Identifying Software Engineering Challenges in Software SMEs: A Case Study in Thailand

Chaiyong Ragkhitwetsagul[*], Jens Krinke[†], Morakot Choetkiertikul[*], Thanwadee Sunetnanta[*], Federica Sarro[†]
[*]SERU, Faculty of Information and Communication Technology (ICT), Mahidol University, Thailand
[†]CREST, UCL Computer Science, University College London, UK

***Abstract*—Small and medium-sized software enterprises (SSMEs) are a vital part of emerging markets. Due to their size, they are not capable of adopting advanced software engineering techniques or automated software engineering tools in the same way large and ultra-large companies are. We study the software engineering challenges in SSMEs in Thailand, an emerging market in software development, using semi-structured interviews with four SSMEs. After performing a thematic analysis of the interview transcripts, we found a number of common challenges such as lack of testing, code-related issues, and inaccurate effort estimation. We observed that in order to introduce advanced automated software engineering tools and techniques, SSMEs need to adopt contemporary best practices in software engineering like automated testing, continuous integration and automated code review. Moreover, we suggest that software engineering research engage with SSMEs to enable them to improve their knowledge and adopt more advanced software engineering practices.**

***Index Terms*—empirical study, case study, software SMEs**

I. Introduction

Small and medium-sized enterprises (SMEs) are vital to economic growth of a country and account for 95%–99% of all enterprises in 36 OECD member countries [1]. Software SMEs (SSMEs) including software startups are among the key drivers in software industry and important for a country's competitiveness and innovation [2]. For example, a recent study [3] shows that the Thai software industry accounts for $4 billion in 2020 with a steady growth over the years. The country's software sector comprises of over 8,000 software companies, large and small, with more than 100,000 software engineers [3]. With the transformation plan known as Thailand 4.0, the software industry has become a cornerstone for the long-term plan of the country as it supports the development of technology clusters and future industries [4], [5]. Similar trends are also found in other Asian developing countries [6]–[8].

Nonetheless, SSMEs are facing several challenges while developing their software products. A technical report [9] outlines some of the major weaknesses of SSMEs include configuration management, quality assurance, and project assessment and control. Unlike large or ultra-large software companies, SSMEs do not have much manpower. Moreover, SSMEs may still use traditional software development approaches and tools and rely heavily on manual tasks performed by their programmers, which are costly and tedious approaches. Thus, by identifying the software engineering challenges faced by SSMEs, we can recommend state-of-the-art automated software engineering (ASE) techniques that SSMEs can adopt to reduce the cost and increase the speed of their software development and delivery. Only when the challenges leading to the weaknesses are understood can they be addressed and advanced software engineering practices be adopted.

Our study aims to fill in the gap by investigating the software engineering challenges that specifically occur in SSMEs located in the Asian culture, especially in Thailand, and also to study their tool usage for future recommendation of automated software engineering (ASE) tools and techniques. Due to the different working cultures from Western region [10], this study reveals new insights into Thai, and possibly Asian, SSMEs.

II. The Study

This study investigates current practices being used in the day-to-day software development routines in Thai SSMEs and identifies the challenges they are facing in terms of costs, time, and software quality that prevent the adoption of advanced software engineering practices. According to the guidelines by Benbasat et al. [11], we aim to study a few entities (i.e., SSMEs) in their natural setting without experimental control or manipulation. Thus, a case study is an appropriate method for our aim. We use semi-structured interviews [12] and thematic analysis [13] to extract the common challenges found by the companies. We start by identifying research questions. Then, we design the interview guide and perform semi-structured interviews with the companies' employees. Lastly, we perform thematic analysis on the transcript and create a theme map to identify common software engineering challenges.

A. Research Questions

We aim to answer the following research questions:

1) **RQ1 (SE Challenges):** *What are the challenges in the SSMEs' day-to-day software development?* The answer to this research question will identify the issues that need to be addressed in order to adopt advanced software engineering practices.
2) **RQ2 (Current Practices):** *What tools are being used?* By considering the tools the companies are using as a proxy, we can gain insights into their current practices. The answer to this question will help us to align recommendations of advanced practices and automated tools and techniques with the current practices.

To be published in the 29th International Conference on Software Analysis, Evolution, and Reengineering (SANER'22) in Honolulu, Hawaii.

TABLE I
STUDIED SSMEs

Company	Products	No. of Developers
Company A	E-Learning platform	15
ProGaming	Web and mobile games	10
Roots	Enterprise solutions	40
Zwiz.AI	Enterprise AI chatbot	6

B. The Studied Companies

We study the challenges and practices of four Thai SSMEs: Company A (pseudonym), ProGaming, Roots, and Zwiz.AI. An overview of the four companies is shown in Table I. Their software products/services cover different business types including games, e-learning, chatbots, and enterprise solutions. Company A offers an online e-learning platform with more than 1,800 courses, 10,000 users a month, and 280 educators. ProGaming has developed more than 50 games, which have been downloaded 2 million times. They also provide game development services to Thai organizations. Roots offers business consultancy and serves several large enterprises in Thailand. Zwiz.AI provides AI chatbot services which are serving more than 30,000 businesses and helping more than 10 million users. All of them are small (<50 employees) and three of them are very small (<25 employees).

C. Interview Design

The goal for the interviews is to explore the challenges in software development in the four companies. We perform semi-structured interviews which are based on a general grouping of topics and questions, instead of a predefined set of questions [12]. This method performs a verbal discussion with one programmer at a time. The interview script is updated after each interview especially when the interviewees start providing the same answers (saturation effect). The interview process terminates when no new insights or observations are made. The interview is recorded and then transcribed. The maximum duration of each interview is set to 30 minutes.

The interview participants are suggested by the executives of the companies based on their suitability for the project. The subjects must satisfy the following inclusion criteria to be included in the interview: (1) the subject must be a full-time employee and must have worked at the company for at least one month, (2) the subject must be involved in the software development such as a CTO, a technical lead, or a developer.

To address the possibility of participants being forced by their managers to participate, the authors took multiple steps to mitigate the issues and the participation or non-participation should not have had any negative effect on the employees.

D. Semi-structured Interviews

The semi-structured interviews occurred at the companies' offices in February 2020[1]. The first author visited each company and performed the semi-structured interviews in a dedicated room provided by the companies. The interviews were in Thai and the recordings from the interviews were transcribed. The initial questions used for the interviews are listed in Table II. These questions were used to start the conversations and the interviewer could ask more follow-up questions depending on the answers of each interviewee.

TABLE II
INITIAL QUESTIONS USED IN THE INTERVIEWS

	Please explain your role at the company. How long have you been working at the company?
RQ1	Please explain your day-to-day activity. How do you develop the software product? What works well in your company's software development? What would you suggest to improve?
RQ2	Please explain the tools you use during software development.

E. Thematic Analysis

After transcribing the interview recordings, reflexive thematic analysis, which is widely used to analyse qualitative data, was applied on the transcript units. We followed the guidelines by Braun and Clarke [13] with a deductive approach in which coding and theme development were directed by the challenges in software engineering. The first, third, and fourth authors (the coders) were assigned a set of transcripts to code. They carefully went through a transcript and assigned one or more codes to the transcript units. The codes were in English to facilitate generalisation and enable working with authors from different countries and comprised of software engineering challenges such as "lack of unit testing", "coding style violations", or "mismatched requirements". The coders shared the set of codes and kept adding new codes if the existing ones did not match with what they found in the transcript. Then, the assigned codes by one coder were validated by another coder and each conflict was discussed until a consensus was found. Next, the first and second authors identified themes by sorting the validated codes into groups according to their content, splitting codes into different groups, or discarding the codes. The process was repeated multiple time until no further change could be made. Finally, a set of common challenges in the companies' software development emerged.

III. Results

We interviewed 20 software engineers and software-related employees (details are shown in Table III). The interviewees' roles range from management positions, e.g., Chief Technology Officers (CTO) to developers. The working experience ranges from 4 months to 11 years.

A. RQ1: Identified SE Challenges

The identified software engineering challenges are shown in Figure 1. We discuss each of the challenges in detail below.

[1] One interview occurred via a conference call in March 2020.

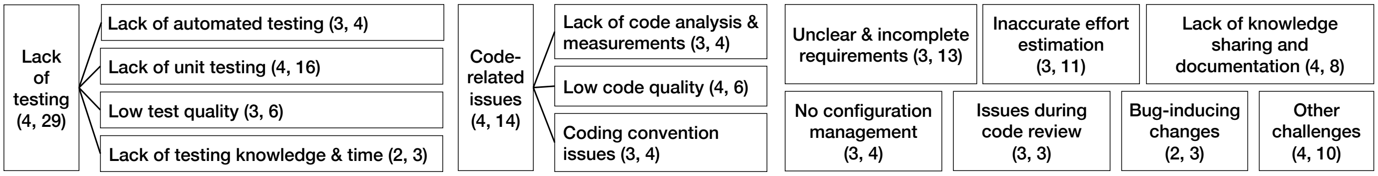
Lack of testing (4, 29)
Lack of automated testing (3, 4)
Lack of unit testing (4, 16)
Low test quality (3, 6)
Lack of testing knowledge & time (2, 3)
Code-related issues (4, 14)
Lack of code analysis & measurements (3, 4)
Low code quality (4, 6)
Coding convention issues (3, 4)
Unclear & incomplete requirements (3, 13)
Inaccurate effort estimation (3, 11)
Lack of knowledge sharing and documentation (4, 8)
No configuration management (3, 4)
Issues during code review (3, 3)
Bug-inducing changes (2, 3)
Other challenges (4, 10)

Fig. 1. Theme map of the software engineering challenges. The number (x, y) show the number of companies (x) and the occurencces of the challenge (y).

TABLE III
INTERVIEWED PARTICIPANTS

Company	Participant (Experience at the company in years)
Company A	Chief Technology Officer (6), Developer (2), Senior Developer (1), Chief Operating Officer (6), Developer (1.5)
ProGaming	Managing Director & Technical Lead (11), Lead Developer (6), Project Manager (5), Developer (3), Developer (1)
Roots	Chief Technology Officer (5), Senior Developer (4), Senior Developer (2), Developer (2), Developer (0.7)
Zwiz.AI	Chief Technology Officer (2), Developer (2), Developer (1), Developer (0.25), Developer (0.33)

1) Lack of testing: The most prominent challenge that occurred during the interviews, coding, and theme mapping is the lack of software testing, which occurs across all four companies. This includes the lack of using automated testing (mentioned 4 times in the interviews by different interviewees), lack of unit testing (mentioned 16 times), low test quality (mentioned 6 times), and lack of testing knowledge and time for testing (mentioned 3 times). The companies focus mainly on implementing the software and testing is mostly performed manually at the end of the development. Unit testing is not usually performed and the developers also do not value unit tests during the development. One interviewee also mentioned a lack of regression testing. This is in line with the previous findings that tests are not well adopted in SSMEs [14] and in Thailand [10]. Moreover, the adoption of automated testing is a challenge because of the lack of time and knowledge to study and deploy in the companies.

2) Code-related issues: The second challenge is about code-related issues. This challenge also applies to all the four companies. We found three issues as follow: (1) the lack of code analysis and measurements (mentioned 4 times) such as not using any static or dynamic analysis tools to analyse source code, (2) low code quality (mentioned 6 times) including buggy code, duplicated code, legacy code that is hard to understand, and code that is difficult to reuse and needs to be refactored, and (3) difficulty of maintaining or using coding conventions in the company (mentioned 4 times).

3) Unclear and incomplete requirements: The third challenge is about requirements (mentioned 13 times by 3 companies). The interviewees explained that they found some of the gathered requirements unclear, too vague, or incomplete. Some of the requirements mismatched the customers' needs. Moreover, they also mentioned an issue of "scope creep" due to new requirements being added during the development.

4) Inaccurate effort estimation: The fourth challenge is difficulties in estimating the amount of work to be done (mentioned 11 times by 3 companies). The interviewees explained several issues. For example, they often needed to put in more time than what was planned for and worked overtime. There were also unplanned tasks added in a sprint. Moreover, some of the interviewees mentioned that they could not make correct estimation to complete the assigned tasks.

5) Lack of knowledge, sharing, and documentation: Some interviewees suggested that knowledge sharing and documentation could be enhanced (mentioned 8 times by 4 companies). For example, establishing ways to transfer knowledge to new team members to get started quickly or having training to enhance work-related skills.

6) No configuration management: There are a few challenges also in software configuration management (mentioned 4 times by 3 companies). For example, difficulties in managing configurations of environments such as supporting different browsers and screen sizes, different setups on local development and deployment, and no information of stable versions of the required frameworks or tools.

7) Bug-inducing changes: Three interviewees mentioned that new changes resulted in additional bugs (mentioned 3 times by 3 companies). This also relates to the lack of testing to catch such bugs induced by the changes.

8) Issues during code review: We have found that the companies had a few issues performing code review (mentioned 3 times by 2 companies) such as the changes to be reviewed did not have all the files required for the review or no automated tools was employed during the code review. One of the companies does not perform code review in the development.

9) Other challenges: Some of the challenges cannot be classified into the previously mentioned challenges. Examples include lack of software process, lack of automated deployment, and difficulties in cross-team communication.

To answer RQ1, we found that the SSMEs in Thailand are facing several software engineering challenges in their day-to-day software development. The four main challenges include lack of testing, code-related issues, unclear and incomplete requirements, and inaccurate effort estimation.

An overall observation that can be made is that the lack of testing and the lack of adoption of automated code review impacts challenges 2, 5, 7, and 8. Such contemporary best practices need to be adopted before further improvements can be made.

B. RQ2: Identified Current Practices

To gain more insights into the current practices of the companies, we asked the participants about automated tools that they regularly use. We observed some interesting findings as follows.

1) Unit testing frameworks: Despite using continuous integration, the companies did not use unit testing frameworks. Interestingly, there was one interviewee mentioning using JEST and Mocha, the unit testing frameworks for Javascript. However, the other interviewees from the same company mentioned the lack of unit testing adoption. This mentioning of unit testing frameworks possibly comes from the interviewee's own expertise and may indicate that the company still needs to improve the company-wide adoption of testing.

2) Code structure and formatting are locally enforced: The interviewees from the four companies explained that they used linters, such as ESLint or PyLint, and code formatters, such as Prettier, in their IDEs. Nonetheless, from answers to RQ1, we found that some companies do not perform code review nor include any automated tools during code review. Thus, code structure, formatting, and coding conventions may not be enforced at company level but at individual level.

3) Mixed use of project management tools: We observed that the four companies use dedicated project management tools such as Asana, Trello, or Clickup, to manage their tasks and track their projects' progresses. However, two companies also mentioned using Google Sheets, the general purpose spreadsheet tool to handle some parts of their projects.

4) Mixed use of communication tools: The interviewees from three companies mentioned that they mainly used LINE and Discord, which are popular chat applications in Thailand. Only one company used a dedicated communication tool, MatterMost. Since LINE and Discord are mainly personal chat applications, this shows that the developers may mix their work with their personal life. In part, this can be from the specific culture in Thailand that people tend to respond faster on their personal chat platforms than work-related platforms. There is no mention of having internal Q&A forums or knowledge management platforms possibly due to the relatively small size of the companies.

> To answer RQ2, we found that some of the SSMEs have mixed use of dedicated project management and general-purpose tools such as spreadsheets to manage their projects. Moreover, they also use personal chat applications for work. Lastly, some tools are being used or known only at the individual level such as code formatters or linters, which may cause inconsistencies at the company level.

IV. Discussion

Outcomes: The case study is part of an ongoing multi-national industry-academia collaboration project to support SSMEs in the adoption of suitable automated software engineering tools and techniques, similar to those that are successfully used in large and ultra-large companies.

The presented study was intended to identify the challenges of SSMEs in an emerging market like Thailand and address the identified challenges through automated software engineering. While performing this study as part of the project, we identified one particular challenge which needs to be addressed before further advanced practices or automated software engineering can be adopted: lack of testing. Contemporary best practices in software engineering are to use automated testing tools to test the software in continuous integration settings, and automated software engineering often requires a large amount of testing. Without automated testing, there is no foundation for improving the software development process.

Lessons Learned: An important lesson learned is the observation that before one can attempt to support SSMEs in the adoption of automated software engineering tools and techniques, one must first ensure that the SSMEs have adopted contemporary best practices in software engineering. In the context of the ongoing project, the study results led to a change of the project's focus and aims: Instead of introducing the most advanced automated software engineering tools and techniques, the project will focus on the effect of the adoption of best practices in software development like automated testing, continuous integration, and automated code review.

Takeaway Messages: Knowledge transfer between software engineering research and industry has been a key factor in the success of the adoption of automated software engineering tools and techniques. Such cooperation and knowledge transfer usually occurs in large and ultra-large companies. SSMEs, especially in emerging markets, often are not able to participate in such cooperation and knowledge transfer. Research should, therefore, engage SSMEs in order to facilitate the adoption of automated software engineering tools and techniques so that the gap between SSMEs and large and ultra-large companies does not widen.

V. Related Work

Laporte et al. [15] and Habra et al. [16] report their experience of applying ISO29110, a software process guidelines for very small entities (VSEs), to companies. Basri et al. [17] study acceptance of software process improvements in VSEs. Sunetnunta et al. [9] report preliminary results from a gap analysis of 39 Thai VSEs' software development activities. They found that the major weaknesses include configuration management, quality assurance, and project assessment and control. Phongpaibul and Boehm [10] show that the cultural factors affect the adoption of software process improvement in Thailand. Tuape and Ayalew [18] identify that technical factors are the key factors affecting software processes of African SSMEs. Larrucea et al. [2] also report common barriers to software process improvement in SSMEs. Our study investigates the challenges that are not only related to the software process but also to best practices and tools. We found some results that correlate with these previous studies as discussed in Section III.

A few studies [19], [20] identify important factors or best practices in software development such as unit testing, code reviews, coding guidelines, code analysis, version control, and project management tools. Garousi et al. [21] discuss the challenges of test automation management. Ponsard and Deprez [22] report that requirements, technical debt, test/release management, and project management are challenges faced by Belgian SSMEs. Jones et al. [23] study the challenges of DevOps adoption in UK SSMEs. Compared to our study, we found similar challenges such as lack of unit testing and requirement management.

Other studies identify software engineering challenges based on different phases throughout the software development life cycle. Shah et al. [24] identify challenges in the requirement engineering such as communication gap and customer involvement. Mäntylä et al. [25] and Shanin et al. [26] identify challenges in software delivery, deployment, and CI/CD.

VI. Threats to Validity

External validity: The study is performed on four Thai SSMEs and the findings may not be generalised to all SSMEs in Thailand and other developing countries. We mitigated this threat by selecting SSMEs from different domains. Due to cultural differences, some of the findings may only be found in Asian regions and not be applied to SSMEs in other regions.

Internal validity: The selection of participants may affect the validity of the identified challenges. We mitigated this threat by using an inclusion criteria to control the experience with the company's software development process.

VII. Conclusion

This paper studies the software engineering challenges that SSMEs experience and the current practices used in the companies. The study is conducted as a case study using semi-structured interviews and thematic analysis of four Thai software companies. The result shows that the studied companies are facing challenges in contemporary software best practices, which are often well-adopted in large or ultra-large software companies, such as software testing, code-related issues, and requirements. Moreover, we found that the SSMEs are using general-purpose tools such as spreadsheets or personal chat applications for project management and communication instead of dedicated tools. The lack of testing not only affects the quality of the software, but it also prevents the adoption of advanced practices and automated software engineering tools. Such practices and tools are often the result of cooperation and knowledge transfer between software engineering research and large or ultra-large software companies. We call for more engagement of SSMEs in software engineering research to ensure that no one is left behind.

Acknowledgements

This work is supported by the Newton Fund and funded by the Royal Academy of Engineering, UK (Industry Academia Partnership Programme – 18/19 – IAPP18-19/74). It was approved by both universities' research ethics boards. The authors have no financial or proprietary interests in the four studied companies.

References

[1] OECD, *OECD SME and Entrepreneurship Outlook 2005*. OECD Publishing, 2005.

[2] X. Larrucea, R. V. O'Connor, R. Colomo-Palacios, and C. Y. Laporte, "Software Process Improvement in Very Small Organizations," *IEEE Software*, vol. 33, no. 2, pp. 85–89, Mar 2016.

[3] Digital Economy Promotion Agency, "Thailand's Digital Industry Survey 2020 and 3-Year Prediction," https://www.depa.or.th/th/article-view/index-3-years-since-2563, 2021.

[4] W. D. Royal Thai Embassy, "Thailand 4.0 - Agenda 2: Development of Technology Cluster and Future Industries," https://thaiembdc.org/thailand-4-0-2, 2016.

[5] J. Wongwuttiwat and A. Lawanna, "The digital Thailand strategy and the ASEAN community," *EJISDC*, vol. 84, no. 3, p. e12024, May 2018.

[6] P. Samuel, "Vietnam's digital transformation plan through 2025," https://www.vietnam-briefing.com/news/vietnams-digital-transformation-plan-through-2025.html, 2021.

[7] Ministry of Commerce and Industry (India), "Make In India," https://www.makeinindia.com/, 2014.

[8] K. Crismundo, "Industry 4.0 plans to help PH improve innovation rankings," https://www.pna.gov.ph/articles/1114871, 2020.

[9] T. Sunetnanta, S. Suwannaroj, and P. Sangpar, "ISO/IEC 29110 for Competitiveness - Challenges of Digital Cluster Development in Thailand," ISO/IEC JTC 1/SC 7 Software and Systems Engineering Working Group 24, Tech. Rep., August 2016.

[10] M. Phongpaibul and B. Boehm, "Improving quality through software process improvement in Thailand," *ACM SIGSOFT Software Engineering Notes*, vol. 30, no. 4, p. 1, Jul 2005.

[11] I. Benbasat, D. K. Goldstein, and M. Mead, "The case research strategy in studies of information systems," *MIS Quarterly*, vol. 11, p. 369, 1987.

[12] S. B. Merriam and E. J. Tisdell, *Qualitative Research: A Guide to Design and Implementation, 4th Edition*. Jossey-Bass, 2015.

[13] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, Jan 2006.

[14] A. Rodrigues, A. Bessa, and P. R. Pinheiro, "Barriers to Implement Test Process in Small-Sized Companies," in *Communications in Computer and Information Science*, 2010, vol. 112 CCIS, pp. 233–242.

[15] C. Y. Laporte, R. V. O'Connor, and L. H. G. Paucar, "The Implementation of ISO/IEC 29110 Software Engineering Standards and Guides in Very Small Entities," in *CCIS*, 2016, vol. 599, pp. 162–179.

[16] N. Habra, S. Alexandre, J.-M. Desharnais, C. Y. Laporte, and A. Renault, "Initiating software process improvement in very small enterprises," *IST*, vol. 50, no. 7-8, pp. 763–771, Jun 2008.

[17] S. B. Basri and R. V. O'Connor, "Organizational commitment towards software process improvement: An Irish software VSEs case study," in *ITSim'10*, vol. 3, 2010, pp. 1456–1461.

[18] M. Tuape and Y. Ayalew, "Factors Affecting Development Process in Small Software Companies," in *SEiA'19*, 2019, pp. 16–23.

[19] M. Felderer and F. Auer, "Software quality assurance during implementation: Results of a survey in software houses from Germany, Austria and Switzerland," *Lecture Notes in Business Information Processing*, vol. 269, pp. 87–102, 2017.

[20] M. Yaseen, S. Baseer, and S. Sherin, "Critical challenges for requirement implementation in context of global software development: A systematic literature review," in *ICOSST'15*, 2016, pp. 120–125.

[21] V. Garousi, M. Felderer, M. Kuhrmann, K. Herkiloğlu, and S. Eldh, "Exploring the industry's challenges in software testing: An empirical study," *Journal of Software: Evolution and Process*, vol. 32, no. 8, 2020.

[22] C. Ponsard and J.-C. Deprez, "Helping SMEs to better develop software," in *ICSE'18 (SEIP)*, 2018, pp. 213–214.

[23] S. Jones, J. Noppen, and F. Lettice, "Management challenges for DevOps adoption within UK SMEs," in *QUDOS'16*, 2016, pp. 7–11.

[24] T. Shah and S. V Patel, "A Review of Requirement Engineering Issues and Challenges in Various Software Development Methods," *IJCA*, vol. 99, no. 15, pp. 36–45, 2014.

[25] M. V. Mäntylä and J. Vanhanen, "Software deployment activities and challenges - A case study of four software product companies," in *CSMR'11*, 2011, pp. 131–139.

[26] M. Shahin, M. Zahedi, M. A. Babar, and L. Zhu, *An empirical study of architecting for continuous delivery and deployment*. Empirical Software Engineering, 2019, vol. 24, no. 3.