# Software Practice in Small Software Companies: Development Context Constraints on Process Adoption.

MICHEAL, TUAPE*
Department of software engineering, Lappeenranta-Lahti University of Technology, Finland
micheal.tupe@lut.fi

VICTORIA, Hasheela-Mufeti
Dept of Computing, Maths and Statistical Sciences University of Namibia, Windhoek, Namibia
vhasheela@unam.na

PETRUS, T, Iiyambo
Dept of Computing, Maths and Statistical Sciences University of Namibia, Windhoek, Namibia
piiyambe@unam.na

ANNA, Kayanda
Dept of Information Systems, College of Business Education, Dar es Salaam, Tanzania
anna.kayanda@cbe.tz

SHARON, D, Mensah
B-Systems Limited, Accra, Ghana
sharon.mensah@bsystemslimited.com

JUSSI, Kasurinen
Department of software engineering, Lappeenranta-Lahti University of Technology, Finland
jussi.kasurinen@lut.fi

## ABSTRACT

Small software companies (SSCs) interact with the immediate environment, exposing them to challenges that force the organization to undertake adjustments if it must survive and remain in business. These adjustments result into counterproductive practices and changes that create complexities in process adoption. This cross-sectional survey investigates the occurrences around the customer in the development context that affect the adoption of process in SSCs. To answer the research questions, we conducted a survey on 115 respondents and found out that although customer engagement has a significant relationship with reducing rework, inadequacies in the engagement due to the customer's lack of knowledge of software processes, triggers unstructured and ad-hoc methods in SSCs. The main contribution of this paper is a customer engagement framework that seeks to transform software processes by focusing on the customer as a pillar of achieving purpose and value to reduce development effort and time.

## CCS CONCEPTS

• **Software and its engineering**; • **Software creation and management**; • **Software development process management**;

## KEYWORDS

Software Development Context, Software processes, Small Software Companies

**ACM Reference Format:**
MICHEAL, TUAPE, VICTORIA, Hasheela-Mufeti, PETRUS, T, Iiyambo, ANNA, Kayanda, SHARON, D, Mensah, and JUSSI, Kasurinen. 2022. Software Practice in Small Software Companies: Development Context Constraints on Process Adoption.. In *2022 The 3rd European Symposium on Software Engineering (ESSE 2022), October 27–29, 2022, Rome, Italy.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3571697.3571698

*Place the footnote text for the author (if applicable) here.



*ESSE 2022, October 27–29, 2022, Rome, Italy*

ACM ISBN 978-1-4503-9730-8/22/10.
https://doi.org/10.1145/3571697.3571698

## 1 INTRODUCTION

For small software companies (SSCs) to leverage the usefulness of software process, significant attention needs to be given to the development context in which companies operate. The SSCs are defined as companies with less than 50 employees engaged in software development [1]. These companies suffer numerous challenges, including low-quality products and limited satisfaction of clients. A report by Jrgensen [2] suggests that about 75% of software projects fail, 70% of which can be attributed to insufficient practices, Researchers have also attributed lots of cases of failure to the SSCs yet they dominate the software industry by producing over 80% of software products on the market. Different processes have been developed to support practice in software development; unfortunately, the SSCs find these tools unusable [3]. Anacleto et al. [4] prove that only 7% of software companies use tools. Additionally, O'Connor and Laporte [5] cite the lack of quality standards adoption as a result of the perception by SSCs that such standards have been developed for large companies.

In defining the development context, we explore involving all aspects external to the company with significant influence on the functionality of the company while considering what should encompass the development context, aspects like the customers, competitors, suppliers, government, and the social, cultural, political, technological and legal regime have to be considered [9]. These aspects differ from company to company, given that the internal dynamics in a company transform the effects of these aspects, creating constant changes that require constant adjustments. This explains why SSCs continue to be volatile and unpredictable, as suggested by Lee and Chen [10].

The challenges within the development context are responsible for the limited productivity, and ineffectiveness of the tools used during development. Virtanen et al. [6] argue that to mitigate the challenges, SSCs require adjustment of processes to match the organizational context. The authors further observe that since organizations and their contexts differ; thus, the solution most suitable for individual company needs must be modified or localized to fit the case-specific or native contextual demands. However, this is

unfortunate to the SSCs that adjustments and modifications are often a daunting task because of the uniqueness exhibited in these companies [7].

This study is motivated by the findings from the study of Tuape et al., [11], who highlight the usefulness of process adoption for SSCs. In their work, they propose a theoretical framework for process adoption and cite the development context as an influencing factor. The authors, however, call for an investigation of the critical elements in the development context which need attention, and hence the need for SSCs to appropriately adjust internal dynamics as they try to grasp the available opportunities to face the threats that emerge from the development context. Knowledge transfer processes during software engineering are captured from the development context to identify specifics regarding the problem domain, the business functions, and the inherent characteristics of these functions that are transferred through customer engagement.

This study aims at investigating the challenges with the development context and how they affect the utilization of the software tools in while focusing on the customer and how the customer's interaction affects the utilization of the processes by answering the following research questions: (i) What is the software practitioner's perception of the facets of customer interaction within the software development context? We considered the customer's interest in specific development methods. If customer engagement is considered satisfactory during projects, whether customers have no problem paying extra for quality, and whether customers attended validation meetings with developers, the rationale for considering the customers and these facets is that the critical activities in software engineering require a lot of customer involvement. These activities include Requirements engineering [12] and software testing [13]. This is backed by evidence from a recent study by Kuhrmann et al. [14] that highlighted the fact that improvements in stakeholders' collaboration have contributed to making agile a transformation in the software industry, perhaps because of the seen benefits of stakeholder interaction. (ii) What correlation exit between the aspects of customer interaction and rework? The rationale for rework as a measure of effectiveness in software development is that rework affects time, and time is a crucial determinant of several factors, including measuring performance [15], a view also emphasized in [16]. The interaction with the customer as an essential facet of the development context with a measure of rework is vital for software development because it helps us associate a metric to the development context, which is a practical challenge in practice as shared by other researchers in [17].

This work is part of a larger study in which we are studying software practice in SSCs to understand why SSCs minimally utilize processes. So far, we have conducted a Systematic Literature Review (SLR) and consolidated our findings into a process tools theoretical framework [11]. We have also reported on our exploration of the usage of process tools and how organizational dynamics impede process tool utilization. And in this paper, we seek to identify the challenges in the development context that could solve the impasse in software engineering, especially for SSCs.

The remaining sections of this paper, we discuss related work in Section 2, and the methodology, survey results and discussion of the results are presented in Sections 3, 4 and 5. The paper then concludes with limitations of the study, prospects for future works and recommendations to industry and researchers.

## 2 RELATED WORK

The development context in which software is developed has been a concern for several researchers, discussing the impediments of software development generally and specifically to SSCs. Some of the interesting discussions on the development context provoke interest for further discussion, especially at an era where expectations are high from the software industry amidst frustration of rampant failure. In this section, we examine the discussion in three folds: First we take an insight into the development context deliberated generally to software development, secondly we discuss the development context in relation to software process improvement (SPI) and how the proponents view the development context as an impediment to improving software development in SSCs, and finally we highlight the initial arguments put in place for process tools adoption and how it is affected by the development context.

The proponents of SPI [18] argue that SPI offers tools and methods that are useful in standardizing the development practices so that it is easier to assess the current and future state of projects.

Virtanen et al. [6] assert that for today's software business and its productivity, SPI plays a significant role, the authors highlight challenges faced by software producing organizations with the productivity and effectiveness of their operation being key among others. Critical success factors are defined to make the successful improvement procedures more certain while listing numerous methods to make the operation better. The authors further note the importance of the methodologies to be adjusted to match the organizational context, stating that all organizations and their environments are different, and thus the most appropriate solution for individual organizational needs must be modified to fit the case-specific contextual demands, a view also shared by other researchers in [8]. Additionally, the critics of SPI like Kuhrmann, and Münch [19] point out cases where SPI has not been successful and among the issues they raise is "immune reaction", which closely relate to the development context in which software is developed. Similar issues of immune reaction is also raised by other researchers in [20].

Development context and how it affects software development in SSCs has been an issue of discussion by different researchers. Among other issues, Tuape and Ayalew [9] cite the development context which they refer to as development context in which software is developed with focus on the customers level of involvement and customers own knowledge about software engineering process. In another study, Tuape et al. [11] discuss the challenges process tools adoption by SSCs. The authors propose a theoretical framework and posit that process tools have a weak relationship to the production of quality software. They further add that the weak points can be enhanced by taking the contextual factors into consideration, among which are the development context in which they propose an adoption mechanism and among other factors make recommendations that what they call development context as significant and should be considered to create and process tools that are usable in the context of SSCs and support the adoption of the process tools.

Empirical studies dealing exclusively with development context is lacking and most of the studies available rather cite the challenge as a block and do not define the causes or rather what could be the underlying factors in the development context. Moreover, we have not come across any literature attempting to relate the limited utilization of process tools in SSCs to the development context.

## 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 [21], [22], to understand the development context in the SSCs.

### 3.1 Research questions.

The following sub-section will elaborate more on the research questions.

*3.1.1 Research question 1.* To investigate the perspective of the developer on the customer and how they interaction within the company, 4 specific areas of interaction were examined. The areas on interest included (i) whether the customers want specific methods for development, (ii) whether customer engagement is satisfactory during projects, (ii) whether their customers have a problem paying extra for quality, and (iv) examines the clients' participation in validation of requirements to represent the variables V1, V2, V3 and V4. representing customer's choice of specific methods, the extent of customer engagement, customer's ease of paying for quality and customers attendance of validation meeting respectively.

RQ 1.1 Do the customers of the SSCs want specific methods for their projects? RQ 1.2 Is customer engagement during a project of the SSCs satisfactory? RQ 1.3 Do the customers of the SSCs find it easy to pay for quality? RQ 1.4 Do the customers of the SSCs attend validation meetings?

*3.1.2 Research question 2.* The main research question was: What correlation exit between the facets of customer interaction during software development and rework? Then it was followed by four sub-questions. Investigating the existence of correlations between the facets of customer involvement in software development (independent variables) that is; specific methods in projects (IV1); engagement during projects being satisfactory (IV2); ease to pay for quality (IV3), and whether attending validation meetings (IV4) have an influence on rework (dependent variable DV) as follows:

RQ 2.1 Does the customer's preference for specific methods in projects influence rework during software development? RQ 2.2 Does the extent of customer engagement during software development projects influence rework? RQ 2.3 Does the customers' ability to pay for quality easily influence redoing of work to come up with an acceptable product? RQ 2.4 Does the attendance of validation meetings by the customer influence rework during software development?

### 3.2 Sampling

A sample was drawn from a population of software-intensive companies of different sizes developing software products for a wide variety of markets. The survey was sent out to 84, 95 and 103 for Namibia, Tanzania and Ghana respectively totaling to 282 companies. A total of 115 (one respondent per company) data points was returned 30, 38 and 47, from the respective countries representing 26, 33, and 41%. We employed the purposive (heterogenous) sampling also known as maximum variation sampling, based on a criterion whose characteristics are defined for a purpose that is relevant to the study [23]. Our main purpose of selection of this sampling method was to ensure that our sample would be representative of the varying sizes of SSCs. The criterion of selecting the company was the number of persons in the company to fit the definition of SSCs (under 50 persons), the type of software-intensive products from the company, and the role in the software company. Purposive sampling has been used in software engineering studies as advised by Baltes and Ralph [24] and used in similar studies [25], [26].

### 3.3 Characteristics of the respondents

Of the (n=115) are 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. On education level is such that Bachelors, Masters, and Doctorate degrees were 83, 24 and 4, and those with high school and vocational training were 3 and 1. On gender, 76 and 24% in favor of the males and for number of personnel employed, those under 5 were the majority with 40.9% followed by 21 and 25, 26 and 30 with 11 and 14% respectively. The rest employed between 6 and 10, 11 and 16, 31 and 50, 16 and 10 had 7, 7, 4.3 and 1% respectively.

### 3.4 Questionnaire design

The survey questionnaire was designed to investigate software practices (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 [27]. We engaged fifteen practitioners from industry in a pilot study to check the language used in the questionnaire for familiarity to the participants. Some questions were dedicated to the profiles and demographics of the participants. The questions probed data related to development context, with answers of type 5-point Likert scale giving participants options of 1. Never, 2. Rarely, 3. Sometimes, 4. Often, and 5. Always.

### 3.5 Data collection and analysis

Telephone calls were made, to request participation in the study before mailing the questionnaire link on the Webropol survey system. Participant time on a questionnaire was under 30 minutes. The study represents a 40.8% response rate, which presents slightly above the 40% recommended in 28. The data analyses were conducted using Statistical Package for the Social Sciences version 26 to obtain descriptive statistics, cross-tabulation, and spearman's rho correlation ($\rho$) between the outcome variable and the predictor variables.

**Table 1: Frequencies, percentages, and cumulative percentages of the respective responses for the variables V1, V2, V3 and V4 (n= 115).**

| | | Frequency | Percent | Cumulative Percent |
|---|---|---|---|---|
| V1. | Never | 21 | 18.3 | 18.3 |
| | Rarely | 36 | 31.3 | 49.6 |
| | Sometimes | 36 | 31.3 | 80.9 |
| | Often | 19 | 16.5 | 97.4 |
| | Always | 3 | 2.6 | 100.0 |
| V2 | Rarely | 4 | 3.5 | 3.5 |
| | Sometimes | 28 | 24.3 | 27.8 |
| | Often | 53 | 46.1 | 73.9 |
| | Always | 30 | 26.1 | 100.0 |
| V3 | Never | 6 | 5.2 | 5.2 |
| | Rarely | 22 | 19.1 | 24.3 |
| | Sometimes | 35 | 30.4 | 54.8 |
| | Often | 25 | 21.7 | 76.5 |
| | Always | 27 | 23.5 | 100.0 |
| V4 | Never | 9 | 7.8 | 7.8 |
| | Rarely | 12 | 10.4 | 18.2 |
| | Sometimes | 27 | 23.5 | 41.7 |
| | Often | 24 | 20.9 | 62.6 |
| | Always | 43 | 37.4 | 100.0 |
| | Total | 115 | 100.0 | |

## 4 RESULTS

In this section, we present the results of our study answering the two main research questions and the subsidiary questions that are associated with each of them. Firstly, RQ 1, we present the frequency distribution of variables V1 through V4 based on the sample (n=115) concentrated on the four points of interaction with customer in the software development context as presented in Table 1. In RQ 2, we present the relationship of the points of interaction with customer and how they affect rework during software development in Tables 2.

### 4.1 RQ 1

The frequency distribution of the responses to the 4 sub-research question under RQ 1. On the question investigating whether the customers want specific methods for development, 31.3% said they either rarely or sometimes want specific methods for development, whereas only 2.6% responded that they always want such methods. Regarding the question asked to investigate whether customer engagement is satisfactory during projects, 46.1% responded that they were often satisfied, while 26.1% said they were always satisfied with customer engagement. On whether the customers have no problem paying extra for quality, majority (75.6%) of the respondents generally found it easy to pay extra for quality. Finally, nearly 60% of the respondents said that customers attend validation meetings with developers.

### 4.2 RQ 2

We only found one significant correlation that existed after testing the 4 sub-questions for RQ 2. We measured the strength and magnitude of association between variables and found the result shows that there is a significant and NEGATIVE correlation between the extent of customer engagement during project is satisfactory and having influence on rework.

The correlation between the rest of the pairs of variables were not significant. In this subsection we only present the results of comparing engagement during projects being satisfactory and having an influence on rework that were found to be significant. We cross tabulated the results of variables IV and V2, and also presented the results of the Chi square test of association for assessing the goodness of fit between variables IV and V2, in Table 3. the data presents the directional measures of the association between variables IV and V2, the data presents the symmetric measures of the association between variables IV and V2 and finally the results of Spearman rank correlation between variables.

The results of the cross tabulation presented in the data shows that the respondents who think that redoing the work will never produce acceptable products are certain (100% probability or likelihood) that customer engagement is often satisfactory during projects. In addition, those who think that redoing the work will always produce acceptable products are 4.3 times (28.6%/6.7%) more likely to think that customer engagement is sometimes satisfactory during projects.

Additionally, customer engagement satisfaction during projects and rework to come up with an acceptable product are dependent. Alternatively, this means that there is a significant association between these two variables. Therefore, ranking customer engagement satisfaction during projects as a function of redoing of work to come up with an acceptable product differ significantly. In simple terms, the ranking of one variable determines the ranking of the other.

**Table 2: A cross tabulation between variables V2 representing engagement during projects being satisfactory and IV representing having an influence on rework against the participants options of Never, Rarely, Sometimes, Often, and Always presented as 1,2,3,4 and 5.(n = 115)**

| Crosstab | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | V2. | | | | Total |
| | | | 2 | 3 | 4 | 5 | |
| IV | 1 | Count | 0 | 0 | 4 | 0 | 4 |
| | | % Within IV | 0.0% | 0.0% | 100.0% | 0.0% | 100.0% |
| | | % Within V2 | 0.0% | 0.0% | 7.5% | 0.0% | 3.5% |
| | | % of Total | 0.0% | 0.0% | 3.5% | 0.0% | 3.5% |
| | 2 | Count | 3 | 3 | 7 | 11 | 24 |
| | | % Within IV | 12.5% | 12.5% | 29.2% | 45.8% | 100.0% |
| | | % Within V2 | 75.0% | 10.7% | 13.2% | 36.7% | 20.9% |
| | | % of Total | 2.6% | 2.6% | 6.1% | 9.6% | 20.9% |
| | 3 | Count | 0 | 6 | 20 | 14 | 40 |
| | | % within IV | 0.0% | 15.0% | 50.0% | 35.0% | 100.0% |
| | | % within V2 | 0.0% | 21.4% | 37.7% | 46.7% | 34.8% |
| | | % Of Total | 0.0% | 5.2% | 17.4% | 12.2% | 34.8% |
| | 4 | Count | 1 | 11 | 16 | 3 | 31 |
| | | % Within IV | 3.2% | 35.5% | 51.6% | 9.7% | 100.0% |
| | | % within V2 | 25.0% | 39.3% | 30.2% | 10.0% | 27.0% |
| | | % of Total | 0.9% | 9.6% | 13.9% | 2.6% | 27.0% |
| | 5 | Count | 0 | 8 | 6 | 2 | 16 |
| | | % within IV | 0.0% | 50.0% | 37.5% | 12.5% | 100.0% |
| | | % within V2 | 0.0% | 28.6% | 11.3% | 6.7% | 13.9% |
| | | % of Total | 0.0% | 7.0% | 5.2% | 1.7% | 13.9% |
| Total | | Count | 4 | 28 | 53 | 30 | 115 |
| | | % within IV | 3.5% | 24.3% | 46.1% | 26.1% | 100.0% |
| | | % within V2 | 100.0% | 100.0% | 100.0% | 100.0% | 100.0% |
| | | % of Total | 3.5% | 24.3% | 46.1% | 26.1% | 100.0% |

**Table 3: Chi square test of association for assessing the goodness of fit between the variables of engagement during projects being satisfactory and having an influence on re-work (n = 115)**

| Chi-Square Test | Value | Degrees of freedom | Asymp. Sig. (2-sided) |
|---|---|---|---|
| Pearson Chi-Square | 32.194a | 12 | 0.001 |
| Likelihood Ratio | 33.448 | 12 | 0.001 |
| Linear-by-Linear Association | 6.120 | 1 | 0.013 |

These results shows that rework to come up with an acceptable product, is considered as response (dependent or outcome) variable whereas customer engagement satisfaction during projects is considered as explanatory (independent / predictor) variable, then the relationship or association is negative, with a strength or magnitude is 0.260. However, when customer engagement satisfaction during projects is considered as a (dependent or outcome) variable whereas rework to come up with an acceptable product is considered as explanatory (independent / predictor) variable, then the relationship or association is negative, with a strength or magnitude is 0.231.

Finally, if the two permutations/scenarios stated above are disregarded (irrespective of which variable is dependent and which is independent), then the association is negative, and the strength/magnitude is 0.245. Since for all three cases the association is positive, case 2 (Somers'd value = -0.260) is best among the three. We also identified three types of symmetric measures that shows the strength and direction of the association between redoing the work to come up with an acceptable product and customer engagement satisfaction during projects without considering which variable is a dependent and which is an independent. Among the three measure, Gamma value = -0.342 gives us the strongest but negative association between the variables. There is a negative correlation coefficient between customer engagement during software development and rework to come up with accepted product of 0.280 with significant level of 0.01 meaning there is a 99.9% certainty of the relationship as shown in Table 4.

**Table 4: The correlation between variable (V2) *representing engagement during projects being satisfactory* and (IV) representing *having an influence on rework.* using the Spearman's rho rank correlation coefficient. (n = 115)**

| Correlations | | | | |
|---|---|---|---|---|
| | | | IV | V2 |
| $\rho$ | IV | Correlation Coefficient | 1.000 | -0.280** |
| | | Sig. (2-tailed) | . | 0.002 |
| | V2 | Correlation Coefficient | -0280** | 1.000 |
| | | Sig. (2-tailed) | 0.002 | . |

**. Correlation is significant at the 0.01 level (2-tailed).

## 5 DISCUSSIONS

In this section, we discuss the results presented in Section 4. This will be compared with findings from related studies in software engineering in general and some specific cases of SSCs to emphasize the main highlights of this study.

### 5.1 Research question 1

Research question 1 examined the perceived customer actions as part of the development context in determining customers preference of specific tools, perceived customer engagement if considered satisfactory, customers' ability to pay extra for quality and whether customers had a routine of getting involved in validation of requirements.

The results obtained from the study indicates that about 19% of the respondents thought that customers are specific about software development methods, which means that the other 81% of the customers are not really concerned about or do not understand the methods and tools used for developing software in their own projects as highlighted in [29],[39] . The use of tools is very important to enhance efficiency of the team and for quality products to be achieved during development as emphasized in [31],[32]. Other evidence in literature [33] indicate that the non-utilization of tools has led to poor quality software products and failed projects. Additionally, Anacleto [4] relates the low adoption of tools to lack of knowledge by the customer who they believe needs to have sufficient knowledge on what tools can be ideal for their projects. This implies that some of the artefacts during software processes may be difficult to understand, a customer who has interacted with a basic process flow diagram may prefer this to a UML use case diagram because understanding the modeled system under construction is better with a process flow diagram.

On whether perceived customer engagement was considered satisfactory to the respondents, 72.2% answered to the affirmative which is consistent with views of Nunes and Cunha [34] that SSCs are flexible which is a leverage to customer engagement and because of this the SSCs enjoy advantages of easy communication internally and externally. The challenge is however that this advantage is not observed in effective process and general efficiency. The question that remains unresolved may require a deeper insight, regarding what makes communication is SSCs ineffective. A 72% perception of customer engagement would mean that the developers believe in engagement and software development is expected to be much more successful especially if customer engagement is considerably good as cited by Balikuddembe and Tuape [12]. The challenge here is that the engagement could most likely be informal, and this leaves a gap in the expected efficiency of the engagement and communication in delivering quality products. This closely identifies with the finding of other researchers who discuss how SSCs ignore best practices like O'Connor and Coleman [35]. The tools are a medium of communication especially for formal engagement. However, this does not mean informal engagement is not beneficial to the companies, it has benefits, although they may be minimal. This also connects to the first question in which we discussed customers' demand for specific tools, leaving us with an option of suggesting that the two situations interrelate.

Additionally, engagement avails opportunity for a customer to present their expectations for the product, who can define its technological limitations better than the developer? one should not ignore the view of the programmers although it is believed that a considerable proportion of developers are introverts [36], [37]. Unforeseen eventualities can be identified at this point, and this may save time and effort especially when captured at the pre of early stage of design. Practices that encourage engagement with the customer facilitate easy transformation to agile methods because agile methods tend to be successful with effective customer engagement. To further underscore the importance of customer engagement Bosch [38], suggests that for companies to transition from working with planned releases with detailed requirement specifications to continuously experimenting with customers for example, engagement needs to be heightened to make it possible to optimize previously implemented features.

Customers having no problem paying for quality seem to attract slightly above 50%, which is considerably low and was expected to be higher. Because quality is known to be synonymous with cost, this means the appetite for quality is not as high as expected from the customer. However, it is important to note that evidence from literature [9], [39] indicates lack of knowledge on quality, and this could affect the appetite for quality as well as the tools for quality. After all who would want to pay extra for what he or she would not understand or want? On another note, this could also be a result of going basic because of the stereotypical tendencies of thinking that quality is expensive, a view shared by Rivas et al., [40].

The validation meetings are relatively attended by customers, about 60% means a fair number of customers are somehow involved in the processes of evaluation of software through or at the end of the development process to determine whether it satisfies the specified requirements. This is a critical activity and perhaps among the most important in the development context. Validation takes

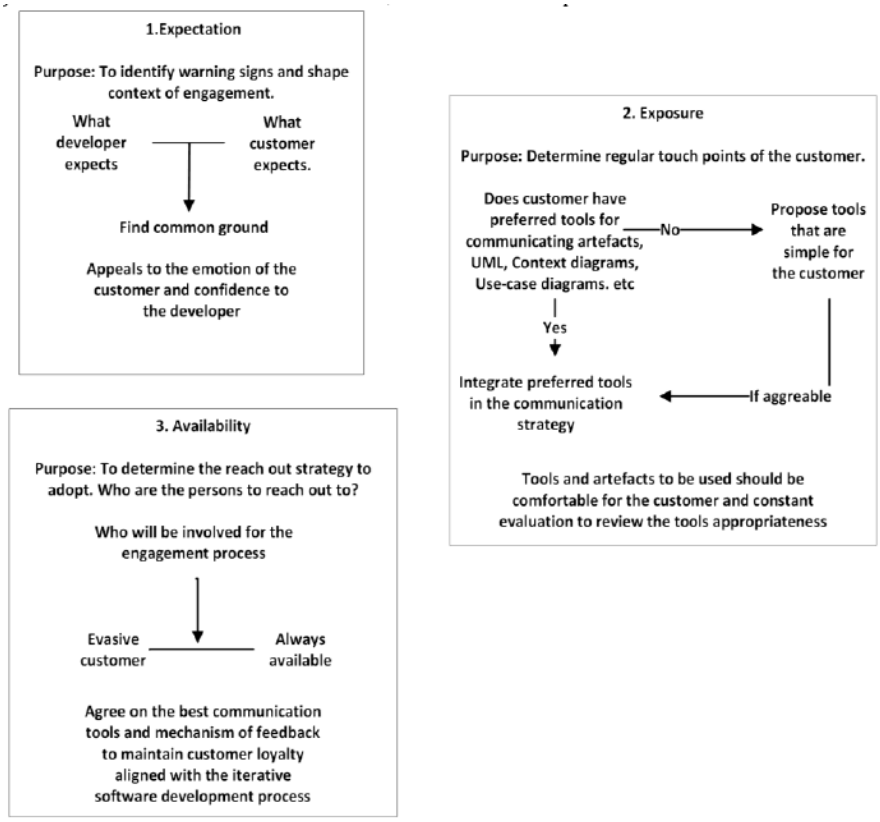


**Figure 1: Customer engagement framework to guide the alignment of communication tools and the development of an engagement strategy during software development and to facilitate customer engagement during agile transformation for SSCs**

care of the interests of the customer all through the development process, and constant feedback shapes the project into the correct trajectory. Although the figure of the participants who indicated that their customers always attend validation meetings is higher and looks encouraging, this whole activity is dependent on the customer, and maybe the question should be, where are the 40% who do not attend the validation meetings?

## 5.2 Research question 2:

What correlation exits between the facets of customer interaction during software development and a measure of rework? The significant and negative correlation between satisfactory customer engagement and rework to come up with a satisfactory product means an increase in customer engagement during software development reduces the chances of rework to come up with satisfactory product and vice-versa. This finding is important although it may not necessarily reflect what exactly happens during software development in SSCs this is probably because of the ad hoc and informal practices of the SSCs. Although the SSCs are minimally benefiting from informal communication, particularly verbal communication, and tacit knowledge at the expense of formal communication, there is a need for formal communication, which entails detailed documentation to tap into the advantages that come with effective communication in the development context. Researchers Laporte et al., 41 associate significant reduction of rework during software development to the use of frameworks as a basis for formalities. This finding is a vote of confidence on process tools that broker communication between the customer and the development team alienating the company from informal communication. It is a good thing that the developers have a good attitude towards communication, however it is important to ensure that the communication is

effective. Informal communication has a high rebound effect on software development that creates an impression of a smooth flow, yet all remain assumptions and not committal. Examples of this can be seen from requirements engineering done in an informal manner, then it ultimately ends up with conflict between the development team and the customer, a view also highlighted by Menezes et al., [42]. Another case in point is software testing, where the customers involvement is informal giving potential challenges at the end of the project. Informal communication is a recipe for evasion of software development tools which include the processes, frameworks, and methods, used in the development of the software in question. This breeds shortcuts as an alternative to quickly manage resources, a characteristic of adopting to ad-hoc processes and methods with the customer who is a key embodiment of the development context at the center of software development.

### 5.3 Customer Engagement framework

Figure. 1 presents the proposed framework that SSCs can use in developing communication with a customer as expressed by Korkala et al., [43] who emphasize the importance of effective communication and associate the failures in software practice to haphazard communication engagement strategy that will lead the SSCs to the realization of customer engagement while ensuring a customer-focused development process that motivates the customer to remain proactive. Although, the customer gets emotionally attached to the software product through the development process, an essential element for acceptance of software product.

Additionally, Sashi [44] observes that customer engagement is synonymous with commitment and satisfaction, critical issues in agile and traditional software development. The advantages of engagement are attained by ensuring that the adequate knowledge transfer processes from the development context to the developing team during software engineering can provide an effective software development process. Good customer engagement leads to appreciating the difference between a good software product and a great one

## 6 CONCLUSION, LIMITATIONS AND FUTURE WORK.

This study investigated the challenges in the development context and how they affect process tools in SSCs. We focused on the customer as viewed by the software practitioners given that software engineering is centred around people making the customer a vital element of the development context and came out with the following conclusions:

The limited adoption of processes in SSCs is connected to the customer's behavior and is a predicate to ad-hoc and unstructured practices that affect the product and development process. Although we found out that the communication seems to be sufficient, it does not give positive results in leveraging development time, cost, and customer satisfaction. This leads to ad-hoc practice, which threatens the utilization of process tools.

Not using tools results in ad-hoc, unstructured and chaotic practices occasioned by informal communication straining project resources like time and development effort, leading to rework and unreasonable delay of project deliverables at the cost of the customer. This also adversely affected the software development team, causing fatigue, lack of fulfilment, and loss of morale in the SSCs. To leverage the advantages of effective customer engagement, which include a better understanding of customer needs in the development process. Alignment of appropriate methods to attain the purpose and value of software development, we propose a customer engagement framework which would facilitate SSCs in 3 ways: (1) Increase purpose and value of software products by increasing utilization of tools, reducing ad-hoc, chaotic and unstructured practices while leveraging customers' emotions on the software product attained with the customer's involvement in the software engineering process. (2) A streamlined customer engagement process reduces software development time and effort that effectively transfers knowledge about domains, functions, and inherent characteristics to software specifications, thereby saving time otherwise spent understanding requirements. And (3) Facilitates the SSCs to a quick and easier transition to Agile software development methods to leverage the benefits of Agile software practice and improve general software development practice.

The interpretation of our findings is limited by the closed-ended questions used in this survey and a small sample size. A detailed qualitative study with open-ended response data needs to be conducted to complement this study. This study is also limited because it looks at the development context through the lenses of the software developer, further investigations on the customers perspective needs to be undertaken to supplement the findings from this study. Our next step is to undertake detailed qualitative studies and develop a classification taxonomy for the SSCs based on an exhaustive understanding of the characteristics of the SSCs, upon which we will construct an adoption framework to help the companies adapt the processes during practice.

## ACKNOWLEDGMENTS

We would like to acknowledge the assistance rendered by those who helped make follow-up calls to the respondents; we also appreciate the experts for reviewing the survey protocol, the individuals who supported us during the survey piloting phase, and all the practitioners from the industry who voluntary participated in the survey.

## REFERENCES

[1] N. Tripathi, E. Annanperä, M. Oivo, and K. Liukkunen, "Exploring Processes in Small Software Companies: A Systematic Review BT - Software Process Improvement and Capability Determination," 2016, pp. 150–165.

[2] M. Jørgensen, "Failure factors of small software projects at a global outsourcing marketplace," *J. Syst. Softw.*, vol. 92, pp. 157–169, 2014.

[3] M. Tuape, V. Hasheela-Mufeti, P. Iiyambo, A. Kayanda, and J. Kasurinen, "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)*, 2021, pp. 35–40, doi: 10.1145/3512716.3512722.

[4] A. Anacleto, C. G. von Wangenheim, C. F. Salviano, and R. Savi, "Experiences gained from applying ISO/IEC 15504 to small software companies in Brazil," in *4th International SPICE Conference on Process Assessment and Improvement, Lisbon, Portugal*, 2004, pp. 33–37.

[5] R. V. O'Connor and C. Y. Laporte, "The evolution of the ISO/IEC 29110 set of standards and guides," *Int. J. Inf. Technol. Syst. Approach*, vol. 10, no. 1, pp. 1–21, 2017, doi: 10.4018/IJITSA.2017010101.

[6] P. Virtanen, S. Pekkola, and T. Päivärinta, "Why SPI Initiative Failed: Contextual Factors and Changing Software Development Environment," in *2013 46th Hawaii International Conference on System Sciences*, 2013, pp. 4606–4615, doi: 10.1109/HICSS.2013.609.

[7] L. Rivas, M. A. Pérez, L. E. Mendoza, and A. Grimán, "Towards a Selection Model for Software Engineering Tools in Small and Medium Enterprises (SMEs)," in *Proceedings of the Third International Conference on Software Engineering Advances, {ICSEA} 2008, October 26-31, 2008, Sliema, Malta*, 2008, pp. 264–269, doi: 10.1109/ICSEA.2008.51.

[8] M. Tuape, P. Ntebane, and P. Majoo, "Does Context Matter? Assessing the Current State of Quality Practice During Software Development in Small Software Companies," in *Proceedings of the Future Technologies Conference*, 2020, pp. 341–356.

[9] M. Tuape and Y. Ayalew, "Factors Affecting Development Process in Small Software Companies," *Proc. - 2019 IEEE/ACM Symp. Softw. Eng. Africa, SEiA 2019*, pp. 16–23, 2019, doi: 10.1109/SEiA.2019.00011.

[10] J.-C. Lee and C.-Y. Chen, "Exploring the determinants of software process improvement success: A dynamic capability view," *Inf. Dev.*, vol. 35, no. 1, pp. 6–20, 2019.

[11] M. Tuape, V. Hasheela-Mufeti, A. Kayanda, J. Porras, and J. Kasurinen, "Software Engineering in Small Software Companies: Consolidating and Integrating Empirical Literature into a Process Tool Adoption Framework," *IEEE Access*, 2021.

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

[13] V. Kettunen, J. Kasurinen, O. Taipale, and K. Smolander, "A Study on Agility and Testing Processes in Software Organizations," in *Proceedings of the 19th*

*International Symposium on Software Testing and Analysis*, 2010, pp. 231–240, doi: 10.1145/1831708.1831737.

[14] M. Kuhrmann *et al.*, "What Makes Agile Software Development Agile," *IEEE Trans. Softw. Eng.*, 2021.

[15] J. D. Blackburn, G. D. Scudder, and L. N. Van Wassenhove, "Improving speed and productivity of software development: a global survey of software developers," *IEEE Trans. Softw. Eng.*, vol. 22, no. 12, pp. 875–885, 1996.

[16] P. Castañeda and D. Mauricio, "New factors affecting productivity of the software factory," *Int. J. Inf. Technol. Syst. Approach*, vol. 13, no. 1, pp. 1–26, 2020.

[17] S. Datta, "Agility measurement index: a metric for the crossroads of software development methodologies," in *Proceedings of the 44th annual Southeast regional conference*, 2006, pp. 271–273.

[18] S. Sanchez-Gordon, M. Sánchez-Gordón, M. Yilmaz, and R. V O'Connor, "Integration of accessibility design patterns with the software implementation process of ISO/IEC 29110," *J. Softw. Evol. Process*, vol. 31, no. 1, p. e1987, 2019.

[19] M. Kuhrmann and J. Münch, "SPI is Dead, Isn't It? Clear the Stage for Continuous Learning!," in *Proceedings of the International Conference on Software and System Processes*, 2019, pp. 9–13, doi: 10.1109/ICSSP.2019.00012.

[20] T. Hall, A. Rainer, and N. Baddoo, "Implementing software process improvement: an empirical study," *Softw. Process Improv. Pract.*, vol. 7, no. 1, pp. 3–15, 2002.

[21] P. Ralph *et al.*, "ACM SIGSOFT empirical standards," 2020.

[22] J. Hardy, S. Wyche, and T. Veinot, "Rural HCI research: Definitions, distinctions, methods, and opportunities," *Proc. ACM Human-Computer Interact.*, vol. 3, no. CSCW, pp. 1–33, 2019.

[23] C. Andrade, "The inconvenient truth about convenience and purposive samples," *Indian J. Psychol. Med.*, vol. 43, no. 1, pp. 86–88, 2021.

[24] S. Baltes and P. Ralph, "Sampling in software engineering research: A critical review and guidelines," *arXiv Prepr. arXiv2002.07764*, 2020.

[25] O. Cico, "Towards transferring lean software startup practices in software engineering education," in *{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, pp. 1686–1689, doi: 10.1145/3368089.3418542.

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

[27] V. Garousi and J. Zhi, "A survey of software testing practices in Canada," *J. Syst. Softw.*, vol. 86, no. 5, pp. 1354–1376, May 2013, doi: 10.1016/j.jss.2012.12.051.

[28] S. Baltes and S. Diehl, "Towards a Theory of Software Development Expertise," in Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2018, pp. 187–200, doi: 10.1145/3236024.3236061.

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

[30] S. Kirbas, A. Sen, B. Caglayan, A. Bener, and R. Mahmutogullari, "The effect of evolutionary coupling on software defects: an industrial case study on a legacy system," in *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2014, pp. 1–7.

[31] P. Achimugu, A. Selamat, and R. Ibrahim, "A Process Model for Efficient Software Engineering Practice," in *NEW TRENDS IN SOFTWARE METHODOLOGIES, TOOLS AND TECHNIQUES*, 2014, vol. 265, pp. 721–735, doi: 10.3233/978-1-61499-434-3-721.

[32] M. Cusumano, A. MacCormack, C. F. Kemerer, and B. Crandall, "Software development worldwide: The state of the practice," *IEEE Softw.*, vol. 20, no. 6, pp. 28–34, 2003.

[33] A. Majchrowski, C. Ponsard, S. Saadaoui, J. Flamand, and J. Deprez, "Software development practices in small entities: an ISO29110-based survey," *J. Softw. Evol. Process*, vol. 28, no. 11, pp. 990–999, 2016.

[34] N. J. Nunes and J. F. e Cunha, "Wisdom: {A} Software Engineering Method for Small Software Development Companies," *{IEEE} Softw.*, vol. 17, no. 5, pp. 113–119, 2000, doi: 10.1109/52.877877.

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

[36] T. Hall, H. Sharp, S. Beecham, N. Baddoo, and H. Robinson, "What do we know about developer motivation?," *IEEE Softw.*, vol. 25, no. 4, pp. 92–94, 2008.

[37] M. Omar, N. Katuk, S. L. Syed Abdullah, N. L. Hashim, and R. Romli, "Assessing personality types preferences amongst software developers: a case of Malaysia," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 3, pp. 1499–1504, 2015.

[38] J. Bosch, "Speed, data, and ecosystems: the future of software engineering," *IEEE Softw.*, vol. 33, no. 1, pp. 82–88, 2015.

[39] M. Tuape and Y. Ayalew, "A roadmap for a comparison framework for an adaptable software process improvement framework in small software companies," *Ann. Comput. Sci. Inf. Syst.*, vol. 20, pp. 133–141, 2019.

[40] L. Rivas, M. Pérez, L. Mendoza, and A. Grimán, "Selection criteria for software development tools for SMEs," in *Tenth International Conference on Enterprise Information Systems (Barcelona, Spain, 2008)*, 2008, pp. 252–255.

[41] C. Y. Laporte, R. V O'Connor, and L. H. G. Paucar, "Software Engineering Standards and Guides for Very Small Entities - Implementation in Two Start-ups," in *{ENASE} 2015 - Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering, Barcelona, Spain, 29-30 April, 2015*, 2015, pp. 5–15, doi: 10.5220/0005368500050015.

[42] J. Menezes Jr., C. Gusmao, and H. Moura, "Risk factors in software development projects: a systematic literature review," *Softw. Qual. J.*, vol. 27, no. 3, SI, pp. 1149–1174, Sep. 2019, doi: 10.1007/s11219-018-9427-5.

[43] M. Korkala, P. Abrahamsson, and P. Kyllonen, "A case study on the impact of customer communication on defects in agile software development," in *AGILE 2006 (AGILE'06)*, 2006, pp. 11-pp.

[44] M. Sashi, "Customer engagement, buyer-seller relationships, and social media," *Manag. Decis.*, 2012.