software to its environment, accommodating new user requirements, and increasing software reliability [9].

B. Hybrid Model : Waterfall and agile

This hybrid model that proposed in [10, 16] is a new approach to the IT project development and management by blending Scrum, an agile method, into the traditional plan-driven project development and management. The proposed model uses the terms proposed by Cohn [11] "Waterfall-Up-Front" and "Waterfall-At-End" as part of the traditional waterfall method.

We can see the benefit of this hybrid development from three aspect. First, the project team and the client can apply a "waterfall-up-front" to specify the requirements, and link them together as one contract. The risk of confusion in terms of project objectives and deliverables, will reduce. Second, agile methods in the design, implementation and unit testing phases can be applied by the team. This speeds up the process and reduce the risk of Rework, delays and rescheduling that we often see in the traditional development of project. Third, the team and the client can request a "waterfall-at-end" for high-level testing and acceptance. The hybrid V-model is the suggestion that, while many business organizations are cherry picking agile methods to develop their own styles, little evidence of how and when they should blend agile methods to their projects effectively. Many project teams confusing mix of methodologies and combining them with agile methods in their organizational realities. [10]

CONCLUSION

Like these two reviewed hybrid models, other project management methodologies must be implemented and improved to make it more applicable in practice. For many projects, the pure traditional project management is not effective, and hybrid models are the most appropriate solution. The traditional project management method is better fit for the projects with a clear goal and solution, and there are number of these during the year. On the other hand, projects that have no clear goal and solution are managed more with the agile method, but the pure agile method is not good enough for many such projects, and the agile method is adjusted flexibly, resulting in the flexible hybrid method. It's believed that this combination of agile techniques with traditional method forming the hybrid ways of managing the IT projects is just emerging, and we will see more of it in the future,

REFERENCES

[1] K. Beck, et aI., Manifesto for Agile Software Development, 2001. www.agilemanifesto.org/

[2] BOEHM, B. & TURNER, R. 2005. Management challenges to implement agile processes in traditional development organizations. *IEEE software*, 22(5):30–39.

[3] STOBER, T. & HANSMANN, U. 2010. Agile software development: best practices for large software development projects. Berlin Heidelberg: Springer. 179p

[4] http://www.projectsmart.co.uk/

[5] SNYDER STACKPOLE, C. 2010. A user's manual to the PMBOK guide. Hoboken, NJ: Wiley. 240 p.

[6] SHERRER, A.J. 2009. Project management road trip for the project manager professional. J. Alex Sherrer. 571 p.

[7] CONCHUIR, D. 2010. Overview of the PMBOK guide: shortcuts for PMP certification. Berlin: Springer. 213 p. 8

[8] HIGHSMITH, J. 2010. Agile project management: creating innovative products. 2nd ed. Boston, MA: Addison-Wesley. 432 p.

[9] Y Bassil, A Simulation Model for the Waterfall Software Development Life Cycle, International Journal of Engineering & Technology (iJET), ISSN: 2049-3444, Vol. 2, No. 5, 2012.

[10] Hayata, T., Jianchao Han, A hybrid model for IT project with Scrum, this paper appears in: Service Operations, Logistics, and Informatics (SOLI), 2011 IEEE International Conference on Page(s): 285 - 290; Issue Date: 10-12 Jul 2011

[11] M. Cohn, Succeeding with Agile: Software Development Using Serum. Addison-Wesley Professional. Boston, Massachusetts, USA, 2009.

[12] COHEN, D., LINDVALL, M. & COSTA, A DACS state of the art report: agile software development, 2003.

[13] STOBER, T. & HANSMANN, U. Agile software development: best practices for large software development projects. Berlin Heidelberg: Springer, 2010.

[14] K. Schwalbe, A. College, Managing a Project Using an Agile Approach and the PMBOK® Guide, ISECON conference, New Orleans, LA, 2012,

[15] Winston W. Royce ,Managing the Development of Large Software Systems, Technical Papers of Western Electronic Show and Convention (WesCon), Los Angeles, USA, August, 1970.

[16] E. Bergman, A. hamilton, Agile-Waterfall Hybrid: Living together in perfect harmony!, Agile conference, tennessee , August 2013.

[17] J.Grey, The development of a hybrid agile project management methodology, Potchefstroom Campus of the North-West University, June 2011.