

Exploring the missing framework of Agile Software Development Framework through Adaptation

Kevin Suryaatmaja, Dermawan Wibisono, Achmad Ghazali, Rachma Fitriati

Abstract: *This study focuses on Agile software development implementation in Indonesia, since it is exciting to learn about the growing of Agile software development's user lately, although many of them failed to implement it. Empirical findings in "PREEMINENCE" corporation reveals that several key people who had using the Agile software development mentioned several issues to implement it, which relates to the adaptation of using Agile software development. The previous study had resolved this issue and initiated an additional approach to apply Agile software development, but none of them had investigated from the learning process perspective. Thus, the research gap exists around Agile implementation refers to the process of constructing the adaptation model, to maintain the learning process in the organization. This study offers another perspective to implement Agile software development by using an adaptation strategy as accommodation for the early adopter, which also promotes the learning in the organization. Moreover, indicating the Transactive Memory System as a potential theoretical foundation for this study and Soft System Methodology as potential research methodology, to further promote the Agile software development literature by providing propositions that guide future research.*

Keywords: *Agile, Scrum, Software Development, Software Development Life Cycle, Scrum, Transactive Memory System, Soft System Methodology, Waterfall*

I. INTRODUCTION

The Indonesian insurance business is experiencing its promising moment as it continues to grow from the year 2011 – 2017 [1]. This triggers rapid competition in Indonesian insurance companies to innovate and deliver new products faster to the consumer. Therefore, the development strategy of new financial products and services become crucial for all corporations.

Author validates the business growth to a smaller scope, refer to a company name "PREEMINENCE" as a research subject. This corporation has been striving to strive for the company's vision to be the world's preeminent life insurance provider by conducting several strategies either in

products launching, projects initiatives, and many software enhancements. In 2016 there were more than 40 products launched and more than 200 system enhancements to cater the business needs.

To deal with many projects launched with a tight target date and many inter-correlated systems to developed, all resources were highly utilized to meet projects' deadline within the allocated budget. In early 2016, the Information Technology department had launched the Agile software development (SD), hoping to answer those concerns. It developed to address the core principles of the Agile Manifesto which are collaborative development, reduction of unnecessary work, minimizing wasteful documentation, so that customers/stakeholders are actively shaped and guiding the software development, and to control variation of output [2-4].

Therefore, Agile software development (SD) is used to answer the concern of 'PREEMINENCE' corporation and getting a lesser bug rate, improve customer satisfaction, lesser development cycle, and quicker adaptation for apace changing business requirement [5 – 8].

Based on the study of Agile SD implementation at "PREEMINENCE" company, it has found that the team should focus on specific requirements and not overall requirements which lead to the silo. Another challenge is shorter cycle time from requirement gathering to software implementation, as business users are often changing their requirements. They are thus impacting the learning process, since it is difficult to learn a particular or related function. Moreover, system enhancement related to a specific feature handled by the same person who manages it before, which impact the dependency of certain people. This condition worsens because limited documentation leads to dependence of team's tacit knowledge. Therefore, the issues in this company are Agile SD implementation relates to people, knowledge, and learning process.

To understand more the Agile usage and implementation in the bigger scale and scope, refer to the Chaos Report in 2015 which reflected some findings of 10,000 projects across the world. It underlines the fact that not all projects succeed, even though using the Agile SD (Figure 1). This fact has brought the author's interest to understand why there is still a gap in the Agile SD implementation, although it has been known more than a decade with its vision to bring more successful software implementations.

Revised Manuscript Received on April 07, 2019.

Kevin Suryaatmaja, School of Business and Management, Institut Teknologi Bandung Jl. Ganesha No. 10, Bandung 40132, Indonesia

Dermawan Wibisono, School of Business and Management, Institut Teknologi Bandung Jl. Ganesha No. 10, Bandung 40132, Indonesia, Universitas Pertamina Jl. Teuku Nyak Arief, Simprug, Jakarta 12220, Indonesia

Achmad Ghazali, School of Business and Management, Institut Teknologi Bandung Jl. Ganesha No. 10, Bandung 40132, Indonesia

Rachma Fitriati, Fakultas Ilmu Administrasi, Universitas Indonesia, Gedung M, FISIP UI, Depok, 16424, Indonesia

Exploring the missing framework of Agile Software Development Framework through Adaptation

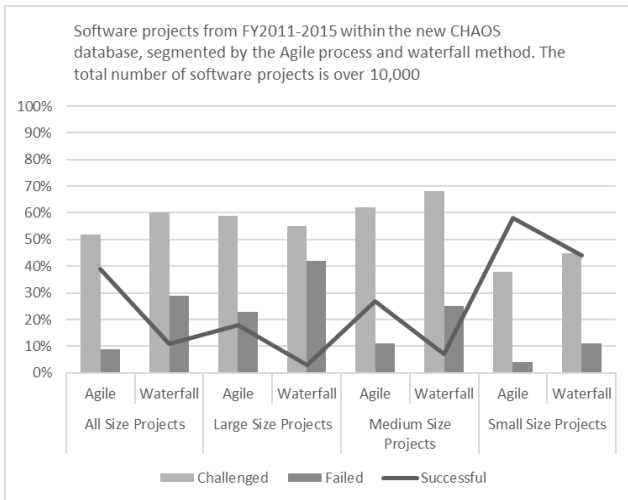


Fig. 1 Comparison Projects Delivery using Agile and Waterfall Source: Chaos Report (2015)

Refer to the previous paragraph, there are some empirical findings in the Agile implementation of 'PREEMINENCE' company and strengthened by the global facts that the project's failure still exists, despite using Agile SD [9].

In general, the outcome of this study is to provide a better Agile SD implementation, especially in the insurance business where projects have been "addicted" or become a habit of using traditional Waterfall SD for sometimes. Shifting to the Agile SD is not easy to implement by following the books or based on training. Therefore, a proper analysis shall be conducted to explore the interaction between the team to gain and transfer knowledge for a sustainable learning process, which is become vital for successful implementation of Agile SD.

The outcome of this study is also to support the company to improve its practices in software development and implementation by proper change management. Thus, able to develop team knowledge by learning process intervention, although some projects handled by a different team with many inter-correlated systems.

This study begins with the explanation of SDLC as the general framework used by software development projects and to provide further proof of Agile SD. After that, a description of the learning process that occurs during project execution using Agile SD. Then, continue to the explanation of Agile implementation findings by previous study and framework needed to fill the gaps. Lastly, exploration of research method using SSM-based action research (AR) to understand human as social actors.

II. LITERATURE REVIEW

Conger [10] mentioned software requirements have some aspects of simplicity and learnability, thus has the knowledge to be learned [11]. Furthermore, software development involves many varieties of unique knowledge which been highly utilized to solve the business problem [12-16]. A complex and short time problem-solving process can be catered by software development [17].

Conboy *et al.* [3] stated that people's contribution to the Agile process is the most critical factor besides technology and process/task, which contributes to the failure of software project if ignored [17-20].

In software development, the engagement between the development team and customers is vital to disseminate the understanding as the development team could have a better understanding through collaboration with customers [17]. Chauet *et al.* [15] strengthened the statement that the knowledge dissemination through the discussion involves individual mental model and skills, and these reflecting universal concepts and understanding.

Knowledge and Learning in the Organization

Vidgen and Wang [21] stated the outcome of close interaction between the member of a team is emersion of the team learning, which is critical for the evolution of an organization. Senge [22] also mentioned the importance of team learning as capacity alignment and development of the team to create results. Therefore, the close and frequent interaction between people (customer and system programmer) is essential for effective knowledge dissemination to achieve a common understanding. At this point, we can conclude that people and the way they interact with others during the learning process are the most crucial factors in the Software Development process which become the focus of this study.

Nonaka [23] mentioned as new knowledge always commence from an individual, this made them as an initiator of the learning process. Hence, the critical activity of a knowledge creating company is to make the personal knowledge visible to others [24]. Argote & Miron-Spektor [25] mentioned that Organization Learning could manifest by a change in cognition and include explicit and tacit knowledge. These knowledge could embed in various repository including people, routines, and TMS where people's memory always under construction by the cumulative history of interaction [26].

Transactive Memory System (TMS)

Lewis *et al.* [27] argued that their "TMS-learning framework" applies to a set of people that perform complicated and fragmented tasks which expect significant knowledge. Which in general, the framework prevails to workgroups that guided by task where the responsibility shared among the members as a trigger to generate group outcomes. The outcome's performance relies on the coordination and integration of activities, knowledge, and skills of a group member.

The first learning cycle of the "TMS learning framework" consists of the initial TMS structure that regulates the storage of knowledge from memories of the member and a collection of transactive processes used by members to encode, store, and retrieve existing knowledge [28]. The development of TMS commences when the particular area of individual knowledge, began to be associated by members of the group. This association information is stored in the TMS structure and identified as location information [28].

TMS structure also stores particular information and details (lower-order information) in which a specific topic exhibited by each member [28].

TMS structure influence the learning content and scope from each member, which award every member a choice to learn in a specific area and become specialized in different areas of knowledge than other member's area [29, 30]. This process develops a condition that every member is responsible in every respective area, thus creates a specialization for a member who is accountable for the particular area [31, 32].

The second component of TMS is Transactive Processes [28], established through early interactions of a group and each member communication to encode, store, and retrieve knowledge which pertains to the task belonging to the group. At the point when communication initiated by members, they depend on the underlying TMS structure to build up transactive processes by utilizing location information to acquire data rapidly and productively once required for the undertaking [27]. Transactive processes are serving members to acquire a more precise understanding of what members apprehend or not apprehend, and recurrent interactions invoke members' location information progressively comparable and precise. Moreover, transactive process influences the TMS structure by helping the member to seek out the linkages of knowledge among members and to a group-generated solution as [28] mentioned "integrations of members' knowledge".

TMS structure led to collective knowledge discovery by integrating relevant details and converted as common higher-order information, characterized as the substance of lower-order information [28], which uncovers the connections among TMS and individual, and collective learning [27]. Therefore, TMS is very helpful for an organization to collect, encode, store, and retrieve knowledge. TMS is also helping the organization to connect the details knowledge of every member and general order information, which the outcome will bring a relevant topic based on general information.

III. EXISTING FRAMEWORK TO SUPPORT AGILE SOFTWARE DEVELOPMENT

Authors deliberately discover 22 research papers discuss the Agile implementation framework with the support of adaptation or adaption. All papers were not able to explore the transformation of experience into knowledge sharing, the creation of common meanings which extrapolate from collective experiences, analysis, informed interpretation, reflection, accessible memory, and processing information. Therefore, this study takes a position of novelty focusing on Agile SD implementation by Learning Process intervention.

Although some researcher had already developed a framework to help the organization to adopt Agile SD such as [33-35] which focuses on Knowledge Management, Human Behavior, and Learning. However, the framework only to evaluate the adoption of critical factors to achieve the highest agility but doesn't reflect 'How' the framework built and doesn't reflect 'How' to accomplish, every stage. Even more, the framework used in the research is a conceptual model only which have not applied in the study.

McAvoy and Butler [36] developed a framework which becomes an ideal reference for this study because besides focus on Human Behavior, Knowledge Management, and

Learning; the framework in the research applied as an operational model. The drawback of the research is that the framework only discusses individual not team and does not reflect 'How' it can be embedded in the Agile process. Even more, the research approach used a deductive approach by verifying the concept of behavioral change by [37, 38].

IV. PROPOSED FRAMEWORK TO SUPPORT AGILE SOFTWARE DEVELOPMENT

An organization with well-developed TMS can enhance problem-solving and coordination, as the member knows to whom they go for advice. This explains how team members can coordinate and learn between expertise for accomplishing collective goals [39]. They also can perform better on recalling tasks for high levels of TMS because the individuals exhibit the encoding, storing, and retrieving knowledge to share the cognitive system [40]. Therefore, TMS can answer the needs of problem-solving, coordination, and recall task that uniquely distributed within the individual as a group of experts.

Communication is considered as a vital part in TMS development since it utilizes the individual expertise [41], how people learn, share, and unite information from others [31] to exchange task-related information and relational information [40]. Furthermore, face to face communication is more advantageous than non-face to face communication [42]. Therefore, TMS can answer the needs on face to face communication as a basis to learn and share information with others.

From the above paragraphs, TMS acts a critical function to accommodate the learning process within the organization. The learning process of expert group by exchanging knowledge that always evolves, which needs a mechanism to manage the process to avoid the problem of knowledge is not updated. Face-to-face discussions on a task that involve information exchange with minimal documentation, which requires a mechanism to recall what has been discussed previously to avoid the problem of the task that is not updated. Therefore, TMS can resolve the potential issues using Agile SD where information and knowledge reside in the brain of experts which become highly depends on expert. TMS can support face-to-face communication as the primary method in Agile SD for discussing the task update that always evolves daily.

Soft system methodology fits in the purpose of organizing the exploration of learning from a complex system. The observer perceived a real world as a messy and ill-defined problem, to solve this by organizing exploration as a learning system [43]. SSM-based AR considers the experience-based knowledge methodologically, thus need to reinterpret SSM from "actuality" point of view [43]. SSM-based AR neglect the existence of system or structure in "reality" but interest on the prospect to accommodate the people interest in "actuality" [44]. The word "accommodation" refers to live with different individual's worldviews based on sharing "actuality", therefore "accommodation" process only can be designed or shaped during the process of SSM [43].

Exploring the missing framework of Agile Software Development Framework through Adaptation

To conduct SSM-based AR, the research model must be designed to enable the generation of knowledge about the theoretical framework and real-world problem situation to answer research questions and generate new insights. TMS should be performed as a theoretical framework as a catalyst to later proven the construction of learning process based on the comprehension of a human activity system as a complex multi interaction.

V. CONCLUSION AND FURTHER WORK

The implementation of Agile SD in an organization is proven requires considerable approach in adaptation, which if neglected will have an impact on the learning process. Using SSM as a technique for action research is expected to resolve this issue by constructing the adaptation model, to maintain the learning process in the organization. TMS plays a vital role as a catalyst within organizations to support groups of experts in sharing information and knowledge through face-to-face communication. TMS framework corresponds to one (1) of ritual in Agile SD, called “daily scrum”. That is to refresh and maintain knowledge when time to communicate is limited with minimal documentation. Future research is expected to use TMS as a theoretical framework to support SSM-based Action Research.

ACKNOWLEDGMENT

The research work for this study was initiated in a Life Insurance company “PREEMINENCE” by the approval of the Chief of Technology. The Research Question, Literature Review, and Research Methodology was formulated together with my supervisor, Prof. Dr. Dermawan Wibisono, and Achmad Ghazali, Ph.D. to support the extensive guidance during the process. I want to express also my thankfulness to Dr. Rachma Fitriati for helping my research methodology especially introducing the Soft System Methodology.

REFERENCES

1. Financial Services Authority of Indonesia, Indonesian Insurance Statistic, <https://www.ojk.go.id/id/kanal/iknb/data-dan-statistik/asuransi/Pages/Statistik-Perasuransian-Indonesia---2016---Revised-Version.aspx>, (30 July 2018). (URL Link)
2. Dingsøyr, T., Nerur, S., Baliyepally, B.G., A decade of agile methodologies: Towards explaining agile software development, 2012. *The Journal of Systems and Software*, 85(6), pp.1213– 1221. (Journal)
3. Conboy, K., Coyle, S., Wang, X.F., and Pikkarainen, M., People Over Process: Key People Challenges in Agile Development, 2010. *IEEE Software*, 28(4), pp.48-57. (Journal)
4. Conboy, K., Agility from first principles: reconstructing the concept of agility in information systems development, 2009. *Information Systems Research*, 20(3), pp.329–354. (Journal)
5. Cho, J. J., An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum, 2010. All Graduate Theses and Dissertations. (Thesis or Dissertation)
6. Parrish, A., Smith, R., Hale, D., and Hale, J., A field study of developer pairs: Productivity impacts and implications, 2004. *IEEE Software*, 21(5), pp.76-79. (Journal)
7. Miller, K., and Larson, D., Agile software development: Human values and culture 2005. *IEEE Technology and Society Magazine*, 24(4), pp.36-42. (Journal)
8. Boehm, B. and Turner, R., Using risk to balance agile and plan-driven methods, 2003. *IEEE Computer*, 36(6), pp.57-66. (Journal)
9. The Standish Group, Chaos Report, 2015. [https://www.infoq.com/articles/standish-chaos-2015\(30](https://www.infoq.com/articles/standish-chaos-2015(30) July 2018). (URL Link)
10. Conger, Software Development Life Cycles and Methodologies: Fixing the Old and Adapting the New, 2011. *International Journal of Information Technologies and Systems Approach*, 4(1), pp.1-22. (Journal)
11. Nielsen, J., Usability Engineering. San Diego, CA: Kaufmann, 2000. (Book)
12. Apshvalka, D., Donina, D., and Kirikova, M., Understanding the Problems of Requirements Elicitation Process: A Human Perspective. In: Wojtkowski W., Wojtkowski G., Lang M., Conboy K., Barry C. (eds) *Information Systems Development*, 2009. Galway: Springer, pp.211-223. (Journal)
13. Richter-von Hagen, C., Ratz, D., and Povalej, R., Towards self-organizing knowledge-intensive processes, 2005. *Journal of Universal Knowledge Management*, 2, pp.148–169 (Journal)
14. Tiwana, A., An empirical study of the effect of knowledge integration on software development performance, 2004. *Information and Software Technology*, 46(13), pp.899-906. (Journal)
15. Chau, T., Maurer, F., and Melnik, G., Knowledge Sharing: Agile Methods vs. Tayloristic Methods, 2003. In: *Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. Linz, Austria, 9-11 June 2003. Washington: IEEE Computer Society. (Journal)
16. Hefke, M. and Trunko, R., A Methodological Basis for Bringing Knowledge. Management to Real-World Environments. In: *Practical Aspects of Knowledge Management*, 4th International Conference. Vienna, 2-3 December, 2002. Vienna: Springer. (Journal)
17. Ozer, M. and Vogel, D., Contextualized Relationship between Knowledge Sharing and Performance in Software Development, 2015. *Journal of Management Information Systems*, 32(2), pp.134-161. (Journal)
18. Coyle, S., Conboy, K., and Acton, T., Group Process Losses in Agile Software Development Decision Making, 2013. *International Journal of Intelligent Information Technologies*, 9(2), pp.38-53. (Journal)
19. Thomas, R. R., Learning from HOMER: a Case-Based Help Desk Support System, 2004. Berlin: Springer. (Journal)
20. Schmidt, R., Lyytinen, K., Keil, M., and Cule, P., Identifying software project risks: An international study, 2001. *Journal of Management Information Systems*, 17(4), pp.5–36. (Journal)
21. Vidgen, R., and Wang, X., Coevolving systems and the organization of agile software development, 2009. *Information Systems Research*, 20(3), pp.355–376. (Journal)
22. Senge, P., The fifth discipline: The art and practice of the learning organization, 2006. London: Random House. (Book)
23. Nonaka, I, The knowledge-creating company, *Harvard Business Review*, (July - August 2007), <https://hbr.org/2007/07/the-knowledge-creating-company>, (24 November 2018). (URL Link)
24. Wang and Ahmed, Organisational Learning: A Critical Review, 2003. *The Learning Organization*, 10 (1), pp.8-17. (Journal)
25. Argote L. and Miron-Spektor E., Organizational learning: From experience to knowledge, 2011. *Organization Science*, 22(5), pp.1123-1137. (Journal)
26. Ertmer, P.A, Newby, T.J., Behaviorism, Cognitivism, Constructivism: Comparing Critical Features from an Instructional Design Perspective, 1993. *Performance Improvement Quarterly*, 6(4), pp.50-72. (Journal)
27. Lewis, K., Lange, D., and Gillis, L., Transactive Memory Systems, Learning, and Learning Transfer, 2005. *Organization Science*, 16(6), pp.581-598. (Journal)
28. Wegner, D.M., Guiliani, T., and Hertel, P.T., Cognitive interdependence in close relationships. J. Ickes, ed. *Compatible and Incompatible Relationships*, 1985. Springer-Verlag, New York, pp.253–276. (Journal)
29. Hollingshead, A. B., Cognitive interdependence and convergent expectations in transactive memory, 2001. *Journal of Personality and Social Psychology*, 81(6), pp.1080–1089. (Journal)
30. Wittenbaum, G.M., Vaughan, S.I., and Strasser, G., Coordination in task-performing groups. R. S. Tindale, L. Heath, J. Edwards, E. J. Posvoc, F. B. Bryant, Y. Suarez-Balcazar, E. Henderson-King, J. Myers, eds. *Social Psychological Applications to Social Issues: Theory and Research on Small Groups*, 4, pp.177–204. (Journal)

31. Hollingshead, A. B., Communication, learning, and retrieval in transactive memory systems, 1998. *Journal of Experimental Social Psychology*, 34(5), pp.423–442. (Journal)
32. Wegner, D.M., Erber, R., and Raymond, P., Transactive memory in close relationships, 1991. *Journal Personality and Social Psychology*, 61(6), pp.923–929. (Journal)
33. Qumer, A. and Henderson-Sellers, B., A framework to support the evaluation, adoption and improvement of agile methods in practice, 2008. *Journal of Systems and Software*, 81(11), pp.1899-1919. (Journal)
34. Heje, J.P. and Baskerville, R., The translation and adaptation of agile methods: a discourse of fragmentation and articulation, 2017. *Information Technology & People*, 30(2), PP.396-423. (Journal)
35. Hoda, R. and Noble, J, Becoming Agile: A Grounded Theory of Agile Transitions in Practice, 2017. IEEE/ACM 39th International Conference on Software Engineering, 2017. IEEE/ACM, 39th International Conference on Software Engineering, pp.141-151. Buenos Aires, 20–28 May, 2017. New Jersey: IEEE Press. (Conference)
36. McAvoy, J. and Butler, T., A Failure to Learn by Software Developers: Inhibiting the Adoption of an Agile Software Development Methodology, 2009. *Journal of Information Technology Case and Application Research*, 11(1), pp.23-46. (Journal)
37. Argyris, C., Single-loop and double-loop models in research on decision making, 1976. *Administrative Science Quarterly*, 21(3), pp.363-375. (Journal)
38. Argyris, C., Double loop learning in organizations, 1997. *Harvard Business Review*, 55(5), pp.115-125. (Journal)
39. Wegner, D. M., Transactive memory: A contemporary analysis of the group mind, 1986. B. Mullen, G. R. Goethals, eds., *Theories of Group Behavior*. Springer-Verlag, New York, pp.185–208. (Journal)
40. Liao, J., Jimmieson, N.L., O' Brien, A.T., and Restubog, S.L.D., Developing Transactive Memory Systems: Theoretical Contributions from a Social Identity Perspective, 2012. *Group & Organization Management*, 37(2), pp.204–240 (Journal)
41. Stasser, G., W. Titus, Hidden profiles: A brief history, 2003. *Psychological Inquiry*, 14(3/4), pp.304-313. (Journal)
42. Lewis, K., Knowledge and performance in knowledge-worker teams: A longitudinal study of transactive memory systems, 2004. *Management Science*, 50(11), pp.1519–1533. (Journal)
43. Checkland and Poulter, Learning for Action, A Short Definitive Account of Soft System Methodology and its use for Practitioners, Teachers, and Students, 2006. (Book)
44. Uchiyama, K., A concise theoretical grounding of action research: based on Checkland's soft systems methodology and Kimura's phenomenological psychiatry. The Institute of Business, Daito Bunka University, Tokyo, 2009. (Journal)