JIST Journal of Information Science and Technology
Volume 9, NO 1 | JAN – JUN 2019 | 63-70

Experience in Applying of ISO 29110 to Agile Software Development

วราพร จิระพันธุ์ทอง

Waraporn Jirapanthong

วิทยาลัยครีเอทีฟดีไซน์แอนด์เอ็นเตอร์เทนเมนต์เทคโนโลยี มหาวิทยาลัยธุรกิจบัณฑิตย์

College of Creative Design and Entertainment Technology, Dhurakij Pundit University

Received: May 7, 2019; **Revised**: June 14, 2019; **Accepted**: June 18, 2019; **Published**: June 25, 2019

ABSTRACT – We advocate the fact that the standard for software development should be established in an organisation, since the dynamic of software development projects leads more error-prone and time consuming. The standard is expected to reduce the difficulties of development activities, in particular, analysis and design. This research is aimed to apply the standard ISO/IEC 29110 and agile method. The standard is studied and the supporting tools are analyzed. The good practice of applying the standard ISO/IEC 29110 with agile practice is purposed. A sample group of software development teams is created and coached according to our approach. Then, the sample group works on their own software development projects based on the purposed good practice. The questionnaire is given to the group in order to collect the feedback and comments on the approach. Based on the results of questionnaire, it shows that software development teams are satisfied with the approach, especially, in the term of requirements tracking, responding time to customers, and change management with 4.73, 4.52, and 4.33 out of 5, respectively.

KEYWORDS: ISO 29110, Agile, Scrum, Process Improvement, Software Quality Assurance

Corresponding Author : waraporn.jir@dpu.ac.th

1. Introduction

According to the software development, the main activities are analysis, design, and implementation of similar and different aspects of the systems:
1) Analysis – this activity is aimed to explore and justify the requirements of software systems which represent the aspects of the systems.
2) Design – this activity is aimed to elaborate the requirements from the analysis process and to design the software systems.
3) Implementation – this activity is aimed to implement the requirements and design artefacts produced from previous processes as components and to assemble a software system.

However, according to the majority of approaches being used in organizations, there are some issues found: 1) the software approaches proposed recently are not flexible, practical, and appropriate enough to the conventional approaches being used in the organizations, 2) different organizations have various behavioral cultures and traditions depending on the strategies and missions of the organizations. In particular, many organizations found difficulties to adopt the software development standards to fit into their strategies and missions. Consequently, there are still errors and mistakes during the development of software systems that requires the reuse of software components.

We encourage the fact that the standard for software development process should be established in an organization since it leads less error-prone and less time consuming.

Our contribution is to develop a set of practical activities including applying available software tools in order to enable ISO/IEC 29110 into software projects development in an organization. This study aims to purpose a good practice to apply the software development standard, particularly ISO/IEC 29110, with agile methods to support very small companies to develop their software projects. In the following section, the background of ISO/IEC 29110 and agile software development are summarized. Our approach is presented in Section 3, the experimental results is described in Section 4, and the conclusion and discussion is given in Section 5.

2. Background

2.1 ISO/IEC 29110

ISO/IEC 29110 [1-5] is developed with the objective to assist and encourage Small and Medium-size Enterprise in the assessment and improvement of their software processes. It proposes a set of profiles increasingly helping SMEs with more complete and extensive processes. Currently, there are four levels of profiles i.e. Entry Profile, Basic Profile, Intermediate Profile, and Advanced Profile [1]. The Basic Profile (Basic VSEs Profile) [4] is applied for improving working process, particularly for project management process. It is suitable for very small entities which each project team encompasses no more than 25 persons.

The profile defines two processes i.e. Project Management (PM) and Software Implementation (SI). The activities of each process are identified. The tasks and work products of each activity are also identified. In particular, PM process consists of four activities: (a) Project Planning (PM1), (b) Project Plan Execution (PM2), (c) Project Assessment and Control (PM3), and (d) Project Closure (PM4). SI process consists of six activities: (a) Software Implementation Initiation (SI1), Software Requirements Analysis (SI2), Software Architectural and Detailed Design (SI3), Software Construction (SI4), Software Integration and Tests (SI5), and Product Delivery (SI6). The activities include 50 tasks and produce 22 work products. The tasks are performed and their expected input and output artefacts (work products) are produced.

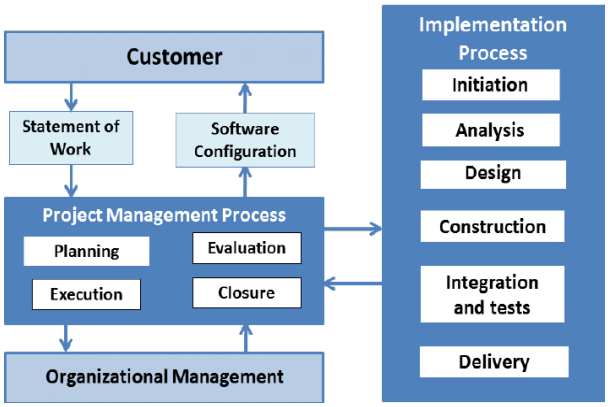

Figure 1.ISO/IEC 29110 Basic Profile processes and activities (from www.iso.org/standard/51154.html).

The two processes produce 22 work products as shown in Figures 2 and 3. However, the coverage of the software implementation activities for ISO/IEC 29110 still varies. The software process standard provides a set of best practices and guidelines for improving the quality of the software process resulting from the process. Otherwise, the standard has not prescribed a particular software development methodology.

Moreover, software development teams have some difficulties on selecting software tools and development methods which are compliant with the standard. Some studies [5][6][7] present the effort of pursuing the software process standards and models. However, those focus on their own local companies which have different organizational cultures and

business process to Thai ones. Although some work [8] studied on software developing process of Thai organizations, they do not focus on the standard of ISO/IEC 29110.

Recently, a number of software development projects held by small teams in Thailand are related to diverse business domain and driven by gamification technology [9][10]. However, those are not applying on the standard for software process.

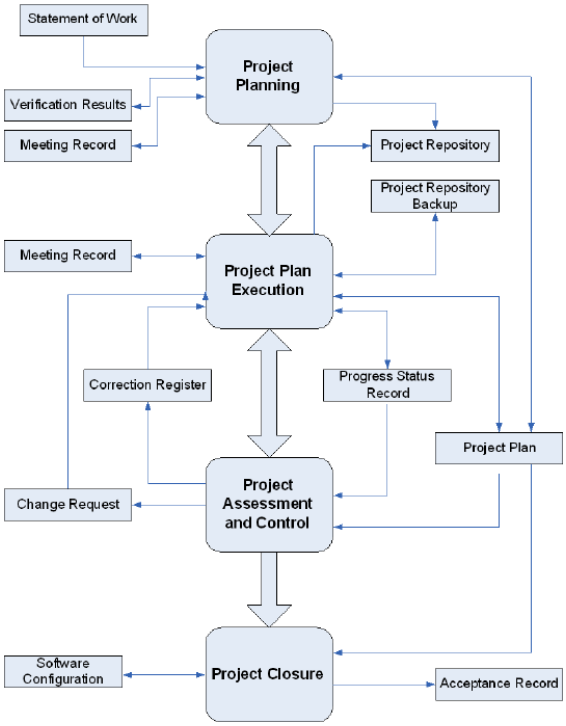

Figure 2. Project Management process diagram [4].

2.2 Agile Software Development

Agile software development is a way of organizing the development process, emphasizing direct and frequent communication. It is preferably face-to-face, frequent deliveries of working software increments, short iterations, active customer engagement throughout the whole development life-cycle and change responsiveness rather than change avoidance. The main components used in the agile development are: (a) each activity in software development process i.e. analysis is managed in a short period (2-4 weeks), (b) product backlog which is a list of prioritized requirements for the product, (c) sprint or iteration backlog which is a selection of items from the product backlog being developed in an iteration, (d) sprint review which is an evaluation of the outcome of a sprint, (e) sprint planning which is done in the start of an iteration or a sprint and results in a sprint backlog, and (f) standup-meeting which is a daily short meeting.

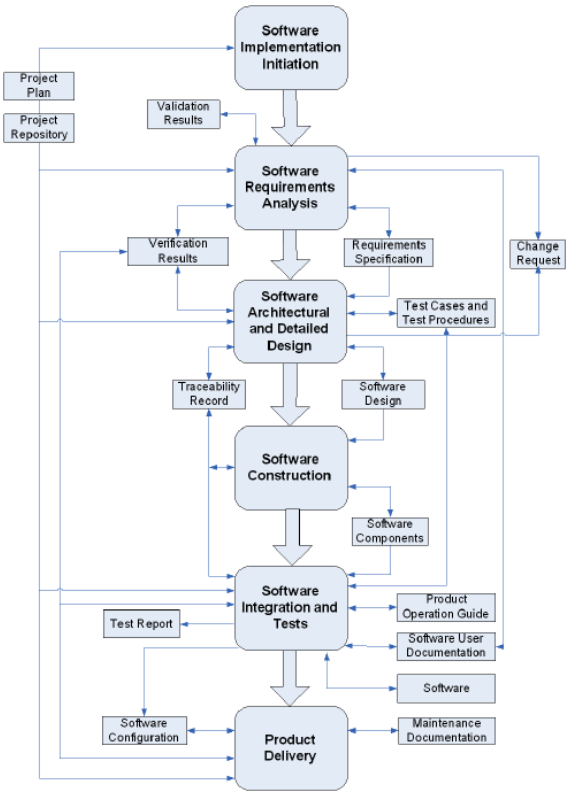

Figure 3. Software Implementation process diagram [4].

The ideas of agile software development have been inspired by agile and learn manufacturing. They have been in use in many types of industries for decades. Some important changes need to be made to make this fit software development. The main principle from lean development being applied is the principle of waste reduction. All work and products which do not directly contribute to the development of software should be considered as waste and thus avoided. Currently, there are common agile methods. For examples, Extreme Programming [11 - 13] has grown surprisingly fast. The basic principles are easy to grasp and seem to address the fundamental problems of developers.

Moreover, Scrum [14] is a framework of developing, delivering, and sustaining complex products. The essence of scrum is a small team of people. The individual team is highly flexible and adaptive. These strengths continue operating in single, several, many, and networks of teams that develop, release, operate and sustain the work and work products of thousands of people. They collaborate and interoperate through sophisticated development architectures and target environments. The scrum team consists of a product owner, development team, and scrum master. Scrum teams are self-organizing and cross-functional. Self-organizing teams choose how best to accomplish

their work, rather than being directed by others outside the team. Cross-functional teams have all competencies needed to accomplish the work without depending on others not part of the team. Scrum teams deliver products iteratively and incrementally, maximizing opportunities for feedback.

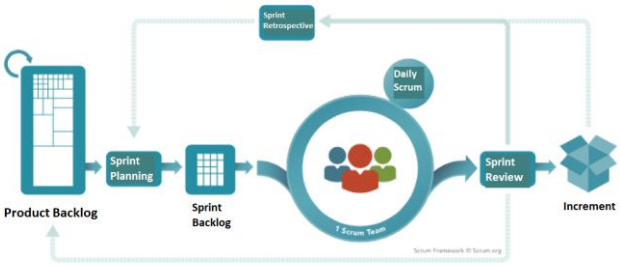

Figure 4. Scrum Framework (from scrum.org).

3. Research Methodology

3.1 Research Framework

Our study consists of the following details: 1) Population is a group of Small and Medium-size Enterprises (SMEs) involved in software development in Thailand, 2) The sample group is a group of 8 Thai SMEs who involve software development and are based in Bangkok and suburbs. Those consists of having passed and never passed the ISO/IEC 29110 certification.

Research tools include, firstly, tools to create good practices in the software development process according to the ISO/IEC 29110 framework. They consist of (a) ISO/IEC 29110 Part 5-1: Management & Engineering Guide for Basic Profile. The guide presents an implementation and various activities including tasks in each activity that is in accordance with ISO/IEC 29110, and (b) an interview form to collect personal and organizational profiles of the sample group regarding practices with ISO/IEC 29110.

Secondly, software tools are used to enable Agile method in the software development process according to the ISO/IEC 29110 framework. Those are Microsoft Project, HacknPlan, and Lucidchart programs for supporting the activities in the software development process. Thirdly, questionnaire is used to collect the feedback and satisfaction of the sample group towards applying the approach.

3.2 Research Method

Our research method consists of: 1) organizational executive staffs and project members of the sample group are interviewed in order to elicit the requirements and attitudes towards applying the international standard of software development process, 2) the sample group is trained for a 3-hour course how to apply with the ISO/IEC 29110 and agile practice as our approach, 3) each team in the sample group has been given the statement of work to develop a software, 4) the sample group starts to develop the software following the proposed approach, 5) the questionnaire are given to the sample group in order to collect the feedback and comments on the approach, and 6) information about agile software development processes according to the ISO/IEC 29110 framework is analyzed and synthesized.

3.3 Our Approach

According to ISO/IEC 29110, the two processes produce 22 work products. Our focus relies on the activities related to each work product which is described as follows:

1) Statement of work is an initial input for the whole process.
2) Project repository is initially set up from Project Planning activity and gradually added from Project Plan Execution. It is then treated as input artefact for activities in software implementation process i.e. *Software Implementation Initiation, Software Requirements Analysis, Software Architecture and Detailed Design, Software Integration and Tests, and Product Delivery* activities.
3) Project plan is created during *Project Planning* activity and updated during *Project Plan Execution* activity. It is used for *Software Implementation Initiation, Project Assessment and Control, and Project Closer* activities.
4) Project Repository Backup is synchronously updated to project repository during *Project Plan Execution* activity.
5) Progress status records are created and inquired during *Project Plan Execution* activity and used for *Project Assessment and Control* activity.
6) Verification results are produced and inquired during the activities i.e. *Project Planning, Software Requirements Analysis, Software Architecture and Detailed Design, Software Integration and Tests, and Product Delivery.*
7) Meeting records are produced and inquired during *Project Planning and Project Planning Execution* activities.
8) Correction register is created during *Project Assessment and Control* activity and applied for *Project Plan Execution* activity.
9) Change requests occur during *Project Assessment and Control and Software Requirements Analysis* activities and are

inquired during *Project Plan Execution* activity.

10) Validation results are produced and inquired during *Software Requirements Analysis* activity.
11) Requirements specification is created and inquired during the activity *software requirements analysis*.
12) Software user documentation is created and used for *Software Requirements Analysis* and *Software Integration and Tests* activities.
13) Acceptance record is created during *Product Closure* activity.
14) Software configuration are created and inquired during *Product Delivery and Project Closure* activities and used for *Software Integration and Tests* activity.
15) Software design is produced and inquired during *Software Architecture and Detailed Design* activity and applied for *Software Construction* activity.
16) Traceability records are created during *Software Architecture and Detailed Design*, *Software Construction*, *and Software Integration and Tests* activities.
17) Test cases and test procedures are created and inquired during *Software Architecture and Detailed Design and Software Integration and Tests* activities.
18) Software Components are created and updated during *Software Construction* activity and used for *Software Integration and Tests* activity.
19) Software is produced and updated during *Software Integration and Tests* activity.
20) Test reports are generated during *Software Integration and Tests* activity.
21) Product operation guide is created and updated during *Software Integration and Tests* activity.
22) Maintenance documentation is created and updated during *Product Delivery* activity.

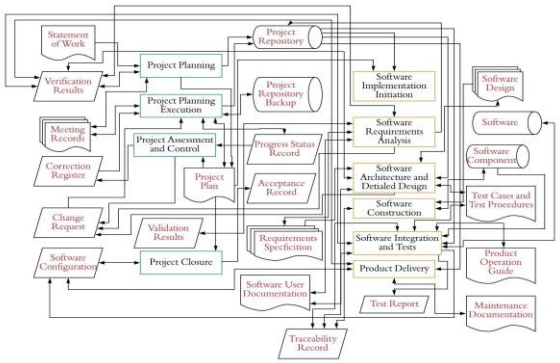

Figure 5. The 22 work products with related activities.

According to software engineering process, there are a variety of models and methods used to plan the process. Software engineering management includes providing comfortable work environment for developers development planning and quality control. There are many software tools that are specialized to varying degrees in project management, as well as numerous general business software solutions for planning, tracking and monitoring software projects. These systems include Atlassian JIRA [15], Bugzilla [16], Redmine [17], GNATS [18], YouTrack [19], Trac [20], and others.

To analyze both early and late start and finish dates for the implementation of the project, the methods of network planning are used [21]. These methods integrate all the stages of the project, thus it is possible to determine the total duration [22].

Large projects are characterized by the complexity of constructing a plan or scheduling the project in distinct work stages. One of the methods that can be used in this case is the GERT method, based on GERT networks. With this method it is possible to determine the expected duration of the project on the basis of three probabilistic time estimates.

As a technique for schedule planning and control PERT is focused on analyzing the tasks necessary for the project implementation. The analysis of the execution time of each individual task, as well as the construction of the network schedule as called PERT network diagram. We can calculate the early and late start dates for the tasks. The flexibility in the schedule can be seen as the slack. It is an amount of time that a task can be delayed without causing a delay to subsequent tasks or project completion date. The sequence of project tasks which ass up to the longest overall duration is called a critical path as shown in Figure 6. It determines the shortest time possible to complete the project. Tasks on the critical path should be completed on time to make sure that the project finishes on time.

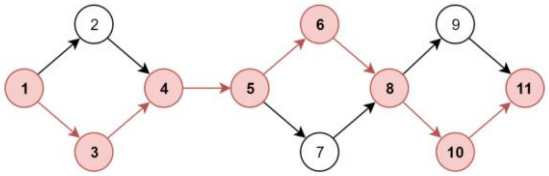

Figure 6. PERT and Critical path.

Additionally, Gantt charts are used to illustrate a project schedule [23 - 24]. It is suitable for planning, especially small-scale projects.

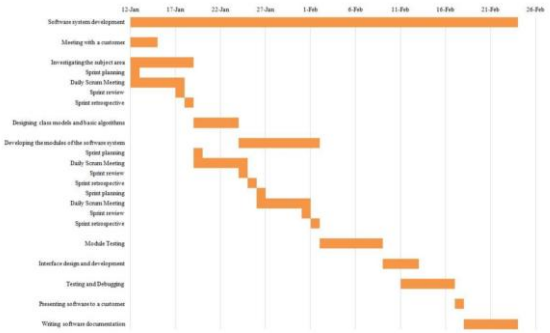

Figure 7. Gantt chart.

There are software development methodologies guided by values and principles based on Agile. Scrum is one of the flexible approaches which have become popular. The rules established for software project management allow developers to use the existing practices of coding, correcting requirements or making tactical changes. The methodology provides the ability to detect and eliminate deviations from the desirable result at earlier stages of the software product development. Iteration in Scrum when functional growth of software is created is sprint which is rigidly fixed in time (see figure4). A sprint length of no longer than 2 weeks is appropriate as such a short sprint makes the process of software development more flexible, releases come earlier and more frequent, consumer responses arrive faster, and less time is wasted.

As shown in Table 1, some software tools [15][16][17][18][19][20][25][26][27][28][29] can be applied in order to support ISO/IEC 29110 work products. Some work products (*) are created based on each team's implementation methods. The supporting tools depend on the methods and their techniques. Some work products (**) are documents which can be created by predefined templates. Also. Some work products (***) can be established based on predefined procedure and documented based on templates.

Table 1. Software tools supporting each work product.

Work Products	Software Tools
Statement of work	**
Project repository	Gitlab
Project plan	JIRA, ProWorkflow, Hive, Redmine, MS Project
Project Repository Backup	JIRA, Bugzilla, Redmine, GNATS, YouTrack, Trac
Progress status records	ProWorkflow
Verification results	***
Meeting records	**
Correction register	JIRA, Redmine, GNATS,
Change requests	**
Validation results	***
Requirements specification	*
Software user documentation	**
Acceptance record	**
Software configuration	**
Software design	*
Traceability records	**
Test cases and test procedures	Bugzilla, GNATS
Software Components	JIRA, Eclipse
Software	Eclipse
Test reports	Bugzilla
Product operation guide	**
Maintenance documentation	**

* depends on team's implementation methods
** template
*** template and predefined procedure

We propose the good practice for applying with the international standard for software development process to Agile software development principle. In particular, we focus on the applying with ISO/IEC 29110 to Scrum method. As shown in Figure 5, our proposed practice is illustrated. The software tools are applied for creating work products or supporting the activities: a) Project management tool for making Gantt chart, b) Project management tool for setting up dashboard, c) Project management tool for assigning tasks, d) Project management tools for tracking systems e.g. Redmine, e) Software tools for managing software projects e.g. JIRA, f) Communication tools for recording correction register and changes and including instant messaging systems e.g. Slack, g) IDE tools for supporting software construction environment, h) Repository tools for documentation and software components e.g. GitHub, i) Version control tools for software construction e.g. Git, j) Testing tools for supporting testing environment, and k) Mechanism for validation activity.

4. Experimental Results

According to the ISO/IEC 29110 standard framework, there are 2 processes, 10 activities, and 22 work products. Each activity is divided into sub-tasks consisting of task name, input work product, output work product, role for each sub-task, and necessary details as follows: (1) *Project Management*

Process which consists of 4 activities, namely PM1, PM2, PM3, and PM4 (2) *Software Implementation Process* consists of 6 activities, namely SI1, SI2, SI3, SI4, SI5, and SI6.

We have each team applied a checklist for controlling all work products. The checklist consists of activities and work products as proposed good practice according to ISO / IEC 29110 standard. There are various details such as process, activity, task, input and output work product and related important information. The criticality is evaluated on a 1 to 5 scale. Based on the results of questionnaire, it shows that software development teams are satisfied with applying on the standard ISO/IEC 29110 with agile Scrum method under the purposed set of activities and software tools. Particularly, they are satisfied in term of requirements tracking, responding time to customers, and change management with 4.73, 4,52, and 4.33 out of 5, respectively. Moreover, the comments and suggestions based on the questionnaire can be summarized as follows: a) the good practice with the approach can be used for other projects in the entire company, b) the project has a clear procedure and templates that assist to reduce conflict issues, c) team members are able to understand their roles and responsibilities easily and clearly, d) project managers can evaluate and analyze the time frame of the project more correctly and precisely and project risks are easily reduced, e) it easily increases project management efficiency, particularly reducing the issues during implementation, f) it better explores business opportunities for the company, and easily follows up with customers, g) it can effectively reduce project costs, h) establishment of ISO/IEC 291110 is quite complicated, although it is found very useful after applying, and i) it lacks of the manual or supporting tools.

5. Discussion and Conclusion

The study applied the standard ISO/IEC 29110 and agile method. The good practice is purposed in order to enable the standard in the scrum practice. All work products according to the standard are studied and the supporting tools are evaluated. We have corporated with software development teams from SMEs which are based in Bangkok and suburbs. They developed their software development projects based on our approach. We found some problems: a) the difficulty to get support from organisations. Due to timing constraints, an organisation usually considers output rather than establishing standard, b) the uncontrolled growth of changes. Ideally, the establishment of the standard needs to have a stable and clear vision of domain to have a solid time frame; however, it needs to be flexible enough to evolve new requirements, c) the difficulty in communication, our approach is based on a collaborative process where people from various disciplines need to communicate each other. In other words, communication is required to facilitate and improve the software system development. However, it is not easy to support communication between various groups of stakeholders in an organisation. Successful communication between stakeholders depends on various factors such as: (i) sufficient resources e.g. staff or tool to facilitate the communication; (ii) differences in organisational cultures; (iii) distinct organisational structures; and (iv) stakeholders' attitudes and aspirations. Unsuccessful communication in an organization leads to misunderstanding and lacks of some concepts during the development of software systems. It is also found that the process can be successfully achieved if the software development team consists of experienced members i.e. system analysts, project managers, and programmer.

Additionally, a number of possible directions for further investigations have been identified. For example, tool for document generation and visualization. A large number of various artefacts can be generated. It is therefore believed that the approach could benefit by providing tool fully support for the specification of documents. In addition, sophisticated techniques for visualization could support the use of documents more efficiently.

References

[1] ISO/IEC TR 29110-1, Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 1: Overview, ISO/IEC 29110, 2016.

[2] ISO/IEC 29110-2-1, Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 2-1: Framework and taxonomy, ISO/IEC 29110, 2015.

[3] ISO/IEC TR 29110-3-1, Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 3: Assessment Guide, ISO/IEC 29110, 2015.

[4] ISO/IEC 29110-4-1, Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 4-1: Profile specifications: Generic profile group, ISO/IEC 29110, 2011.

[5] F. Plante, and C. Y. Laportee, "Implementation of an Agile-ISO 29110 Software Process in a Large Canadian Financial Institution," 2018. [Online]. Available: ResearchGate, https://www.researchgate.net/publication/326271657_Implementation_of_an_Agile-ISO_29110_Software_Process_in_a_Large_Canadian_Financial_Institution. [Accessed: June 5, 2019].

[6] M. Takeuchi, N. Kohtake, S. Shirasaka, Y. Koishi, K. Shioya, “Report on an assessment experience based on ISO/IEC 29110,” Journal of Software: Evolution and Process, 26(3), pp. 306-312. 2014.
[7] D. A. Marca and B. Perdue, “Software engineering approach and tool set for developing Internet applications,” in Proceedings of the 22nd international conference on Software engineering, 2000. pp. 738-741.
[8] W. Jirapanthong, "Comparative Study Between Software Product Line and Waterfall Process," Journal of Information Science and Technology, Vol.1, No.1 , pp.1-8, Jan 2010.
[9] **บัญญพนต์ พูลสวัสดิ์ และ พนมพร ดอกประโคน, "เกมบนโปรแกรมเชิงจินตภาพและแนวคิดเชิงคำนวณอย่างเป็นระบบ,"** Journal of Information Science and Technology, Vol.6, No.2, pp.9-16, July 2016 .
[10] I. Suttichurchart, P. Metapatara, P. Chuangyanyong, and P. Chiravirakul, " Esc Challenge Game application for enhancing executive functions ," Journal of Information Science and Technology, Vol.8, No.1 , pp.35-46, Jan 2018.
[11] “Manifesto for Agile Software Development”. 2001.[Online]. Available: https://agilemanifesto.org/. [Accessed: March 26, 2019].
[12] K. Beck, Extreme Programming Explained: Embrace Change, 2nd edition. Addison-Wesley, 2004. ISBN 978-0-321-27865-4.
[13] J. Carroll and D. Morris. Agile Project Management in easy steps, 2nd edition. In Easy Steps. p. 162. ISBN 978-1-84078-703-0.
[14] “The home of Scrum”. [Online]. Available: https://www.scrum.org/. [Accessed: April 23, 2019].
[15] “Atlassian JIRA”. [Online] Available: https://www.atlassian.com/software/jira. [Accessed: March 17, 2019].
[16] “Bugzilla”. [Online]. Available: https://www.bugzilla.org/. [Accessed: March 17, 2019].
[17] “Redmine”. [Online]. Available: https://www.redmine.org/. [Accessed: March 19, 2019].
[18] “GNATS”. [Online]. Available: http://manpages.ubuntu.com/manpages/precise/man7/gnats.7.html. [Accessed: April 4, 2019].
[19] “YouTrack”. [Online]. Available: https://www.jetbrains.com/youtrack/. [Accessed: April 4, 2019].
[20] “Trac”. [Online]. Available: https://www.jetbrains.com/youtrack/. [Accessed: April 5, 2019].
[21] A. Chernigovskiy, R. Tsarev, and D. Kapulin, “Scheduling algorithms for automatic control systems for technological processes,” Journal of Physics: Conference Series 803 art. no. 012028, 2017.
[22] A. Chernigovskiy, D. Kapulin, E. Noskova, T. Yamskikh, and R. Tsarev, “Production scheduling with ant colony optimization,” IOP Conference Series: Earth and Environmental Science 87 art. no. 062002, 2017.
[23] W. Clark, The Gantt Chart: A Working Tool of Management. New York, NY: Ronald Press, 1922.
[24] T. Flouris and D. Lock, Managing Aviation Projects from Concept to Completion. Ashgate Publishing Limited, 2012. ISBN 978-1-4094-8613-8.
[25] “Gitlab”. [Online]. Available: https://gitlab.com/gitlab-org. [Accessed: April 11, 2019].
[26] “ProWorkflow”. [Online]. Available: https://www.proworkflow.com/. [Accessed: April 11, 2019].
[27] “Hive”. [Online]. Available: https://hive.com/. [Accessed: April 11, 2019].
[28] “MS Project, Release notes for Monthly Channel releases in 2019". [Online]. Available: https://docs.microsoft.com/en-us/officeupdates/monthly-channel-2019. [Accessed: April 19, 2019].
[29] “Eclipse”. [Online]. Available: https://www.eclipse.org/. [Accessed: April 11, 2019].