International Journal of Engineering & Technology, 7 (4.28) (2018) 510 -514

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Preliminary Study of Methodology on the Impact of Different Teamwork Skill Levels on the Efficiency on the Software Development Workflow Based on Industry in Malaysia

Zairina Ibrahim[1*], Md Gapar Md Johar[2], Normy Rafida Abdul Rahman[3]

[1,2]Information Technology and Innovation Centre, Management and Science University, Selangor, Malaysia.
[3]Faculty Business Management and Professional Studies, Management and Science University, Selangor, Malaysia
**Corresponding author E-mail: zairina@msu.edu.my*

Abstract

The software development life cycle (SDLC) is the important element in software development. In fact, there are needs to upgrade the sequence of methodology in software development. The SDLC methodology is the highest level of abstraction and is defined as a collection of problem-solving methods governed by a set of principles and common philosophies in solving targeted problems for extended project timeline teamwork as well as a decrease of a high team collaboration and understanding in the software development. Thus, the SDLC is very crucial in order for them to ensure the quality of skills is placed accordingly in the workflow. This research contributes to the development of a new approach in system development workflow with the aim to properly manage system development projects. It started by providing some background data related to the previous mode of operation in the teamwork samples as shared by the stakeholders of the transformation projects and the new proposed Analysis System Development Framework (ASDF) method team members. Then, the key findings related to themes of 1) input for User Requirement Specification (URS) and 2) System Requirement Specification (SRS), 3) proses for module, 4) process for database, 5) process for User Acceptance Testing (UAT) 6) output for Final Acceptance Testing (FAT) and empowerment for the whole level based on ASDF method. This research contributes mainly to the findings, more than previous research has found, that significantly support the perception that a high quality of skills in a teamwork results in better performance of software development.

Keywords: *Software Development Life Cycle; Quality Skill of Team Work; System Analysis.*

1. Introduction

In the 21st century of Malaysian context, by supporting ubiquitous access to any kind of web-based software development, there are needs to upgrade the sequence of software development life cycle (SDLC) methodology. The software development life cycle methodology is defined as a collection of problem-solving methods governed by a set of principles and common philosophies in solving targeted problem, and it is the highest level of abstractions. In fact, the system analyst, project manager, and top management are the parties involved throughout the software development. Thus, the software development life cycle is very crucial in order for them to ensure the quality of skills is positioned accordingly in the workflow.

According to Agarwal & Rathod (1), the world of workflow is severe and it acknowledges the problems of workflow management systems are due to their demanding and formal nature. Most of the time the natural human interaction incorporates flexibility, opportunistic behavior, social awareness and compromise. However, the workflow implementations tend to be coercive, isolationistic and inflexible.

In the 21st century teaching and learning, the main concern is how to measure the skills. Among the civil rights advocates, there is a concern that the measurement would affect the progress towards ensuring common standards of learning for all students due to the high cost and time required as well as the difficulty to score in the tests which result in a slow adoption of such tests. Collectively, these concerns derailed efforts in the late 1990s to move toward the use of performance-based assessments such as portfolios, exhibitions, and projects (2).

The workflow as the ability to support workflow processes in which a static representation of the process and its resources has to be made. This static representation is referred as a workflow definition, such a workflow definition can be divided into two important parts: the process model and the resource classification. The process model consists of various steps and stages while the resource classification indicates the kind of work an employee is allowed to perform (3).

The new proposed software development workflow in this research is being compared to the agile method. Therefore, the agile method was defined as a holistic product development strategy and flexible in which the development team works as a unit to reach a common goal. Generally, in software development, the tasks are designed an interdependently in which team members handling those tasks should work interdependently on each other (4). Subsequently, as the task interdependency increases, the need for coordination among the team members would increase.

Furthermore, depending upon the complexity of the goal, the complexity of its underlying tasks will increase exponentially and call out for a tight coordination among the team members to work

collectively and interact effectively, which is accomplished through teamwork.
This research introduces a proposed new development workflow known as analysis system development framework (ASDF). Hence, it is up to each development process to decide the workflow that is deemed appropriate for the system analyst. The ASDF has six steps to complete the system that are user requirement system (URS), specification requirement system (SRS), module, database, user acceptance test (UAT) and final acceptance test (FAT). Thus, this workflow is proven to be the most efficient in software development workflow as it reduces the workflow time and cost especially when it is being compared to the agile methodology.

2. Literature Review

Nowadays, the existence of a system that becomes a foundation for a rapid development of the information technology sector is able to create a new environment around the world. Historically, the development of information technology that focuses on the workflow development such an agile method is no longer to practiced due to the fast advancement of system development workflow should be enhanced in accordance with the current advancement. In fact, the software development should be focusing on the quality of skills required in improving the performance in order to reduce the risk and improve the skills in software development.
Teamwork is an accurate organizational measure that shows many different features in all type of organizations including non-profit organizations (5), thus, teamwork can be considered as a significant tool of a new type of working organization. In fact, one research study concluded that teamwork is necessary for all types of the organization including non-profit organizations (6) in which each team member has the opportunity to enhance knowledge, skills, and abilities while working in a team (7).
However, Bacon & Bylton (3) highlighted self-management team and interpersonal team skills as the two important factors that enhance the communication as well as the interpersonal relationship between team members and also boost the employee performances. One research study concluded that a good manager is someone who has the ability to assign responsibilities to his or her subordinates in a form of a group or team in order to optimize the output from them (8).
Another study concluded that it is possible to design a team building software for employees within every organization in order to establish best practices and obtain maximize output. In addition, the main purpose of having such a software in place is to improve employee learning (9). According to Ingram (8), teamwork is a strategy that has a potential to improve the performance of individuals and organizations, but it needs to be nurtured over time. Organizations need to constantly look at ways to improve performance, especially in a rapid changing working environment. The management needs to inculcate teamwork activities within the organizations, be flexible to promote it and be willing to allow the teams to be part of decision making. In fact, Conti & Kleiner (10) reported that teams offer greater participation, challenges, and feelings of accomplishment.
For the previous of findings research, according to Langfred (11) and Hoegl & Gemuenden (12), Salas et al. (13), the issues of what processes and components comprise teamwork and how teamwork contributes to team effectiveness and team performance has been much studied, but there is no consensus concerning its conceptual structure. Therefore, using recent research and previous reviews, Dickinson and Mclntyre (14). Identified and defined seven core components of teamwork. In fact, using these components and their relationships as a basis, they proposed the teamwork model that is used in this work. The model consists of a learning loop of the following basic teamwork components such as communication, team orientation, team leadership, monitoring, feedback, backup and coordination.
According to Kraut (15), in the Dickinson and Mclntyre model, the components of monitoring, feedback and backup are in intermediate processes for ensuring effective teamwork. Finally, the output component is coordination because it defines the performance of the team. The communication is a transversal component of particular importance, because it links the other components. To build software effectively, there is a need for tight coordination among the various efforts involved so that the work is completed and fits together.
According to Diefenbach (16), another possible limitation is that we based much of our data collection and analysis on semi-structured interviews. The use of multiple data sources made it possible to find evidence for episodes and phenomena from more than one data source. We also observed, talked to, and interviewed the team members over a period of 9 months, which made it possible to study the phenomena from different viewpoints as they emerged and changed. Therefore, this limitation can be solving with the new proposed ASDF method to validate the skill of quality in teamwork.

3. Workflow System Model in Teamwork

According to Dingsoyr et al. (17), Agile software development methods have led to a number of changes in the way software is developed One of the principles of the agile manifesto states that "the best architectures, requirements, and designs emerge from self-organizing teams". Despite reports on the major improvement of agile development methods over traditional development methods (18), team performance is still a challenge. Team performance has been studied in a number of research fields, such as management science and psychology, which led to the development of teamwork effectiveness models. Research by Stray et al. (19), summarized the following challenges of teamwork: wrong tasks being solved by team members as they are working on low priority items, critical decisions are made without team commitment due to a lack of communication, and very minimal time is spent by many agile teams to reflect their work process. Thus, not releasing the learning potential.
According to Rodd (20), there are five“5” stages of developing a team. The first stage is a stage in which a new group of people comes together to start working as a team. The stage two deals with conflict resolution in a team which aims to bring the team members towards greater acceptance, increased trust and commitment to the task. In stage three, team members are encouraged to participate and contribute to the tasks assigned. This is followed by stage four in which the team members contribute to the tasks assigned. This is followed by stage four in which the team members contribute equally and collaborate with each other in achieving common goals. Finally, stage five is the stage in which team members will reflect and celebrate their achievements and later moving on to the next level.
According to Hoegl & Gemuenden (12), Liang et al. (21) and Henderson & Lee (22), software quality was shown to be dependent on good teamwork. In fact, the statistics reported by The Standish Group (23) showed 32% of information system development projects have a low success rate, 44% of the projects exceeded the planned budget and time while the remaining 24% of the projects experienced a complete failure. Software development is primarily a team effort (24), thus, it is important to understand the factors or characteristics that have significant influences on software development team performance. This piece of information would be very beneficial in promoting the project success.

4. System Development Teamwork Skills

Teamwork has to be treated as a situation in which everyone should be equal in terms of contributing and be responsible for the quality and success of a project. If in any case a failure can be pointed to any of the team members, the success of the project can

no longer be guaranteed. Focusing on teamwork also indicates that every team member is equally important and this helps to motivate them throughout the project. This leads to a high identification of team members with the product as highly motivated team members will contribute to the high-quality product as they are more focused and diligent when performing their work. The same is applicable to a software development process in which it should include a well-defined team structure, including an efficient task assignment and clear communication guidelines (25).

According to Highsmith (26), developers are found of agile development methods because they reflect how software really gets developed. As most software nowadays is developed by teams, the research has taken the approach of showing the strong connections between motivation and effective teamwork, and later showing how agile development methods are related to the latter.

However, a high-quality software depends just as much on high-quality collaboration within the team. As the success rate of software development projects is low (27), it is crucial to understand and identify the characteristics of interactions within software development teams that have significant impacts on the team performance. Thus, Hoegl & Gemuenden (12) reported proven evidence for the relation between teamwork quality and software quality through the use of the six factor teamwork quality (TWQ) model such as teamwork factors, communication, coordination of expertise, cohesion, trust, mutual support, value diversity, and project performance. This study further extends the findings of Hoegl & Gemuenden (12) with the aim of identifying additional factors that may have significant impacts on software team performance.

5. Literature Studies on Software Development Life Cycle Model Concerning Quality of Skills in a Teamwork

Management, especially in this era of rapidly growing competition, recognize the importance of teamwork than it was used to before. Collaboration among team members helps to expand the outputs of each team. Employees who are working in teams become the benchmark for the organization (28). It is the mean of improving manpower utilization and potentially raising the performance of the individual. When an employee works confidently in a team and with a support from upper-level management it will increase the productivity of the organization. Employees, in the new business world, are being assigned to more team projects which provide them with opportunities to strengthen their knowledge and develop their skills (29). A recent study shows that employees working within a team rather than individually are producing more output (30). This research uses a new model to find out the impact of teamwork, team spirit, team trust and recognition and rewards for employee performance. The objectives of this research are to find out the quality of skills in a teamwork on employee performance and also to find out the relationship between employee teamwork and employee performance.

According to Yu Beng et al. (31), software development life cycle is a process of building or maintaining software systems. Typically, it includes various phases from preliminary development analysis to post-development software testing and evaluation. In fact, it consists of models and methodologies that development teams use to develop the software systems. The said methodologies form the framework for planning and controlling the entire development process. A software application or an information system is designed to perform a particular set of tasks, which involve complex computation and processing that will produce well-defined results. It is, therefore, a tough and tedious job to administer the entire development process. This is to ensure that the end-product comprises of a high degree of integrity and robustness, as well as user acceptance. Thus, in order to accomplish the characteristics of a successful system, it requires a systematic development process which is able to emphasize the understanding of the scope and complexity of the total development process. In fact, this research is focusing on software development life cycle methodology that is agile development, which is commonly adopted by most system developers. Nevertheless, the system design is the most important aspect of software development. There are several different approaches to software development. Some take a more structured, engineering-based approach to developing business solutions while others may take a more incremental approach in which software evolves as it is developed piece-by piece (32). This research is focusing on the comparison between the agile and ASDF methods.

6. The Agile and ASDF Method

Research by Linda & Norman (33) pointed out that, in today's software development environment, in order to meet fluctuating business demands, the requirements keep on changing during the product development life cycle which results in never-ending headaches for development teams. Thus, a discussion on the experience of implementing the agile software development process to address these concerns is required. For the last two decades, the agile is a most widely practiced method that has been successfully used in software development projects. The methodology has been successfully applied to education, manufacturing and an array of other industries apart from being mostly practiced in a commercial software environment. Similar to other lean methods, agile optimizes limited resources and creates efficiencies. Agile also empowers teams to be free from outside interference that is to self-organize and work at a sustainable pace. This enables the management to focus on the company's vision instead rather than on day-to-day management and at the same time enables teams to achieve their full potential. Over the years, many additional layers of agile practices and patterns have been developed due to the need for agile to address the inevitable complexities of real life.

The analysis system development framework is a development methodology for software application. There are combination of information system and software development to develop the software phases from the client to application requirement. There have six phases to completed the application and consist input for user requirement specification and system requirement specification, process for module, database and user acceptance testing. During this phases, there needs to start the process development, review application and feedback from the client and developer. The output is for final acceptance testing with the approval from the client to release to the market. The structure of analysis system development framework as shown in figure 7.1.

Despite the exact scope of the roles and responsibilities may differ from one system development to another, System Analysis (SA) has been perceived by those involved in system developments as a specific set of responsibilities within a more general process of system transformation and change. Nevertheless, if system analysis is not appropriately positioned within a defined framework that guides the overall change process in an integrated design, the value of system analysis work in software development may be deteriorated.

Certain aspects of a typical analysis change process have associated methodologies. For example, a software development may be guided by Waterfall, Rational Unified Process (RUP), Agile, Dynamic System Development Method (DSDM), Structured System Analysis and Design Method (SSADM) or a host of derivatives and for project management, one might resort to methodologies like PRINCE, among others.

However, it looks like an over-arching framework that covers business change and transformation in a holistic way, right from the expression of a change initiative within a strategic change programme and ending in a delivered change accepted as Fundamental Process of IT (FPIT), with the benefits of the change duly evaluated does not exist until now. Thus, system analysis may be able to be positioned appropriately together with other roles and disciplines if the said framework exists.

Nevertheless, it is a wise move if a general workflow model to be put in place in order to control the process of system change prior to developing a methodology specifically to cater SA. Once the general workflow is in place, it is easier to identify which part of the workflow is relevant and under the purview of System Analyst together with other disciplines involved. This approach, in a way, is in-line with software application development, for example, a Solution Architect is responsible for completing certain tasks together with other experts in order to develop or modify a software product.

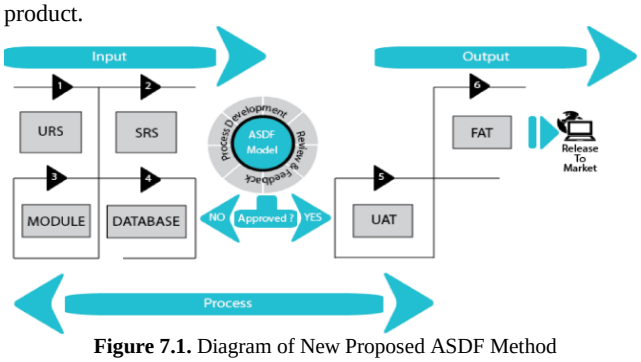

Figure 7.1. Diagram of New Proposed ASDF Method

7. The ASDF Works

The new approach to develop the software is used analysis system development framework (ASDF). The ASDF is being compared to an agile method through informal discussions and interviews with vendors, clients and stakeholders of software development projects. First, it will describe the case study through the use of the questionnaire in conducting a research. The aim of this study is to focus on developers in a public sector, private sector, statutory body as well as students to measure the quality of skills in teamwork. Second, it will present the methodology used to gather primary data that is thematic interviews through the use of a questionnaire. Third, it will describe the samples and respondents participating in this study as well as the interview recruitment techniques used. Lastly, the validity and credibility of thematic interviews through the use of a questionnaire as a qualitative research methodology are evaluated.

Being a new proposed method, analysis system development framework allows projects to be developed more efficiently during the requirement process by having work tracks running in parallel rather than waiting for fully developed requirements that may have little interaction with the industry and little opportunities for mid-course correction. There are heaps of benefits compared to the agile way of requirement elicitation and elaboration. This new proposed analysis system development framework method definitely marks a big change in the way solutions are designed and delivered in shorter time frames by various organizations. The comparison is shown in Table 7.1 below.

Table 7.1: Comparison of ASDF and Agile Models

Agile Model	ASDF Model
In the case of some software deliverables, especially for a short time frame project of over three "3" months, it is difficult to assess the effort required at the beginning of the system development.	An adaptive teamwork which is able to respond to the changing requirements.
Lack of emphasis on necessary designing and documentation.	The teamwork does not have to invest time and effort, by the time they deliver the product, it is found that the client's requirement has changed. The documentation is crisp which results in time saving.
The project has the tendency to be off track if the client representative is not clear with what is expected to be the final outcome at the end of the project.	Face to face communication and continuous inputs from the client representative leave no space for guesswork.
Only senior programmers are capable of making the required decisions during the development process. Hence there is no place for newbie programmers unless with the existence of experienced resources.	The end result is high-quality skills of teamwork in the least possible time frame and satisfied client.

8. Conclusion

This research of agile and new proposed ASDF method is to measure the quality product in software development this preliminary research to do the comparison of the best methodology to used for developer. Therefore, for determine the project development is on tie in order to control and increase the quality of skills in a teamwork collaboration and understanding in software development.

Despite the emphasis on teamwork quality in the agile community, in the traditional and the agile surveys alike, both the evaluation of teamwork quality it self and its effect on team performance and team members' success were similar. However, the agile survey showed lower agreement among the ratters regarding evaluation of team performance than was the case in the traditional survey. An implication of this survey is that the quality of teamwork is a major factor in improving team performance, especially for improving the quality of the team's product. Note that when trying to optimize team performance, one needs consensus of whose view of team performance should be considered. For the future, we recommend that more research efforts be made to validate the teamwork quality construct and to advance the measurement of team performance with to enhance the agile method by proposing a new ASDF software development method.

References

[1] Agarwal, N., and Rathod, U., 2006. Defining success for software projects: An exploratory revelation. International Journal of Project agile transformation. Information and Software Technology 53(3). 276–290.

[2] Ali, Munassar, N., M., and Govardhan, A., 2010. A Comparison Between Five Models of Software Engineering. IJCSI, Vol.7, Issue5. approach with agile practices, Empire Software Engineering 15:654– 693. Approaches School of Engineering and Information Technology Universiti Malaysia Sabah, Malaysia. between two new product development teams. Management Science, 38(3), 360-373.

[3] Bacon, N., and Blyton, P. (2006). Union co - operation in a context of job insecurity:

[4] Cockburn., A., and Highsmith, J., 2001. Agile software development: The people factor. Software Management, 34(11), 131-133.

[5] Mulika. (2010). The Impact of Teamwork on Employee Performance in Strategic Management and the Performance Improvement Department of Abu Dhabi Police, UAE.

[6] Pfaff, E., and P. Huddleston. (2003). Does it matter if I hate teamwork? What impacts student attitudes toward teamwork. *Journal of Marketing Education*25:37–45.

[7] Froebel, P., and Marchington, M. (2005). Teamwork structures and worker perception: a cross na tional study in pharmaceuticals , International Journal of Human Resource Management, 16(2), 256 - 276.

[8] Ingram, H. (2000). Lin king teamwork with performance. Journal of Team Performance Management, 2(4), 5 - 10.

[9] Washer P. Designing a system for observation of teaching. Quality Assurance in Education. 2006 Jul 1;14(3):243-50.

[10] Conti, B., and Kleiner, B. (2003). How to incr ease teamwork in organizations. Journal of Quality, 5(1), 26 - 29.

[11] Langfred, C. W. (2000). The paradox of self-management: Individual and group autonomy in work groups. *Journal of Organizational Behavior*, *21*(5), 563-585.

[12] Hoegl, M., & Gemuenden, H. G. (2001). Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. *Organization science*, *12*(4), 435-449.

[13] Salas E, Sims DE, Burke CS. Is there a “big five” in teamwork?. Small group research. 2005 Oct;36(5):555-99.
[14] Dickinson, T.L., and McIntyre, R.M. 1997. "A Conceptual Framework of Teamwork Measurement," in Team Performance Assessment and Measurement: Theory, Methods, and Applications,M.T. Brannick, E. Salas and C. Prince (eds.). NJ: Psychology Press, pp. 19–43.
[15] Kraut, R.E., Streeter, L.A.: Coordination in software development. Commun. ACM **38**(3), 69–81 (1995)
[16] Diefenbach GJ, Reitman D, Williamson DA. Trichotillomania: a challenge to research and practice. Clinical Psychology Review. 2000 Apr 1;20(3):289-309.
[17] Dingsoyr, T., and Lindsjorn, Y., 2013. Team Performance in a agile development teams: findings from 18 focus groups. Agile Process in Software Engineering and Extreme Programming, 14th International Conference, XP 2013, Vienna, Austria, June 3-7. Proceedings, 40-60, doi: 10.1007/978-3-642-38314-4-4.
[18] Dyba, T., & Dingsoyr, T. (2008). Empiric.al studies of agile software development: A systematic review. *Information and software technology*, *50*(9), 833-859.
[19] Stray, V. G., Moe, N. B., & Dingsøyr, T. (2011, May). Challenges to teamwork: a multiple case study of two agile teams. In *International Conference on Agile Software Development* (pp. 146-161). Springer, Berlin, Heidelberg.
[20] Rodd, H., Wray, A., Wilson, N. H., & Hosey, M. T. (2006). *Treatment planning for the developing dentition*. Quintessence.
[21] Liang, T. P., Wu, J. C. H., Jiang, J. J., & Klein, G. (2012). The impact of value diversity on information system development projects. *International Journal of Project Management*, *30*(6), 731-739.
[22] Henderson, J. C., & Lee, S. (1992). Managing I/S design teams: a control theories perspective. *Management science*, *38*(6), 757-777.
[23] The Standish Group (2009). Chaos Summary Report: The 10 laws of CHAOS. Technical Report. Boston, MA: The Standish Group International.
[24] Faraj, S., & Sproull, L. (2000). Coordinating expertise in software development teams. Management Science, 46(12), 1554-1568.
[25] Khalaf, S. J., & Al-Jedaiah, M.N. (2008). Software Quality and Assurance in Waterfall Model and XP A Comparatice Study. Iss 2, Vol 7.
[26] Highsmith, J., (2002). Agile Software Development Ecosystems. The Agile Software Development Series, ed. Boston, MA: Addison-Wesley.
[27] Wateridge, J. (1995). IT Projects: A Basis for Success. International Journal of Project Management, 13(3), 169-172.
[28] Alie, R.E., Beam, H., & Carey, T.A. (1998). The use of teams in an undergraduate management program. Journal of Management Education, 22(6), 707-19.
[29] Hartenian, L. S. (2003). Team member acquisition of team knowledge, skills, and abilities. *Team Performance Management: An International Journal*, *9*(1/2), 23-30.
[30] Jones, A., Richard, B., Paul, D., Sloane, K., and Peter, F., 2007. Effectiveness of teambuilding in organization. Journal of Management, 5(3), 35-37.
[31] Yu Beng Leau, Wooi Khong Loo, Wai Yip Tham & Soo Fun Tan. (2012). Software Development Life Cycle Agile vs Traditional Approaches School of Engineering and Information Technology University Malaysia Sabah, Malaysia.
[32] Deepshikha, J., (2010). Analysis of Software Quality Models for Organizations University of Jammu Department of Computer Science & IT.
[33] Linda, R., & Norman, S. J., (2000). The Scrum Software Development Process for Small Teams. AG Communication Systems.