Software Development Practice: How Organisation Dynamics Inhibit the Utilization of Process Tools in Small Software Companies

Micheal, Tuape
Department of Software Engineering
Lappeenranta-Lahti University of Technology, Lappeenranta, Finland

Victoria Hasheela-Mufeti
Department of Computing, Mathematical and Statistical Sciences, University of Namibia, Windhoek, Namibia

Petrus Iiyambo
Department of Computing, Mathematical and Statistical Sciences, University of Namibia, Windhoek, Namibia

Anna Kayanda
Information Systems Department, College of Business Education, Dar es Salaam, Tanzania

JUSSI, Kasurinen
Department of Software Engineering
Lappeenranta-Lahti University of Technology, Lappeenranta, Finland

ABSTRACT

Low-quality software products are synonymous with small software companies, yet thousands of process tools and methods are available, although unfortunately, they remain unutilized. The utilization of these process tools are said to be very challenging to small software companies with the difficulty of adaptability tagged to the context in which the companies operate since the process tools are not designed to take care of the operational context of the small software companies. This survey study was undertaken to empirically investigate the implication of the organizational dynamics on using process tools in SSCs. A total of 115 respondents from Tanzania, Namibia, and Ghana were reached through a structured questionnaire. Our findings reveal that the ad-hoc behaviors and attitude breed a culture that inhibits procedural and process-intensive practices synonymous with most software engineering process tools. This finding is interesting because this area has been underestimated while looking at software practice, yet it is significant in understanding practice in relation to software process tools, significantly how small software companies can be helped to adopt the different tools and also provide a mechanism to help the designers of process tools make tools that are usable in the context of the practice of SSCs.

CCS CONCEPTS

• **Software organization and properties**; • **Software notations and tools**; • **Software creation and management**;

KEYWORDS

Small software companies, Utilization of process tools, Software development practice, Organisation dynamics

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICSIE 2021, November 12–14, 2021, Cairo, Egypt
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8431-5/21/11...$15.00
https://doi.org/10.1145/3512716.3512722

ACM Reference Format:
Micheal, Tuape, Victoria Hasheela-Mufeti, Petrus Iiyambo, Anna Kayanda, and JUSSI, Kasurinen. 2021. Software Development Practice: How Organisation Dynamics Inhibit the Utilization of Process Tools in Small Software Companies. In *2021 10th International Conference on Software and Information Engineering (ICSIE) (ICSIE 2021), November 12–14, 2021, Cairo, Egypt.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3512716.3512722

1 INTRODUCTION

Low-quality software products have been associated with small software companies (SSCs), despite different process tools to guide software engineering processes. Studies continue to report poor engineering practices accompanied by volatile development environments in which the SSCs are faced with high competition, short time to market, impossible deadlines and limited resources [29],[31]. Different solutions have been developed to transform development practices, among which are adaptive methods that have seen the companies leapfrog in the names of being agile to no avail. It is important to note that the SSCs are significant to the software industry, that the SSCs produce over 80 percent of software products on the market and the ubiquitous world of today is run on software. Software significantly touches every aspect of human lives, and a successful software industry is in the world's interest [8].

Researchers have listed countless factors affecting software development generally, and another author has specifically contextualized the factors to SSCs [18]. Unfortunately, very little seem to have yielded results in an attempt to address the said factors to put the predicament at bay. We take an interest in the authors' work in [33] who cited four factors affecting software development processes: technical factors, organization factors, the business environment, and the governance in the SSCs. Similarly, other researchers in [28],[8] have also raised similar views and, in addition, associated the different factors with subsequent complexities, which among others, make it difficult for the SSCs to use the prosses tools in the development processes [31],[32]. The researchers propose a process adoption framework and cite the four factors as the influencing factors to support the envisaged adoption mechanism to address the limited utilization of the process tools.

In this paper, we singled out the organizational factors, which we referred to as organizational dynamics. We investigated the significant elements that influence the people, the purpose, the work patterns, the culture, the leadership, and the structure of that particular organization affecting the delivery of the intended products expected to be produced by that specific organization. Most processes in software development, particularly requirements engineering [4] and software testing, are people-intensive processes with extensive people involvement [9]. Given that people play a significant role in an organization and the dynamics in an organization affect the people. From this, we believe that the said dynamics affect the utilization of process tools, which has not been explored and fronted for discussion. Similarly, we also front the notion that paying attention to the organizational dynamics while developing an adoption mechanism and the complexities that arise requires taking a specific interest in investigating what organization practices influence effective utilization of process tools in SSCs.

To understand this, we first looked at the organizational culture and specifically the ad-hoc [28] behaviors in an organization occasioned by development processes due to technology change. Secondly, we looked at the organization's structure, believing that it creates complexity on the management practice in an organization which is viewed through the lenses of the number of roles one takes in an organization. Additionally, the number of roles staff take up, the attitudes in organizations that create a stereotypical tendency, for instance, the impression that it could be challenging to adopt tools and techniques for development, the 'confidence' in products followed by an attitude of no testing. In some cases, the companies have a software testing strategy that may not be referred to, yet it is developed in-house. Additionally, what seems to look like organization flexibility by taking up additional responsibility for the already available routine work during a typical project, coupled with the flexibility in resource allocation, exhibited by liberty to switch resources from one activity to another. These concerns aggregate to organizational behavior that affects the ability of a team to follow procedural processes and look at multiple tasks in processes as backbreaking while opting for ad-hoc practices. Moreover, most tools in software engineering have multiple tasks and need time, making it difficult for the SSCs to take up these tools.

Investigating the parameters will enable us to make meaningful conclusions on how organization dynamics make it complex for SSCs to utilize process tools. The rest of the paper is structured as follows: In section 2, we present related work, section 3 covers the methodology used in the study, in section 4 we present the results, which are then discussed in section 5 and finally, the conclusion and future work in section 6.

2 RELATED WORK

Contextual complexities while developing software have become a matter of concern to different authors [2], [21], [24], with significant attention given to implications of entrepreneurial characteristics, maturity models and software process improvement, all discussing business success of the SSCs, project management and the technical aspects some of which are covered the software body of knowledge as process areas. Although these initiatives have ended up with new tools and frameworks that are unfortunately not used to transform software development processes in the SSCs, they have contributed significantly to literature in understanding the challenges in practice.

The interests of some scholars in the associated challenges faced in the production of software by SSCs have been given to the organizational aspects of SSCs with significant concern on the entrepreneurial characteristics of SSCs [24],[27]. Although the authors highlight the importance of success in the SSCs, the studies are more concerned about the organizational cultural characteristics for the organization's survival in areas of marketing and engagement with customers. Giving no attention to the fact that these same cultural issues harm the development practices and the development tools that give purpose for the company, especially that quality is related to marketing and satisfaction of the customers.

The proponents of SPI [19],[21],[22] and [23] present evidence to the effect that SPI helps transform the practice and at the same time contend that SPI models such as ISO 9000 and the Capability Maturity Model Integrated (CMMI) have assisted software development organizations by harnessing their experience and supporting them so that production of software is on time, within budget and to a high level of quality. However, this presents contraindications to SSCs. This challenge is further reinforced by [19], who acknowledge the lack of adoption of standards in SSCs and point out the perception that these tools have been developed for large software companies and not with the SSCs in mind. The reasons cited, such as limited resources regarding people, money and organization challenges, are also alluded to in [21]. The authors proposed what they referred to as whitewater interactive systems development with object models to address the specific needs of SSCs. Additionally, the authors in [8] list organizational phenomena such as culture and structure, which set out a significant effect on the company's ability to deliver technical aspects of software production and improve the processes of producing software as a whole. A similar argument is elaborated by [20], who examine the impact of organizational culture in supporting knowledge sharing on the success of SPI.

Although all these researchers highlighted significant points on the organizational dynamics of the SSCs and subsequent challenges it poses to the company's, it is notable that they all highlighted the fact that the SSCs find adoption of specific tools complex[32] and, as such, propose tools designed explicitly for the SSCs which are unfortunately not utilizable. However, it is important to note that the SSCs are not of the same in character and with organizational dynamics, so bounding then as all small and tailoring tools for all SSCs is still a futile attempt. This, therefore, is the gap that remains unresolved, and in this study, we pay attention to the organizational characteristics of the SSCs and their implications on the utilization of tools, as highlighted in [31]. In this study, we hypothesize that SSCs are different and that the difference in character implies culture and practice. The study seeks to illustrate how the differences in the organizational dynamics affect the technical utilization of process tools.

3 METHODOLOGY

We use a quantitative design study in a cross-sectional survey with close-ended questions answered with a type 5 Likert scale. Quantitative study designs are commonly used in software practice research[15, 25], to understand the organizational dynamics in the

SSCs, the characteristics of practitioners and their usage of software tools in software practice by SSCs, which has limited studies.

The survey was sent out to 84, 95 and 103 for Namibia, Tanzania, and Ghana, totaling 282 companies. A total of 115 data points was returned with 30, 38 and 47 responses from the respective countries representing 26, 33, and 41 percent. Purposive sampling was used based on a criterion whose characteristics are defined for a purpose that is relevant to the study [3]. A sample used is from a population of software-intensive companies of different sizes developing software products for a wide variety of markets. The criterion of selecting the company was the number of persons in the company to fit the definition of SSCs of under 50 persons, the type of software-intensive products from the company, and the role in the software company. Purposive sampling is commonly used in software engineering studies [10],[7] as advised in [6]. Telephone calls were made to request the participants to participate in the study before mailing the questionnaire link on the Webropol survey system.

Participant time on a questionnaire was under 30 minutes. The study received a response rate of 40.8 percent, which is above the minimum 40 percent as recommended in [5]. The respondents' characteristics were such that those developing software solutions and web products made 90 and 94 hits while corporate systems and business tools hit 63 and 32, respectively. The composition in terms of roles is such that developers, software engineers, project managers and business owners were 82, 52, 34 and 25, respectively. The respondents' level of education is such that bachelors, masters, and doctorate degrees were 83, 24 and 4 than high school and vocational training 3 and 1. On gender, 76 and 24 percent in favor of the males, and the number of personnel in the companies under 5 was the majority with up to 41 followed by 21 to 25 and 26 to 30 with 11 and 14 percent respectively and the rest with single-digit percentage points.

The survey questionnaire was designed to investigate software practices, specifically the tools used in practice by the SSCs. We developed a draft set of questions aiming at covering the software practice comprehensively. Consideration was made to the size of the questionnaire and the number of questions, with the guidelines in [14]. Fifteen practitioners in the industry checked the language used in the study as familiar to the participants through a pilot. Evidence shows that researchers and industrial practitioners use different terminologies while conducting industrial studies yet, consensus on terminology is required [13]. A set of questions was dedicated to the profiles and demographics of the respondents. The questions probed data related to software companies in the context of characteristics, software development tools used in software practice with answers of type 5-point Likert scale giving participants options of 1. Never, 2. Rarely, 3. Sometimes, 4. Often, and 5. Always.

Assessment of Reliability and Validity of the questionnaire was done using Cronbach's alpha α as proposed in [11] to measure of internal consistency ("reliability"). The reliability of the 64 sub-questions measured on the 5 Likert scale given a dataset of 115 was evaluated by internal consistency analysis, using coefficient alpha α [23]. The result was evaluated to 0.92, which means that our questions' internal consistency or reliability is acceptable. According to the rule a minimum of 0.7 and above is acceptable from the ranges are from zero to one.

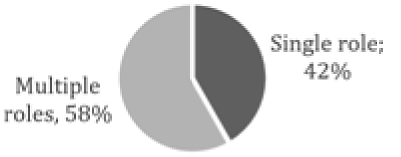

Figure 1: A representation of the respondents who have more than one role in the company against those with only one role (*n*=115)

The data analysis was done using cross-tabulation and spearman's rho correlation, cross-tabulation is a tool used to analyze categorical data. This type of data involves values that are mutually exclusive to each other. Data was collected in numbers, but numbers have no value unless they mean something. Similarly to the method used by [16], the results of testing these assumptions showed that kurtosis, skewness, and the one-sample Kolmogorov-Smirnov tests for all variables were within the acceptable range for the normal distribution assumption. In addition, the assumptions of homoscedasticity, linearity, and independence of the error terms were supported, and no influential observations were identified. In other words, there were no extreme violations to the basic assumptions underlying the chosen data analysis techniques that could justify the use of less powerful nonparametric statistics. The spearman's rank correlational coefficient calculated using equation 1 is a nonparametric test used to measure the strength of association between two variables also commonly used in software development related research like in [1, 17]. All quantitative analyses were conducted using Statistical Package for Social Science (SPSS) version 26.

$$\rho = 1 - \frac{6\Sigma d_i^2}{n\left(n^2-1\right)} \tag{1}$$

Where ρ = Spearman's rank correlation coefficient, d_i = difference between the two ranks of each observation and n= number of observations

4 RESULTS

4.1 Organization culture

4.1.1 Ad-hoc practices that get enshrined as culture. The results present a correlation between the occurrences of change in development practice and the number of software professionals in a company depicting an ad-hoc behavior that tends to become a practice and culture in a company. The results present a positive correlation, as illustrated in Table 1. We also use the question on the roles taken in the company to evaluate multiple roles Figure 1 shows that 58 percent had more than 2 roles while 42percent had only one role in the company

4.1.2 Attitude that shapes culture. Figure 2 presents the responses to whether the respondents had trouble in adopting tools during software development. Of the 115 respondents,64 stated that they rarely had trouble, 16 stated they never, 30 stated they sometimes experience difficulty, and lastly, 5 stated that they often do.

Table 1: representation of the correlations between the change in the development process due to change in technology(XT_DISRUP) and number of professionals employed in a company (NUM_PROF)n=115

Correlations				
			XT_DISRUP	NUM_PROF
Spearman's rho	XT_DISRUP	Correlation Coefficient	1.000	-.215*
		Sig. (2-tailed)	.	.021
	NUM_PROF	Correlation Coefficient	-.215*	1.000
		Sig. (2-tailed)	.021	.

*. Correlation is significant at the 0.05 level (2-tailed).

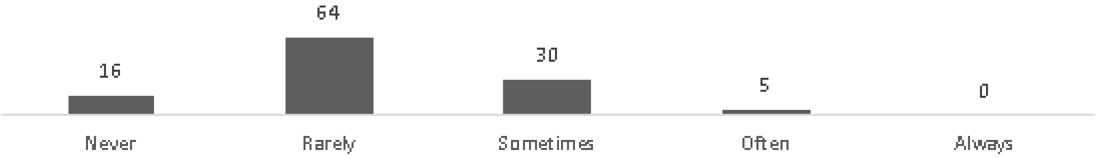

Figure 2: Representation of the answers to the question to test if the respondent has had difficulty in adopting tools during software development (n= 115)

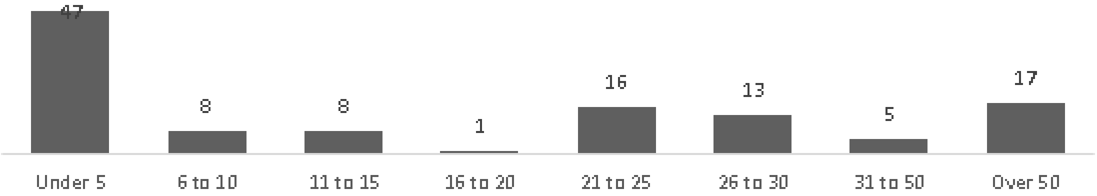

Figure 3: Representation of the number of software engineering professionals working in the companies represented by the respective respondents n =115.

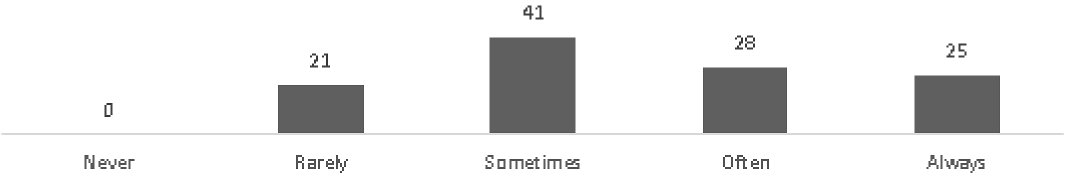

Figure 4: Responses from the question on liberty to switch resources from one activity to another n=115.

We also tested the attitude of the respondents on software test strategy developed in-house within the organization, on whether they referred to their strategy. We observed a correlation between the variables "We refer to our company test strategy" and "Approximately how many software professionals are employed by your company?" The results presented a p-value of 0.003 (spearman's rho value = 0.331) significant at 0.01 level of significance. There exists a significant positive association between the variables.

4.2 Organization Structure

4.2.1 Number of staff defining structure. As many as 98 of the 115 fulfilled the definition of the SSCs with less than 50 employees. Figure 3 shows the number of responses in each category on the number of software development professionals in a company.

4.2.2 Flexibility on resource allocation. Figure 4 shows that up to 94 of the 115 respondents chose the option to have either sometimes, often or always had the liberty to switch resources from one activity to another up to respondents, while 21 respondents chose the option of rarely and none of the respondents chose the option of never.

5 DISCUSSION

An organization's culture shapes the practice of an organization; in small organizations, ad-hoc [28] behavior has adverse effects on the procedure and intensive knowledge activities. In our case, SSCs are faced with a dynamic and versatile situation that leaves them with no choice but to change their development processes due to technology change.

Finding evidence of a correlation of the change in the development process being associated with the number of software professionals in a company speaks to this. This also comes with the pressure that makes it complex, and it tends to create room for ad-hoc practices, which are in most cases counterproductive while using process tools. Concerning this, our findings also reveal that the number of roles an individual takes up in an organization was up to 56 percent. Role theorists in a scarcity model in [26] posit that a solid commitment to one role may preclude attachment to other roles. This takes us down to the individual in the organization whose efficiency is curtailed with overlapping duties, which breed conflict of interest and confines one's capacity to follow procedures and processes, hence developing ad-hoc and shortcutting tendencies. This situation inculcates the volatility observed in SSCs [8],[30] which is occasioned by pressure and has led the SSCs to have no choice but to rather amour themselves with attitudes, since some process tools have increased overhead, meaning that the SSCs will not see the tool as being a viable solution or even worthy of trying to use. This creates a stereotypical tendency, for instance, the impression that it could be difficult to adopt tools and techniques for development. Although our data show considerable high numbers of respondents stating that they rarely found difficulty in adopting tools, this interestingly points to the fact that the limited tools used could reveal such findings. Limitation of usage of process tools is also cited by Laporte [19].

Additionally, individuals working under pressure while taking up multiple roles tend to ignore details that were done in the earlier role. For instance, a software tester may not have been as strict when he developed the artefact being tested, as we observe in the 'confidence' in products followed by an attitude of no testing. In some cases, the companies have a software testing strategy that may not be referred to.

The structure of an organization talks a lot to this, the management practice in an organization which is viewed through the lenses of the number of roles one takes in an organization. On what seems to look like organization flexibility by additional responsibility for the already available routine work during a typical project, and finally resource allocation, in which there is the liberty to switch resources from one activity to another. This tends to be seen as an unfortunate element to the SSCs, especially that they are already constrained by resources as also eluded to in [12],[22]. The concerns affect the ability of a team to follow procedural processes and look at multiple tasks in processes as backbreaking while opting for ad-hoc practices. Most tools in software engineering have multiple tasks and need time. This makes it difficult for the SSCs to take up these tools.

6 CONCLUSION LIMITATIONS AND FUTURE WORK

This work helps us understand how the organizational dynamics impedes the SSCs from utilizing process tools as a step towards streamlining software engineering practice in the SSCs and a revolution in software quality. The study points out key standpoints as far as organizations and utilization of process tools are concerned. The developers of process tools must take note of the organizational dynamics and make tools that are usable in the context of this type of organization. The findings pointed out in this study fit into the characterization and classification of SSCs, which will become a guide while choosing process tools that are appropriate for the specific organization.

This study is limited to a small sample of data and may not generalize the entire software development in SSCs. However, we have attempted to take care of the African context of software development, especially by triangulating the data in 3 countries spread all over Africa. Data collection is still ongoing in 3 other European countries, and this data will be used to validate our findings and aggregate on the other studies on how the business environment affects the utilization of software process tools in SSCs. Additionally, we are in the process of publishing the findings of a study in which we have developed the characteristics of SSCs, leading to the development of a classification taxonomy and ultimately an adaptability framework to help the SSCs choose process tools that are easily usable in their operational context.

ACKNOWLEDGMENTS

The authors would like to acknowledge Victoria, Sharon, Pulafela, Joseph and Jonathan for the follow-up calls to the respondents and Maria, Annika, Sami, Bilal, Shola, and Saltan for reviewing the survey protocol. We also appreciate the support of Denis, Kwaku, Joseph, Babatunde Nicole, Harriet and Daniel during the survey piloting phase and all the practitioners from the industry who voluntary participated in the survey.

REFERENCES

[1] Abukwaik, H. and Rombach, D. 2017. Software interoperability analysis in practice: a survey. Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering (2017), 12–20.

[2] Anacleto, A. *et al.* 2004. Experiences gained from applying ISO/IEC 15504 to small software companies in Brazil. 4th International SPICE Conference on Process Assessment and Improvement, Lisbon, Portugal (2004), 33–37.

[3] Andrade, C. 2021. The inconvenient truth about convenience and purposive samples. Indian Journal of Psychological Medicine. 43, 1 (2021), 86–88.

[4] Balikuddembe, J.K. and Tuape, M. 2017. An Ambiguity Minimization Technique during Requirements Elicitation Phase. 2017 International Conference on Computational Science and Computational Intelligence (CSCI) (2017), 945–950.

[5] Baltes, S. and Diehl, S. 2018. Towards a Theory of Software Development Expertise. Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (New York, NY, USA, 2018), 187–200.

[6] Baltes, S. and Ralph, P. 2020. Sampling in software engineering research: A critical review and guidelines. arXiv preprint arXiv:2002.07764. (2020).

[7] Barr, M. and Parkinson, J. 2019. Developing a work-based software engineering degree in collaboration with industry. Proceedings of the 1st UK & Ireland Computing Education Research Conference (2019), 1–7.

[8] Basri, S. *et al.* 2019. The Organisational Factors of Software Process Improvement in Small Software Industry: Comparative Study. Emerging Trends in Intelligent Computing and Informatics - Data Science, Intelligent Information Systems and Smart Computing, International Conference of Reliable Information and Communication Technology, {IRICT} 2019, Johor, Malaysia, 22-23 September, 20 (2019), 1132–1143.

[9] Brhel, M. *et al.* 2015. Exploring principles of user-centered agile software development: A literature review. Information and software technology. 61, (2015), 163–181.

[10] Cico, O. 2020. Towards transferring lean software startup practices in software engineering education. {ESEC/FSE} '20: 28th {ACM} Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Virtual Event, USA, November 8-13, 2020 (2020), 1686–1689.

[11] Cronbach, L.J. 1951. Coefficient alpha and the internal structure of tests. psychometrika. 16, 3 (1951), 297–334.

[12] Dangle, K.C. *et al.* 2005. Software Process Improvement in Small Organizations: {A} Case Study. {IEEE} Softw. 22, 6 (2005), 68–75. DOI:https://doi.org/10.1109/MS.2005.162.

[13] Garousi, V. *et al.* 2016. Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. Information and Software Technology. 79, (2016), 106–127.

[14] Garousi, V. and Zhi, J. 2013. A survey of software testing practices in Canada. JOURNAL OF SYSTEMS AND SOFTWARE. 86, 5 (May 2013), 1354–1376. DOI:https://doi.org/10.1016/j.jss.2012.12.051.

[15] Hardy, J. *et al.* 2019. Rural HCI research: Definitions, distinctions, methods, and opportunities. Proceedings of the ACM on Human-Computer Interaction. 3, CSCW (2019), 1–33.

[16] Javed, T. *et al.* 2004. A study to investigate the impact of requirements instability on software defects. ACM SIGSOFT Software Engineering Notes. 29, 3 (2004), 1–7.

[17] Kirbas, S. *et al.* 2014. The effect of evolutionary coupling on software defects: an industrial case study on a legacy system. Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (2014), 1–7.

[18] Klotins, E. *et al.* 2019. Software engineering in start-up companies: An analysis of 88 experience reports. Empirical Software Engineering. 24, 1 (2019), 68–102.

[19] Laporte, C.Y. and O'Connor, R. V 2016. A multi-case study analysis of software process improvement in very small companies using ISO/IEC 29110. European Conference on Software Process Improvement (2016), 30–44.

[20] Lee, J.-C. *et al.* 2016. Examining the impacts of organizational culture and top management support of knowledge sharing on the success of software process improvement. Computers in Human Behavior. 54, (2016), 462–474.

[21] Nunes, N.J. and e Cunha, J.F. 2000. Wisdom: {A} Software Engineering Method for Small Software Development Companies. {IEEE} Softw. 17, 5 (2000), 113–119. DOI:https://doi.org/10.1109/52.877877.

[22] O'Connor, R. V *et al.* 2010. Exploring managerial commitment towards SPI in small and very small enterprises. European Conference on Software Process Improvement (2010), 268–279.

[23] O'Connor, R. V and Coleman, G. 2009. Ignoring'best practice': Why Irish software SMEs are rejecting CMMI and ISO 9000. (2009).

[24] Parry, S. *et al.* 2012. Marketing for survival: a comparative case study of SME software firms. Journal of Small Business and Enterprise Development. 19, 4 (Jan. 2012), 712–728. DOI:https://doi.org/10.1108/14626001211277488.

[25] Ralph, P. *et al.* 2020. ACM SIGSOFT empirical standards. (2020).

[26] Randall, D.M. 1988. Multiple Roles and Organizational Commitment. Journal of Organizational Behavior. 9, 4 (Oct. 1988), 309–317.

[27] Ruokonen, M. and Saarenketo, S. 2009. The strategic orientations of rapidly internationalizing software companies. European Business Review. 21, 1 (Jan. 2009), 17–41. DOI:https://doi.org/10.1108/09555340910925166.

[28] Saiedian, H. and Carr, N. 1997. Characterizing a software process maturity model for small organizations. ACM SIGICE Bulletin. 23, 1 (1997), 2–11.

[29] Tripathi, N. *et al.* 2016. Exploring Processes in Small Software Companies: {A} Systematic Review. Software Process Improvement and Capability Determination - 16th International Conference, {SPICE} 2016, Dublin, Ireland, June 9-10, 2016, Proceedings (2016), 150–165.

[30] Tuape, M. *et al.* 2020. Does Context Matter? Assessing the Current State of Quality Practice During Software Development in Small Software Companies. Proceedings of the Future Technologies Conference (2020), 341–356.

[31] Tuape, M. *et al.* 2021. Software Engineering in Small Software Companies: Consolidating and Integrating Empirical Literature into a Process Tool Adoption Framework. IEEE Access. (2021).

[32] Tuape, M. and Ayalew, Y. 2019. A roadmap for a comparison framework for an adaptable software process improvement framework in small software companies. Annals of Computer Science and Information Systems. 20, (2019), 133–141.

[33] Tuape, M. and Ayalew, Y. 2019. Factors affecting development process in small software companies. 2019 IEEE/ACM Symposium on Software Engineering in Africa (SEiA) (2019), 16–23.