Development of Software Process Guideline to Improve Software Process Using ISO/IEC 29110-5-1-2 Software Implementation Process

Sahanah A/P Pandu Rengan, Hazlifah Mohd Rusli*

Universiti Teknologi Malaysia
sahanahpr@gmail.com, hazlifah@utm.my

Article history

Received:
1 December 2021

Received in revised form:
6 December 2021

Accepted:
10 December 2021

Published online:
22 December 2021

*Corresponding author
hazlifah@utm.my

Abstract

Many software development organizations all over the world are small and medium enterprises. Successful implementation of Software Process Improvement (SPI) methodologies in very small and software enterprises (VSEs) is generally not possible because such organizations cannot invest the cost of implementing these programs. There are various SPI methodologies to address these issues, which have also been deployed in organizations. In this study, recent and significant ISO/IEC 29110 SPI methodologies and its Software Implementation (SI) process are compared with the current software team's current practices. The comparison result serves as the basis for developing a process guideline that best suits the software team. The process guideline will facilitate the maturity of software process and standardization within the software team, contributing to software product quality produced by the software team in the future.

Keywords: *Software Process Improvement, Software Implementation Processes, ISO/IEC 29110, Very Small Software Enterprises, SPI Methodology*

1. Introduction

There are many reasons that cause software projects to be delivered late. One of the most crucial reasons is the poor software implementation process executed by the software team members. Even though software process improvement (SPI) activities cannot ensure project completion on time, poor software implementation activities often delay project delivery. This paper will discuss the study conducted on a small software development team on improving the team's existing software development process by referring to software improvement (SI) activities in ISO 29100-5-2-1 standards. The ISO 29100 standard was chosen as it is a compilation of guidelines designed to meet the requirements of Very Smaller Entities (VSE). According to Mittner & Buchalcevova, "a small company can be described as less than 50 software engineers and a small project as less than 20 software engineers working on"[1]. Thus, the software development team in this study can be considered a VSE as it has less than 25 employees working in the team.

The study aimed to develop software process lifecycle guidelines that can improve the software development process. The software development team is part of a multi-media business group of a semiconductor manufacturer located in the Klang Valley. The software team is in charge of conducting research and developing new liquid crystal display (LCD) TV systems for the LCD integrated circuit (IC)

* *Corresponding author. hazlifah@utm.my*

controller. Some of the currently implemented software projects are Android and Smart TV system on a chip (SoC) software solutions for TV brand manufacturers.

Most of the software was designed to accommodate the latest TV technology, and when new requirement arrives, the software gradually evolves to match the changes. The software team has encountered significant challenges in recent years, with many projects experiencing schedule overrun and delay in delivering software products. One of the main contributors to this problem is poor software implementation activities within the software development team. The team had problems adhering to the current development process. The developed documentation had insufficient information, and the document control versioning was absent. Thus, a study was conducted to understand the gap between what is practiced by the current software team and the procedures proposed by ISO 29100-5-2-1 standards. Based on the gap analysis, a process guideline for the software development lifecycle to be used by the team was developed. A pilot project was constructed to test the developed process guideline, and finally, the efficiency of the developed process guideline was analysed. The scope of the study focuses on the ISO 29100-5-2-1 standards, and the pilot project was conducted among the internal software team members.

The study is significant as it allows the software team to know the inefficiencies in the existing software development process, thus allowing them to find a solution that best suits them. The developed guideline also serves as a reference for future upcoming projects. The following sections will discuss the relevant literature related to the study, the methodology used, the findings, and the conclusion.

2. Literature Review

Software process is the collection of operations, processes, and transformations that developers will use to build and manage software and the related products, such as product plans, designs documents, code, test cases, and user manuals. Process improvement is the alteration and adjustment of a process to achieve a better result from the initial and offers an improved Return on Investment (ROI) [2]. The principle of process improvement for any process is apparent. Process improvement involves assessing the quality of a process and identifying techniques to improve them. In terms of software, it means constant monitoring of the mistakes made during the development process and avoiding them in the future [3]. To avoid making those mistakes again, a software developer may need to modify the software development process. Software process improvement provides best practices and procedures for enhancing the existing approaches in companies to deliver higher-quality software products.

SPI has driven many businesses, corporations, and organizations to improve software development processes to produce high-quality products over the last two decades [4]. An efficient software process focuses on an organization's entire structure, integrating the employees, resources, processes, and technologies involved. As a result, it improves the software organization's performance and product quality and reduces risk and cost [5]. If an organization has a well-defined process, the management can make better decisions on equipment purchases, work engagement, project management, and project target achievement [6].

Several models have been proposed to support software process improvement, and they evaluate the software product, quality, project, and weakness. Though the overall goal of the models is to control and optimize an organization's software process, various tools and techniques were used. Some examples of the SPI model are Capability Maturity Model Integration (CMMI), Personal Software Process (PSP), Team Software Process (TSP), and International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) 29110.

2.1. Capability Maturity Model Integration (CMMI)

CMMI focused on the managerial viewpoints of the software development process. A software development model is a process for software developers to perform coding, testing, deploying, and developing their software. A software management model is a process for software development projects to plan, design, and distinguish what aspects are required to manage the project. It is a model practiced to attain insight into and control the development progress to predict and improve project activities towards achievement [7]. The model includes guidance and suggestions to assist business entities in addressing issues and improving output. CMMI is practiced by more than 5000 companies around 70 countries across the world to assist in determining and achieving business objectives [8].

2.2. Personal Software Process (PSP)

Personal Software Process (PSP) model was developed with the aim of adapting the CMM's underlying principles to the software development process of an individual programmer or developer [9]. It is different from the rest of the standard procedures; PSP is a structured software development approach. It is developed to assist the software engineers and developers in improving their performance by keeping track of their expected and actual code development process. It provides the software engineers with structured processes for developing their software. In PSP, there are seven process levels. Each level includes comprehensive scripts, checklists, and templates that help the engineer go through the essential measures and improve his software process.

2.3. Team Software Process (TSP)

Team Software Process (TSP) is an extension of PSP approaches, developed for a team of developers or engineers in an enterprise who are PSP certified or at least have received formal training on PSP and have fully implemented the model [10]. TSP aims to develop independent teams that can organize and monitor their work, establish their project targets, and own their processes. The team may improve their software process and deliver quality goods with planned costs and efficient schedules. Although the PSP is not as commonly practiced as other SPI models, a few studies have shown that it improves the software engineering process.

2.4. International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) 29110

ISO 29110 Lifecycle Profiles for Very Small Entities is a compilation of guidelines designed to meet the requirements of Very Smaller Entities (VSE) [11]. The ultimate goal of the ISO 29110 framework is to help small software companies evaluate and improve their software processes. The generic profile group contains four profiles known as Entry, Basic, Intermediate, and Advanced. Entry Profile is suitable for VSEs working on small projects, for example, at most six person-months effort) and for new VSEs. Basic profiles are ideal for software development practices of a single application by a single project group of a VSE. Intermediate profiles target VSEs developing many projects with more than one project team. Advanced Profile is targeted at those VSEs want to sustain and grow as a competitive software development business.

2.6. Software Process Improvement for Very Small Entities

The International Organization for Standardization (ISO) or ISO/IEC JTC1/SC7 Working Group 24 defined "very small entities" as "an organization with up to 25 participants" [12]. VSE is the common term used in SPI initiatives to describe small organizations. It is difficult to define the concept 'Small' or 'Very Small' entities because there is no widely accepted definition and comprehensive study.

3. Methodology

The study consisted of four main phases, which were the initiation phase, development phase, implementation phase, and analysis phase. Figure 1 shows the phases and activities associated with each phase.

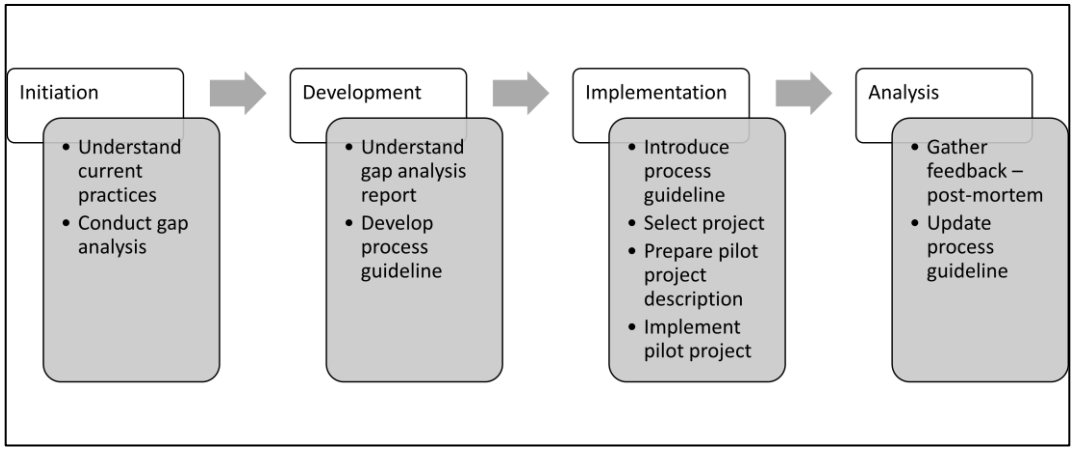

Figure 1. Phases and Activities

3.1. Initiation Phase

The initiation phase consisted of two main activities: understanding the existing software implementation practices and available documentation in the software team and conducting a gap analysis on the current software implementation approaches practised by the team.

For the first activity, three primary resources were used:

- Existing Software Operation Process (SOP) documents kept in local shared folders and some that were accessed via the company's intranet. The SOP documents included the practices for analysing new or updated requirements, code repository preparation, code review formats, code commit procedures, etc.
- Interview sessions with software team members and managers. Two types of interviews were conducted, which were structured and semi-structured. The structured interviews comprised closed-ended questions and were conducted online with senior engineers and managers within the software team to verify the current software implementation processes. Experts holding senior positions and over three years of working experience were selected for the structured interviews. A semi-structured interview was conducted online with other team members with less than three years of working experience. They were queried about their detailed practices for each development project. At least fifteen team members, including the managers, were involved in the interview sessions to help facilitate research study and directions.
- Observation of software team development practices. The observation focused on gaining diverse insights about workplace loads, team occupancy rate, the time required for each process, tight situations, and communication paths among the team members.

In the second activity, the selected activities in the SPI model were compared to the current software development life cycle (SDLC) process to assess its suitability for the software team. A gap analysis tool evaluated the software development process and benchmarking process performances. Most importantly, in this activity, the analysis focused on diagnosing which practices have been accidentally left out, deliberately eradicated, and practices lacked in procedures, activities, or which skills or methods still need to be developed. The output from the second activity serves as the input to the second phase, which is the development phase.

3.2. Development Phase

The development phase aims to understand the gap analysis report and develop the process guideline based on the selected standard, ISO 29110, and selected software implementation (SI) for the software team. The development phase consisted of two activities: developing the process guideline and reviewing the process guideline.

The development of the process guideline focused on providing a systematic guide to the software development process, including detailed procedures, roles, responsibilities, and templates. The primary reference for the guidelines structure was the ISO 29110 Part 5-1-2: Software Implementation (SI) Basic Profile as a primary source. The objective, activity description, input/output products of each activity in SI from ISO 29110-5-2 was understood first. Depending on the requirements, processes were added or adopted, removed, or updated. Once the process guideline was developed, a review meeting was conducted to discuss the process guideline. Feedback from the review meeting was used to revise the process guideline.

3.3. Implementation Phase

The implementation phase aims to implement the proposed process guideline in a pilot project. The activity involved in the implementation phase consisted of introducing the process guideline to related members, selecting a small development project as a pilot project, preparing a pilot project description, and implementing the pilot project.

A meeting session was carried out among the related members to introduce the pilot project flow according to the proposed process guideline. The members were assigned tasks as software leader, software engineers, developers, and testers during the meeting. Each member was given a softcopy of the process guideline. If there were any feedback from the related members, the process guideline was updated.

The next crucial step was to select a small development project as a pilot project. During this activity, a meeting was conducted with the management to discuss selecting a suitable development project for the pilot project. The decision was necessary because the project's timeline was limited, and the project had to suit the given timeline so that there was adequate time for project completion and analysis.

The next activity was preparing the pilot project description. The details of the pilot project, requirements, processes involved during implementation based on the process guideline, roles involved, and other information were included in the pilot project description.

Finally, the pilot project was implemented to test the effectiveness of the developed process guideline. The software leader was responsible for confirming the practices executed while implementing the project. At the end of the project, the software leader evaluated the implementation results.

3.4. Analysis Phase

The analysis phase aims to gather feedback on the pilot project from related members and analyse the result. The input and suggestions were crucial to improving the process guideline. A post-mortem was conducted among the pilot project team members after project completion. The post-mortem was scheduled as a one-hour meeting to discuss everything that went right, wrong, and everything in between. The post-mortem was scheduled right after completing the pilot project while it was still fresh in each related member's mind. The primary aim of the meeting was to conduct a constructive review of the project and determine a better process guideline.

The feedbacks included questions from the author to the team members such as: did the team get to achieve their target from the pilot project, which of the methods and practices worked exceptionally well, which of the methods used were difficult to use, how would team members do things differently next time to avoid such difficulties, what was the better suggestion for the practices, and what was the most gratifying satisfying part of the project. The feedbacks were gathered and analyzed to get the process guideline updated according to the software team's preference.

4. Findings and Discussion

One of the main deliverables of the study was the process guideline. The process guideline was developed, tested in a pilot project, and revised for further improvement. Feedbacks from the review meeting is as follows:

- The development team found the process guideline a bit complicated when used for the first time as they were unfamiliar with the terminologies.
- The roles in the process guideline were proposed to be combined as the development team was small.
- A series of ISO 29110 training sessions were suggested to be held to understand the standard better.

Some alterations and changes to the process guideline were required to be made to be a comprehensive reference for the software development processes in the software team. The contents of the pilot project description should be reviewed while also adhering to the ISO 29110-5-1-2 SI guideline.

Due to the limitation of the project period, it was impossible to develop the whole process of the ISO 29110-5-1-2 standard in the process guideline. Only the software implementation (SI) processes were considered in this study to address problems identified in the software team. However, it would be more efficient for the organization's project developments in the future if the process guideline included the project management processes for reference.

5. Conclusion

The goal of the study was to provide a comprehensive process guideline to improve the software development process in the software development team. ISO 2911-5-1-2 standard was used as the project's primary reference. This standard was chosen because it was explicitly developed for VSE, as the software team had less than 25 developers. The developed process guideline was reviewed and implemented in a pilot project. Feedback obtained from the implemented pilot project improved the process guideline.

In summary, the project's result contributes significantly to the software team's development activities. The study helped spread awareness of the benefits of software process improvement to the organization and its impact on the products. Furthermore, the organization learned about the ISO standard, especially on the ISO 29110 Software Implementation (SI) processes and identified the weaknesses in the software development process, and solved it with the developed process guideline and ISO 29110.

Acknowledgments

The authors would like to thank the software development team of Realtek Malaysia for their support and cooperation regarding their involvement in this study.

References

[1] Mittner, J., & Buchalcevová, A. (2014). Towards the IT Support of Processes in Small Software Companies.

[2] Suteeca, K. (2020). A Software Process Gap Analysis Methodology for Very Small Entity. 2020 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering, ECTI DAMT and NCON 2020, 190–193. https://doi.org/10.1109/ECTIDAMTNCON48261.2020.9090745.

[3] Unterkalmsteiner, M., Gorschek, T., Islam, A. K. M. M., Cheng, C. K., Permadi, R. B., & Feldt, R. (2012). Evaluation and measurement of software process improvement-A systematic literature review. IEEE Transactions on Software Engineering, 38(2), 398–424. https://doi.org/10.1109/TSE.2011.26.

[4] Wongsai N., Siddo V., W. R. (2015). Factors of Influence in Software Process Improvement. 2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE), 12–17.

[5] Tsunoda, M. (2018). Analyzing Software Maintenance Cost Based on Work Efficiency and Unit Cost. 2018 IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD), 102–108. https://doi.org/10.1109/BCD2018.2018.00024.

[6] Olumide Akerele. (2018). System dynamics modelling of the impact of agile practice on the quality of continuous delivery projects. Innovations in Systems and Software Engineering, 14(3), 183–208. https://doi.org/10.1007/s11334-017-0296-z.

[7] Zapata Jaramillo, C. M., Valderrama Betancur, J., & Jimenez Pinzon, L. D. (2015). Representation of CMMI-DEV practices in the Semat kernel. IEEE Latin America Transactions, 13(10), 3476–3481. https://doi.org/10.1109/TLA.2015.7387257.

[8] Keshta, I. (2019). A model for defining project lifecycle phases: Implementation of CMMI level 2 specific practice. Journal of King Saud University - Computer and Information Sciences,. https://doi.org/10.1016/j.jksuci.2019.10.013.

[9] Humphrey, W. S. (2011). The personal software process. Software Process Improvement, November, 193–200. https://doi.org/10.1109/9781118156667.ch5.

[10] Humphrey, W. S. (2002). Team Software Process (TSP). Encyclopedia of Software Engineering, November. https://doi.org/10.1002/0471028959.sof352.

[11] Laporte, C. Y., O'Connor, R. V., & Paucar, L. H. G. (2015). Software engineering standards and guides for very small entities: Implementation in two start-ups. ENASE 2015 - Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering, 5–15. https://doi.org/10.5220/0005368500050015.

[12] O'Connor, R. V., & Laporte, C. Y. (2017). The evolution of the ISO/IEC 29110 set of standards and guides. International Journal of Information Technologies and Systems Approach, 10(1), 1–21. https://doi.org/10.4018/IJITSA.2017010101.