International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-10 Issue-4, February 2021

A Social Network Scrum Framework for Software Development under COVID-19 Pandemic

Nalinee Sophatsathit

Abstract*: This research proposes a framework for social network scrum meeting that serves as an alternate means for work continuation under the COVID-19 pandemic. Conventional agile and scrum methods that require in-person meeting on daily basis, as well as scrum process become impractical under stringent 'social lockdown' mandates. To prevent any disruptive discontinuity, the proposed framework sets up an online meeting to replace the in-person stand-up meeting and scrum. Some supporting practices are also established to adjust both agile and scrum event flows that suit this online encounter. They are production development setup and social network meeting. The former offers industrial practices that are well entrenched and proven, while the latter has been used extensively in this digital age. The proposed method is tested with computer science student's projects. Students are able to continue their meeting, discussion, and some outputs rather than being isolated with no fruitful outcome. The proposed method does establish some ground work to be explored for future software development environments that will suit to the imminent digital technological advancement.

***Keywords*: Agile, COVID-19 Pandemic, Scrum, Social Network meeting.**

I. INTRODUCTION

Software development is a rich endeavor that encompasses many activities in different domains. From complex and dynamic spectrums of work, these activities culminate software product, operations, maintenance, and support. Many companies attempt to get the product done using a systematic development process in the form of software project, where a diversity of models is employed, for example, waterfall, spiral, V, prototype, and iterative models, just to name a few. Unfortunately, some projects fail due to various factors [11-14]. The failure might be missing deadlines, not meeting the scope or requirements, or budgeting problem. Notwithstanding these factors, human is the culprit that is conducive toward all failures. Organizations are working to arrive at systematic process standards such as International Organization for Standardization (ISO) 9001 and Capability Maturing Model Integration (CMMI) to establish guidelines for process improvement. Despite such contributing establishment, small companies which constitute the majority of software development community do not adopt these standard practices since they are too costly to implement.

The advent of agile model [3, 6] has brought about new paradigm of project and team development. There are pros and cons concerning the agile model at work which can be looked up in many literatures. One of the important concepts of this model is Scrum [15] which requires the development team to gather together for a short meeting. The outcome of the meeting will be a decision on some assignment, action, consensus, etc. This is a usual practice in projects that adopt the agile method. The success and failure of project using agile method depends primarily on all team members' communication during scrum. The objective of this research is to investigate alternate means for scrum meeting in the presence of COVID-19 pandemic. The epidemiological measures make it difficult to resort to proper meeting formats of scrum that is effective, yet abiding by those preventative regulatory guidelines. We propose a social network scrum meeting (SNSM) for an agile development team and conduct some preliminary experiments gauging the performance of SNSM as oppose to conventional in-person gathering. Performance statistics were tallied to validate the viability of the proposed method that could serve as future software environment work practice. The organization of this paper is as follows. Section 2 describes some related work to this research. Section 3 sets up the proposed method to be applied with a class project. The accompanying experiment is elucidated in Section 4. Some relevant findings and course of action are discussed in Section 5. Section 6 concludes this work with some future prospectus.

II. RELATED WORK

The studies of software development process have been extensive as software proliferates in this digital age. A number of new development processes have been proposed to supplement or even substitute the traditional waterfall model for reasons of better software product and timely delivery. One major method is agile development. Others in the same vein are scrum, crystal, Extreme Programming (XP), Feature Driven Development (FDD). We will look into some relevant prior works that would be embraced and applied to the proposed SNSM framework. For brevity, we will investigate the bases of SNSM, i.e., agile method, scrum, and team and development framework.

Manuscript received on February 08, 2021.
Revised Manuscript received on February 17, 2021.
Manuscript published on February 28, 2021.
* Correspondence Author
Nalinee Sophatsathit*, Computer Science Program, Faculty of Science and Technology, Suan Sunandha Rajabhat University, Bangkok, 10300, Thailand. Email: nalinee.so@ssru.ac.th

A. Agile Method

Moreira et al. [16] introduced agile in a succinct and straightforward way as follows:

"Agile is all about empowering the team and getting closer to what the customer wants. In place of rigorous upfront planning and the phase-based process, it offers a dynamic, iterative build-and-test cycle, where change is handled well. One of Agile's hallmark features is that it drives the decision-making process lower in an organization, making that organization more responsive and adaptive."

The Agile Manifesto spells out well known foundation principles of agile method [18]. There are many benefits that are precipitated from the agile practice. However, Broman [17] identified a number of agile pitfalls, namely, organization culture, people and process challenges, resource allocation, insufficient planning due to too early delivery, etc. We will exercise the agile practice with care within our domain of research.

B. Scrum

Schwaber and Sutherland [7] defined scrum as follows:

"Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems."

The authors succinctly described the entire scrum fundamentals and organization that made up scrum practice. A general framework of scrum outlined by Rubin [6] encompassed: (1) Grooming, including Product backlog, (2) sprint planning, (3) sprint backlog, (4) sprint execution, including daily scrum, (5) partially shippable product increment, (6) sprint review (inspect, adapt), and (7) sprint retrospective (inspect, adapt). All these steps constituted a flexible flow system that iterated till proper software product was obtained.

C. Team and Development Framework

The fact that Scrum is intended for managing software development project does not mean its applicability is confined to this domain. Marchesi et al. [8] implemented the EURACE distributed scrum in a research project reaffirmed that the technique could well be applied to working group of people to achieve a common task. There were several roles in the group that some of them could be adopted in our experiment, namely, Project Owner, Scrum Master, Unit Coordinator, Unit Members, and Research Unit. After composing the team, defining, managing, and improving knowledge work could follow kanban system summarized by Anderson [1] which served as a delivery flow system of work in progress. The team performance would then be improved with the help of a coach described by Shamshurin et al [9]. Molokken-Ostvold and Jorgensen [5] studied group process in effort evaluation of team performance with the help of experts. However, they also pointed out that people were bias and prone to error. Moreover, accurate performance measurement might not be needed as contended by Shepperd and MacDonell [4]. We opted to gauge some empirical results to support our prospectus. Soares and Meira [2] presented a transition strategy from traditional development to agile method that would be in compliance with CMMI project management practices in software organizations. As such, we could smoothly move from conventional waterfall model to agile method within CMMI guidelines. These prior works will be adapted to the proposed SNSM which upholds on-line scrum for the software development team.

D. Online Communication—Social Network

The advent of social network changes human behavior in many respects. From face-down society to live communication, people connect to others in ways no one can imagine. Baruah [19] recounted the use of social network/media as a major communication tool by people, businesses, education, and the likes. The majority of use was interactive media. This tool offered many advantages and disadvantages such as sharing of ideas, low costs/cost effective, less time consuming, etc. Yang et al. [21] analyzed human network, constructed the sociograms, and mined them to demonstrate the small world phenomenon based on four aspects, namely, density, accessibility, centrality, and block model. These showed the interpersonal communication and closeness degree among subgroups and members. We could exploit these prior works to support the evolution of relationship among software development team members. Santos and Sampaio [22] studied on creation and diffusion of information and knowledge in software maintenance and project execution over time. This revolutionized the software industry to incorporate managing of an integrated social-based environment that would support a transition from usual in-person work environment to virtual and distance environment as described by Santos et al. [20]. This research will exploit the social network as a means for team communication.

III. PROPOSED SOCIAL NETWORK SCRUM FRAMEWORK

The proposed framework combined and transformed the relationships of basic and advanced process areas of agile strategy for implementing CMMI project management practices [2] and focused on three domains of research, i.e., planning, monitoring, and organizing of event flow system. The proposed framework will be performed as follows:

1. Proposed framework which will adapt agile method established by prior works,
2. Social network scrum which will replace the traditional in-person meeting, and
3. Performance measurement which will incorporate production development measurements such as inventory, batch size, and velocity.

Relevant areas to the development domains are product backlog, sprint review, sprint retrospective, daily scrum, and work in process (WiP) [6]. Details are described in the sections that follow, namely, the proposed agile framework, social network scrum, and performance measurement.

A. Proposed Agile Framework

The proposed framework intends to be simple and flexible in accordance with the agile method.

Figure 1 depicts the combined relationships of basic agile process areas and advanced process areas established by Soares et al. [2]. Due to the limited applicability of local source of information, where software consultants participate in the research on a voluntary basis, we narrow the scope to encompassing Agile Project Monitoring (APM), Agile Project Planning (APP), and Agile Organizational Project Management (AOM). They are enclosed in the dotted frame which will suit the subsequent experiment to be described in Section 4. The activities involved in the event flow system are sufficient to carry out the information exchanges in a face-to-face team member gathering that include team meeting, document transfers, and discussions. Unfortunately, the COVID-19 pandemic prevents all these to take place. An alternate communication means must be set up to handle those activities in a timely manner without missing a beat.

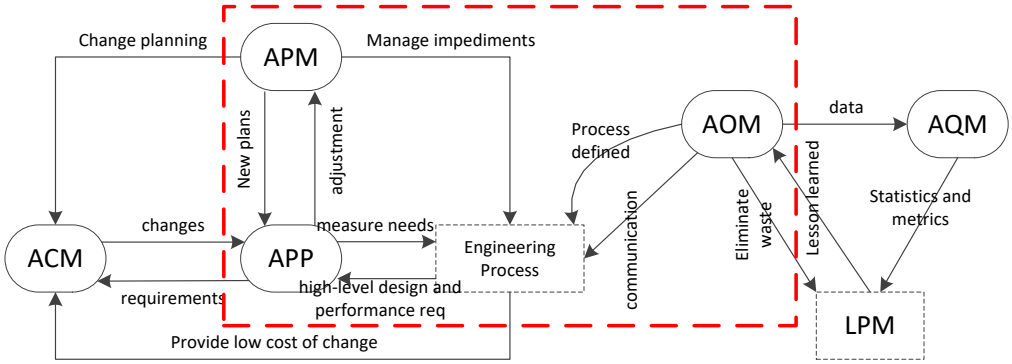

Fig. 1.Proposed agile flow system (consolidated from [2]).

APM	Agile Project Monitoring	**AOM**	Agile Organizational Project Management
APP	Agile Project Planning	AQM	Agile Quantitative Project Management
ACM	Agile Change Management	LPM	Lean Project Management

In Figure 1, the inset illustrates the event flows among APM, APP, Engineering Process (EP), and AOM. The heart of this system lies in the EP where work done takes place, along with exchanges of event flow among the remaining activity nodes during project execution. We will describe these exchanges in the next section.

B. Social Network Scrum

The driver of this research is scrum. The process flow is illustrated in Figure 2. We begin the project by establishing project requirements in the form of product backlog. Each product backlog is further broken down to sprint backlog, which enters the sprint to iterate till the story is done. This might take a few daily sprints to complete the task. Upon completion of this product backlog, a working increment of the software is delivered.

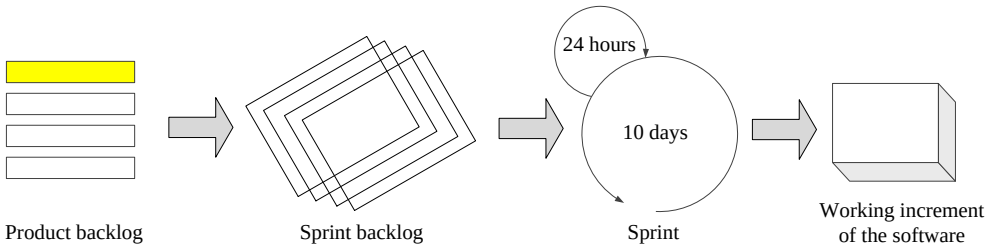

Fig. 2.Scrum process.

We group the product backlog into APP, sprint review and sprint retrospective into APM, daily scrum into Engineering Process, and WiP into AOM, respectively. Starting from new plans in the APP, the product backlog is broken down into stories. They are checked in APM along the sprint process. In some cases, the stories might come back for revision and retry. In the meantime, measure needs are also planned to furnish supplement for performance in high level design by the Engineering Process, where process descriptions are established by AOM. We assume all process assumptions are well-defined that, in an actual project environment, are handled by LPM. Unfortunately, this node is outside the scope of this research. From the above set up, the event flow can be exchanged to and fro among the three nodes that support the Engineering Process of the project on social network rather than conventional in-person scrum. This will prevent all team members from coming in contact and risking virus contraction. One noteworthy activity performed in AOM is control of WiP to keep the Engineering Process manageable. Project data can be archived, shared, disseminated, and retrieved 24x8 on the network. In so doing, proper workload assignment from WiP on the development team will yield the quality of software product to be delivered. We will further describe this aspect in the next section.

C. Performance Measurement

An important outcome of this research is performance measurement of the agile project development. Since we propose a social network scrum as the main information exchange vehicle, gauging the performance on this new vehicle is different from conventional setting. We employ two estimating measures for this research, namely, velocity and WiP. The former measures the team's rate of progress in terms of story points, while the latter measures the work that has started but not yet finished. The rationale behind selecting these two measures is to gauge the amount of work product generated by social network scrum as oppose to in-person scrum. This will reflect the efficiency of social network scrum and the economy of scale as to how much work should be assigned to the team members as they are no longer working side-by-side. The assigned work, aka inventory and subsequently becomes WiP, can be used to determine the costs of delay and effort during the course of project execution. The proposed framework is validated by an experiment to be described in next section.

IV. EXPERIMENT

Due to the novelty of this research and the COVID-19 pandemic situation, several industrial settings declined to participate as they could not afford the loss of productivity and risk of project failure. We decided to resort to our computer science students since they were ideal apprentices that made comparable work force with software engineering professionals [10]. The students formed teams of three members. They were encouraged to rotate task roles among themselves. For example, they switched role from programming to testing, reading code, designing user interface, writing project report, and playing user. At some predetermined breakpoints, they gathered to hold an online scrum meeting for status update and outcome. They could also converse on social network as often as they would like. In so doing, it maintained development agility of team activities.

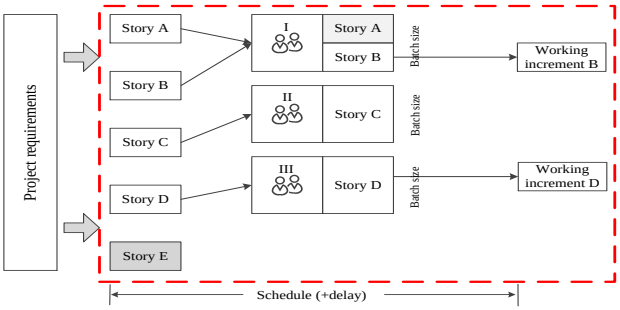

Fig. 3.Event flow of the proposed SNSM.

Figure 3 illustrates the event flow of the proposed SNSM. Project requirements were decomposed into stories, which were considered to be inventory of the scrum process, namely, A, B, C, D, and E. Stories A and B were assigned to developer I, C to II, and D to III, leaving story E unassigned. Each batch size was different depending on the size of the story. Two working increments B and D were accounted for team velocity, while A and C were counted as WiP. The time span from story development to rollout of working increment was production schedule, which also included some delays such as process delay, technical/non-technical delays, personal allowances, etc. The dotted area encompassed scrum meetings that were held online via social network.

Table 1 shows work breakdown of each member assignment. They are inventory (story, Spoint or storypoint, duration), batch size, velocity, WiP, sched+delay, and effort. The units of measure are (-, Spoint, day), Spoint, Spoint, Spoint, hour, and manhr-Spoint or man-hour-storypoint, respectively. For example, story D is estimated to take 4 Spoints, 2 days to complete, having the batch size of 4 Spoints, counted as 2*4 = 8 velocity Spoint, zero WiP Spoint, 2*8 = 16 hours (assume 8 hours per day), and an effort of 16*4 (from Spoint column) = 64 manhr-points to create the working increment. Note that story E requires an estimated [144] manhr-points to create the corresponding working increment. For the time being, it has not yet been assigned to any developer, but it gives us an estimate as to whom and how we would assign the story, i.e., estimation of this task should be slightly higher than the comparable story C.

Table- I: Work breakdown assignment

Inventory			Batch size	Velocity	WiP	Sched+delay	Effort
Story	Spoint	duration					
A	3	2	6	0	3	16	48
B	3	1		3	0	8	24
C	5	3	5	0	5	24	120
D	4	2	4	4	0	16	**64**
E	6	3	6	0	0	24	[144]

Unit: duration:day, Batch size:Spoint, Velocity:Spoint, WiP:Spoint, Sched+delay:hour, Effort:manhr-Spoint.

From the above assignment, the team had to spend time on daily stand-up meeting and scheduled scrum meet. The proposed SNSM mandated that these activities had to perform over social network. All information exchange among team members was carried out via the social network.

Hence, time statistics for performance measurement could be collected automatically.

Figure 4 shows an SNSM session displaying code fragment for discussion by the owner. We used Google Meet as the main tool and Line as supplementary informal conversation tool.

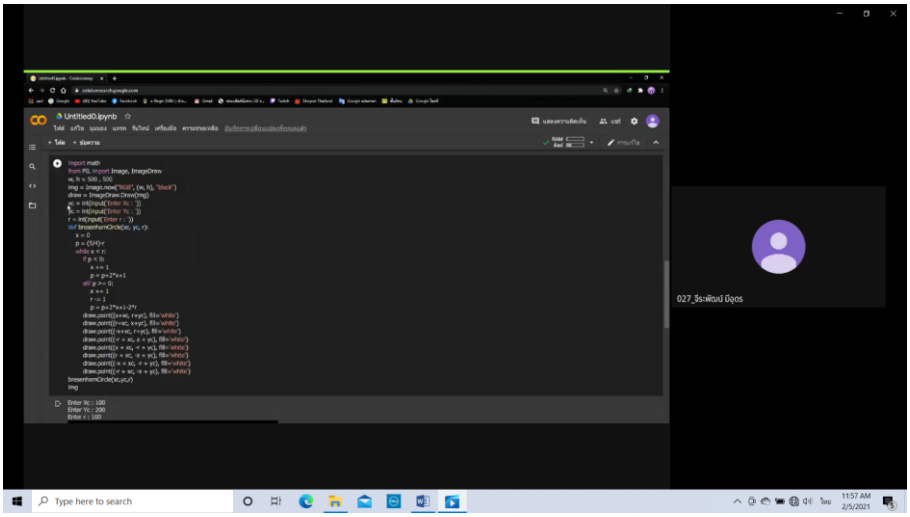

Fig. 4.An SNSM session.

Table 2 shows time statistics being collected by story. Statistics of daily meet and scrum are approximate average over the project meetings. The number of fixes reflects corrections after the meetings. Assuming 5 working days per week, the meet time is obtained from 5*daily meet + scrum. The *attempt ratio* is computed from *meet time* over the *effort* expended in Table 1. The *Fix%* determines how efficient corrections are performed over the effort used. For example, story D took, on average, 8 minutes on daily meet and 40 minutes on scrum, 14 fixes after getting feedback/help from team members, 8*5+40 = 80 minutes of meet time, 80/**64** = 1.25 attempt ratio, and (14*100)/**64** = 21.88 Fix%.

Table- II: Online performance measurement

Story	Daily meet	Scrum	Fix	Meet time	Attempt ratio	Fix%
A	3	40	7	55	1.15	14.58
B	5	40	11	65	2.71	45.83
C	9	40	16	85	0.71	13.33
D	8	40	14	80	1.25	21.88
E	0	0	0	0	0.00	0.00

Unit: Daily meet:minute, Scrum:minute, Meet time:minute.

Note that the statistics of completed stories look slightly better than the rest because they have already been finalized with no more time and effort to be expended.

V. DISCUSSION

To address the focus of this research stated earlier, we set up the inventory plan to advocate the project plan. The notion of WiP, along with velocity, was introduced as the means for monitoring progress. The event flow system was brought in as coverage of project organization. And last, social network was deployed to gauge how efficient and the extent to which it could substitute traditional in-person scrum. Detailed discussions are described below.

A. Benefits of Social Network

The application of social network permitted various statistics that could be gathered directly and accurately. Consequently, results from several attributes were tallied and computed such as time, number of exchanges, for subsequent verification replay and future reference. This turned out to be a big advantage over in-person situation since the statistics usually were not methodically recorded, i.e., daily meet, scrum minutes, and (total) meet time.

Finally, the most beneficial aspect of this SNSM research was undisruptive work under COVID-19 pandemic that has disrupted all business and commerce establishments, social activity, political administration, public health, and daily life of the people world-wide.

B. Production Development Setup

The adoption of production planning scheme contributed greatly to our agile development setup. We started by transforming typical project requirements into inventory. Thus, process planning, monitoring, control, adjusting, and executing were determined based on well-defined production measurements such as batch size, velocity, WiP, schedule, effort, fix, and time statistics. This offered

- standard performance measurement,
- similar monitoring methods and tools in the same manner as other industrial settings, and
- computerized information processing, exchange, archive, and retrieval as opposed to verbal, minutes, and written notes.

The amalgamation of agile method and social network certainly propelled traditional development practice up to the fore-front of digital technology, and kept pace with the technological transformation.

VI. CONCLUSION

This research proposes a social network scrum meeting (SNSM) as an alternate means to substitute in-person scrum meeting in software project development under the COVID-19 pandemic. The proposed framework implemented two setups, namely, production development and online meeting. An experiment was carried out and found to be fruitful. Normal work assignments were not disrupted as a result of social lockdown. Daily meeting and scrum were fine tuned to suit the 'new normal' situation, for example, errors were fixed after online meeting. Thus, development work was business as usual. The results were somewhat satisfactory since every member could 'work from home' and got some outputs done that would have otherwise stagnated. The only shortcoming was the statistics of in-person scrum meeting which was unfortunately prohibited to perform. But the benefits offset this shortfall, that is, production development setup with scrum method and a framework for social network scrum meeting were precipitated. The novelty of the proposed SNSM certainly warrants future work for adaptation to production scale software development.

REFERENCES

1. D. J. Anderson and A. Carmichael, *Essential Kanban Condensed*, LeanKanban University Press, Seattle, Washington, 2016.
2. F. S. F. Soares and S. R. L. Meira, "An Agile Strategy for Implementing CMMI Project Management Practices in Software Organizations," *Proceedings of the 10th Iberian Conference on Information Systems and Technologies (CISTI)*, 17-20 June 2015.
3. K. Schwaber, *Agile Project Management with Scrum*, Microsoft Press, Redmond, Washington, 98052-6399, 2004.
4. M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," *Information and Software Technology*, 54, 2012, 820– 827.
5. K. Molokken-Ostvold and M. Jorgensen, "Group Processes in Software Effort Estimation," *Empirical Software Engineering*, Kluwer Academic Publishers, 9, 2004, 315–334.
6. K. S. Rubin, *Essential Scrum—A Practical Guide to the Most Popular Agile Process*, Pearson Education, Inc., 2013.
7. K. Schwaber and J. Sutherland, *The Scrum Guide*, the Attribution Share-Alike license of Creative Commons, 2020, accessible at https://creativecommons.org/licenses/by-sa/4.0/legalcode.
8. M. Marchesi, K. Mannaro, S. Uras, and M. Locci, "Distributed Scrum in a Research Project Management," *International Conference on Extreme Programming and Agile Processes in Software Engineering*, XP 2007: Agile Processes in Software Engineering and Extreme Programming, 240-244.
9. I. Shamshurin, and J. S. Saltz, "Using a coach to improve team performance when the team uses a Kanban process methodology," *International Journal of Information Systems and Project Management*, Vol. 7, No. 2, 2019, 61-77.
10. I. Salman, A. Misirli, and N. Juristo, "Are students representatives of professionals in software engineering experiments?," *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Florence, Italy, 2015, 666-676.
11. R. Marques, G. Costa, M. Silva, and P. Gonçalves, "A Survey of Failures in the Software Development Process," in *Proceedings of the 25th European Conference on Information Systems (ECIS)*, Guimaraes, Portugal, June 5-10, 2017, 2445-2459, ISBN 978-989-20-7655-3 Research Papers, http://aisel.aisnet.org/ecis2017_rp/155.
12. G. Arcidiacono, "Comparative research about high failure rate of IT projects and opportunities to improve," *PM World Journal*, Vol. VI, Issue II, February 2017, 1-10.
13. J. Verner, J. Sampson, and N. Cerpa, "What factors lead to software project failure?," *Second International Conference on Research Challenges in Information Science*, Marrakech, Morocco, 3-6 June 2008, 71-79.
14. K. Emam and A. G. Koru, "A Replicated Survey of IT Software Project Failures," *IEEE Software*, Vol. 25, Issue: 5, Sept.-Oct. 2008, 84-90.
15. H. Takeuchi and I. Nonaka, *The New New Product Development Game*, Harvard Business Review, January 1986.
16. M. E. Moreira, M. Lester, and S. Holzner, *Agile for Dummies*, Wiley Publishing, Inc., 2010.
17. S. Broman, *Agile Pitfalls*, Thesis, Helsinki Metropolia University of Applied Sciences, 2017.
18. Manifesto for Agile Software Development. https://agilemanifesto.org/
19. T. D. Baruah, "Effectiveness of Social Media as a tool of communication and its potential for technology enabled connections: A micro-level study," *International Journal of Scientific and Research Publications*, Volume 2, Issue 5, May 2012, 1-10.
20. R. P. Santos, , M. G. P. Esteves, G. S. Freitas, and J. M. Souza, "Using Social Networks to Support Software Ecosystems Comprehension and Evolution," *Social Networking*, 3, 108-118, Published Online February 2014 (http://www.scirp.org/journal/sn).
21. M. Yang, Y. Wang, and X. Hou, "Research on Accurate Information Pushing Based on Human Network," *Social Networking*, 6, 2017, 181-196, http://www.scirp.org/journal/sn.
22. J. L. Santos and R. R. Sampaio, "Temporal Analysis of the Diffusion of Knowledge in Networks of Software Maintenance and Development Project Team," *Social Networking*, 8, 2019, 122-146, http://www.scirp.org/journal/sn.

AUTHORS PROFILE

Nalinee Sophatsathit receives her bachelor degrees in Communication Arts from Sukhothai Thammathirat University and Computer Science from Phetchaburi Teacher's College, Masters in Computer Science from College of Engineering, Chulalongkorn University, and Ph.D. in Innovation Management from Innovation College, Suan Sunandha Rajabhat University. She is a faculty member in Computer Science Program, Faculty of Science and Technology, Suan Sunandha Rajabhat University. Her principal publications are in software engineering. She has written a book on UML Standards for Object-Oriented Software Engineering (in Thai). Her research interests are innovation in IT and software usability. She can be reached at nalinee.so@ssru.ac.th.