

RESEARCH ARTICLE

Theory on Non-Technical Characteristics Affecting Process Adoption in Small Software Companies: A Grounded Theory Study

MICHEAL TUAPE¹, (Member, IEEE), VICTORIA T. HASHEELA-MUFETI²,
AND JUSSI KASURINEN¹

¹Department of Software Engineering, Lappeenranta-Lahti University of Technology, 53850 Lappeenranta, Finland

²Department of Computing, Mathematics and Statistical Sciences, University of Namibia, Windhoek 9000, Namibia

Corresponding author: Micheal Tuape (micheal.tuape@lut.fi)

ABSTRACT Despite the inefficiency of software processes and products, small software companies (SSCs) enjoy a promising future. These companies are known to have few employees, creating the inefficiency that makes it challenging for the SSCs to adopt effective software practices. This subsequently introduces additional complexity, affecting software engineering processes' adoption. Using the Glaserian Grounded Theory, we conducted interviews ($N = 18$) with participants from SSCs intensively engaged in software development from four countries. We looked for the common traits that are identifiable as antecedents to the number of employees in a company to affect the adoption of software engineering processes. From the participants' experience, five non-technical characteristics (Risk, Competitive advantage, Resilience, Innovative capacity, and Management ability) emerged, complementing the number of employees to affect the process during software practice in SSCs. By the end of this study, we developed five hypotheses for predicting and explaining the adoption of software engineering processes by small software companies.

INDEX TERMS Small software companies, software engineering processes, software practice.

I. INTRODUCTION

The quality of software continues to be a significant challenge in the software industry, and this challenge is more visible in the Small Software Companies (SSCs) to the extent that up to 70 percent of software projects are challenged either due to defects, cost overruns or overshot scheduling related challenges [1]. Although the products produced by the SSCs fail to meet the expected quality, software is still central in influencing human activity. Recent studies indicate that most aspects of society depend on software; moreover, SSCs are responsible for up to 80 percent of software produced on the market. The contribution of SSCs cannot be underestimated since they represent up to 90 percent of software companies in the industry [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Hui Liu¹.

The challenge of poor quality software products is propagated by several explanations, including insufficient practice and poor understanding of processes, as cited by Tripathi *et al.* [3]. The authors assert that SSCs often lack systematic process knowledge for determining which type of processes are more relevant to their context. The problem is further exacerbated by the lack of process adoption in the practice of SSCs. These tools and processes have been developed to support better quality software production. Unfortunately, the processes are minimally utilized, given the low adoption of process tools amidst low quality software production in SSCs. Von Wangeheim *et al.* [2], cite a low usage of process tools in SSCs to be as low as 8 percent compared to the larger companies, similarly, Anacleto *et al.* [4] put it at 7 percent. Additionally, software engineering is a young field with most research concentrated in larger companies; owing to this, most tools and processes are prescriptive, making them difficult to adopt, especially for the SSCs.

The European Union defines small companies with a staff headcount of under 50 personnel and an annual turnover of 10 million Euros and below [5]. This definition has since been adopted to define SSCs and the staff headcount aspect widely used in software engineering research [1], [2], [26]. These companies face unique challenges that arise out of the structure and the context in which they operate. Nevertheless, most tools are the same and do not consider the specific uniqueness of these companies. For instance, a company with five employees may not effectively use a process for software testing easily utilizable by a company with 50 employees.

Similarly, a company with limited resources may find it challenging to hire more staff to deal with an urgent project but rather increase the workload. This puts pressure on the staff, resulting in ad hoc practices to avoid lengthy procedures in some of the processes during development. These unique aspects must be addressed differently if the SSCs must improve the quality of software products they produce.

The structure of SSCs is anchored on the number of employees in the company. This would determine how structured the software development practice in a company will be. It is important to note that the company's financial status most often determines the number of employees in a company. This means the availability of funds would determine how many people a company would employ, which relates directly to the financing aspect, and influences the extent to which software development practice can be effective. The structural challenges posed to the SSCs because of the number of employees introduce other complexities that ultimately affect software practice.

To overcome the challenges related to the number of employees, we need to identify its related complexities. We attempted to understand what happens in SSCs during practice by seeking the experiences of software practitioners in SSCs by undertaking 18 interviews from 4 countries using the Glaserian Grounded Theory (GT). The data shows the emergence of common traits defined in dimensions and features present differently in SSCs. We attempted to relate the non-technical characteristics to the limited adoption processes that affect software practice and are responsible for the failure of SSCs to attain expected software quality and customer satisfaction.

Identifying characteristics and how they affect process adoption is significant to enrich the understanding practice of the SSCs and advance theories to explain what causes the limited adoption of process tools, which would pave the way for possible solutions to a critical challenge in software practice as far as production quality products in SSCs is concerned. To achieve this, we answer two related research questions: (i) What non-technical characteristics influenced by the number of employees can be identified from the experience of software practitioners in SSCs? (ii) What hypothesis can emerge to explain the non-technical characteristics, and how do they affect the adaptability of process tools during software development in SSCs?

This study attempts to address multiple gaps and, in the process, makes significant contributions. First, the study extends the limited research on understanding the structure (number of personnel) and its effects on software practice and software process adoption, particularly in SSCs. To our knowledge, this is the first study of its kind to identify non-technical characteristics as antecedents to adopting software processes in conjunction with the number of employees. Secondly, we identified non-technical characteristics, assessed their mediating role to the number of employees in affecting process adoption in software practice by SSCs and developed hypotheses for process adoption. Thus, we advance theory through hypotheses to predict and explain how the non-technical characteristics complement the number of employees in affecting process adoption and software practice in general. Thirdly, we have not come across any previous study to our knowledge and through search in peer-reviewed databases that have empirically explored non-technical characteristics in complementing the number of employees in affecting process adoption in software practice of SSCs. Although the non-technical characteristics may seem to apply to all software companies and are not necessarily unique to the SSCs, our findings reveal that their severity differs in SSCs compared to their larger counterparts.

This work is organized as follows: The first section presents the introduction, describing the context in which this research was undertaken. The second section presents the research methodology and describes how the methodology is used. The third section details the research results, and the fourth section presents the developed theory and a discussion on how the theory affects process adoption. In the fifth section, a review of related work is presented, and finally, in the sixth and final section, a conclusion, limitations and future work is presented.

II. METHODOLOGY

A. GROUNDED THEORY

Grounded theory is the systematic theory generation from data analyzed through a rigorous research method [6]. The GT was developed by sociologists Glaser and Strauss [7], and GT has become a popular research method since its inception. It has provided a valuable methodological approach in several disciplines, including medical sociology [8], [9], nursing [10], education [11], [12] management [13], [14] among other disciplines. Despite such success, different researchers have been criticized for casually using GT, disregarding the intricacies of GT, which is complex and founded on an inductive paradigm that is entirely dissimilar from the traditional hypothetico-deductive models of research [15].

The GT has become popular in software engineering and related areas [16], [17], [18], [19]. Despite this growing popularity, Stol *et al.* [20] raise concerns about how software engineering researchers conduct GT. The authors claim that what many authors execute as GT is inconsistent with what

TABLE 1. Description of the participants interviewed from the respective countries.

	Ghana	Tanzania	Namibia	Finland	
First set of interviews	PG1 is a participant software developer with 12 years of industry experience, started a company with 4 people and has grown to 19 people developing software for clients in 6 African countries.	PT1 is a participant with 9 years of software testing & development experience in 2 SSCs in Dar-es-salaam. Currently employing 18 staff	PN1: is a participant with 14 years industry experience in software development and project management. has worked with bigger companies and started a SSC employing 6 developers 8 years ago.	PF1: is a participant with 10 years of industry experience in software development and testing started a SSC developing SaaS employing up to 18 developers.	4
Second set of interviews	PG2 is a software developer with 6 years of industry experience working in a company that has existed 14 years developing Fintech SaaS for both B2B and B2C clients with 24 staff, mostly agile (Scrum) PG3: a software engineer with 10 years of experience company develops medical support process software (medical logistics) use waterfall and agile (Scrum) employs 5 company existed for 7 years. PG4 software developer with 12 years of experience works in a company employing 8 and develops edu-tech and fin-tech software for 15 years so far using both waterfall and agile approaches PG5 a software developer and sometimes takes up the role of project manager for 20 years the company has existed for 20 years developing SaaS and bespoke software products with 27 developers and hybrid methods	PT2 is a software developer with 15 years' experience also owns the company that has existed for 12 years employs 7 developing web applications and ERP systems the company uses both agile (scrum) and waterfall PT3 is a software engineer and business owner with 9 years of experience. The company develops Fin-tech products supporting microlending employs 5, uses mostly waterfall. PT4 is a software developer with 14 years of experience have developed software for agriculture supply chains and Edu-tech software company existed for 9 years employing 4 use scrum and waterfall	PN2 is Software developer 14 years' experience in a 6-year-old company employing 4 people 3 of whom are coopted depending on tasks available. the company develops web applications and mobile apps. Mainly waterfall with some iterative development approaches. PN3 is a software developer and also a project manager with 13-years of experience started the 10-year-old company with 12 staff 4 of whom are not permanent. The company develops bespoke web applications and other software products using waterfall and agile methods. PN4 software architect with 13 years' experience started the 9-year-old SSC developing bespoke software mostly fin-tech support software employing 6 people and uses both traditional and agile methods	PF2 is a software engineer who also has software testing experience for 5 years has worked with this 15-year-old company for 1 year. The company develops SaaS with B2B clients mostly using scrum and employs 46 people PF3 is project manager and has also worked as business analyst with 8 years' experience started the 8-year-old company employing 16 people SaaS products using (agile)scrum in a modernized form. PF4 is a software developer with 10-years' experience in software development. the company has been in existence for 6 years employing 15 people 3 of which are not permanent. Agile (scrum) in a hybrid approach embedded software PF5 Software developer with 7 years of working with the same software company that employs 12 staff permanently and others coopted when required. the company uses agile methods (scrum) and is into Fintech has been in operation for 16 years.	14
	5	4	4	5	18

is considered as GT. In addition, the authors further note that there are many versions of GT, and three distinct versions are most dominant in software engineering research: Glaserian GT, Strauss and Corbin's GT (Straussian GT), and Charmaz's constructivist GT.

We used the Glaserian GT [21] approach because (i) GT is an ideal qualitative research methodology that constructively leverages the researchers to study software practitioners' social interactions and experiences in dealing with structural challenges. (ii) Software development practice focuses on people and how they interact in the software development space. (iii) The GT is optimal for studies that have not been thoroughly explored previously, given the limitation of research on software development practice, specifically SSCs [22], [1].

We followed the guidelines Stol *et al.* [20] developed, in which the authors highlight the importance of upholding consistency with a specific GT version. We aligned the

guidelines with the ideas of Glaser [6], who advises against preconception and recommends approaching the data with an open mind to remain genuinely open to the emergence of theory. We put aside our own beliefs, knowledge and experience about software practice in SSCs to allow the emergence of theory from the participants' experiences. A literature review was done after the emergence of the theory.

B. RESEARCH PROCESS

1) DATA COLLECTION

a: SAMPLE SELECTION

We conducted 18 qualitative interviews in two phases. In the first phase, we interviewed four participants selected through convenience sampling (easily accessible to the researchers). Selected participants were from Ghana (PG1), Tanzania (PT1), Namibia (PN1), and Finland (PF1). From a pool of participants who completed an initial survey, we selected four participants who met our interview criteria based on

their responses. We were interested in participants from different company size categories who also qualify within the SSCs definition to give us the seed categories known as start list. This was followed by interviewing 14 participants selected through purposive sampling in the second phase. We used purposive sampling because we wanted as much insight as possible into the experiences in the different size profiles of SSCs. We applied the heterogenous, also known as maximum variation purposive sampling, to meet our expectation of uncovering the unique and diverse experiences of the participants in theory development.

Details describing the participants are in Table 1. We interviewed four participants about software practice with a vague understanding of general software practices and processes in SSCs. The initial interviews were used as the start list in the data collection and guided the determination of the two research questions.

b: INTERVIEWS

The first author conducted interviews in English between March and November 2021 to ensure adherence to the principles and guidelines. The researchers developed a discussion guide to guide participants. All interviews were recorded and transcribed verbatim with the consent of the participants. Consent was sought prior, and during the interview, we also explained to the participants why we were collecting the data. We started the interviews with a broad theme of understanding software practices in SSCs, and the analysis was on (a) what practitioners thought, felt, and believed about their companies, (b) what makes their companies work the way they do, and (c) how they overcome inherent challenges when developing software, given their company structure and context. We conducted the interviews in two phases; phase 1 had the four initial interviews based on the broad theme, from which we refined the nine guiding questions, and in phase 2, the other 14 interviews, the former refined the questions asked in the latter which were more specific and refined. The interviewer asked participants open-ended questions to allow practitioners to speak freely and openly about their experiences during software development in their companies. The participants spent an average of 40 minutes on each interview. The interviewer occasionally asked clarifying questions in cases where the responses were not detailed enough or unclear. In order to avoid interviewer preconceptions biasing the interview, the questions were asked as they were in the questionnaire guide, with no interpretation allowed by the interviewer.

c: THEORETICAL SAMPLING

Theoretical sampling in GT entails collecting and analyzing (coding) data simultaneously to generate theory. Coding was done in 3 phases: open, selective, and theoretical coding (explained in the following subsections). In each coding phase, data is analyzed to develop theory (explanations) while suggesting other cases to sample. Though out the data collection process, we recorded spontaneous ideas or

empirical indicators written as reflections or ideas (memos) that were eventually used to develop qualitative codes and provide direction for further data collection. We transferred the empirical indicator into analytical memos to generate a code representing the excerpt. We sorted the related concepts using the memos as a starting point (memo sorting). Theme sorting, or theoretical sorting, was a continuous process. Glaser points out that memo sorting is imperative during GT and cannot be ignored [23]. Data collection continued along the coding until theoretical saturation was arrived at. Theoretical saturation occurs when no new categories or properties emerge from the data, and at this point, further data collection is stopped.

2) OPEN CODING

Open coding of the interview transcripts was conducted paragraph by paragraph to identify the discrete parts. In the initial phase of the data analysis, open coding exposed the data to new theoretical possibilities. During the open coding, data was 'fractured' into categories by labelling paragraph after paragraph. Straus [24] argues that it is best to code line by line to achieve optimal theoretical coverage; however, this does not mean that coding sentences, paragraphs, or entire documents should be ignored. The memos were used to dissect the data in sentences. They labelled it to compare similar elements in the data by gathering all data (quotes) already tagged with a specific code using the most appropriate coding strategy for identifying relevant information called empirical indicators. We then assigned labels (called codes) to these empirical indicators. At this point, we constantly compared the new categories with what was initially discovered to ensure that only new categories were recorded. This continued with the interviews while labelling the data as codes until we noticed nothing new coming out of the interviews (reach theoretical saturation).

3) SELECTIVE CODING

As part of selective coding, we recognized the relationships between codes to create key attributes, limiting the coding to variables associated with one or more core variables to develop the theory. Unlike in open coding, in which we broke down the data into discrete pieces, in selective coding, we connected the codes and organized them by applying Glaser's [21]. The 6Cs coding category framework is used to structure the analysis process to develop theory. These include *conditions* (or antecedents), *causes* (including sources), *consequences*, *context*, *contingencies* (or variables), and *covariances* (variables which are connected, changing together, without a causal connection). The 6C is one of the various coding families used to represent the relationship between categories and ultimately generate a theory [21]. We used the 6C coding family to examine the operational context, the causes of issues and other situations, consequences, contingencies, and covariance to determine the empirical indicators evolving around SSCs. The core variable guided data collection while referring to and constantly iterating

the data. We continuously referred to the underlying data when reading the codes repeatedly during selective coding and assembling them into the 19 categories to conclude the selective coding process. The researchers created categories and developed more abstract categories based on existing codes.

4) THEORETICAL CODING

The final step in analyzing the data was to apply theoretical coding; We grouped the attributes around the five core categories to explain the phenomena of interest. We developed 5 core categories based on constant comparison and relationships identified in the 19 selective codes. After initial selective and open coding cycles, related categories emerged from qualitative data. Flexibility, for instance, was initially a core category and was later downgraded to an attribute of management.

We established conceptual relationships between the substantive attributes, resulting in the emergence of the 5 non-technical characteristics from the data. The core category represented the central thesis of this study upon which we induced the hypothesis. To answer the second research question, we used the 5 characteristics and applied tools like red flags, flip-flops, and multiple perspectives while referring to the data and profiled the companies being studied based on the number of employees. We assigned properties to the profiles while looking for variations of the characteristics within the profiles using frequencies, intensity, and time as we compared extremes in the different profiles.

III. RESULTS

A. OVERVIEW

This section presents the two broad categories of description that emerge from the GT in which we answered the first research question; What non-technical characteristics influenced by the number of employees can be identified from the experience of software practitioners in SSCs? The first categories are reflected as attributes of the characteristics evaluated using Glasser's [21] 6C coding family [25] during the respective coding processes. The concepts, their relationships, circumstances of their occurrences, and consequences arising from software practice were evaluated to generate attributes using descriptive-focused coding in cases where the research questions were clear. In contrast, interpretation-focused coding was used where the research questions were unclear. The second set of categories is reflected as the characteristics of the SSCs that emerged after applying the presumption-focused coding strategy on the 16 attributes.

B. ATTRIBUTES OF THE CHARACTERISTICS

This subsection presents categories/selective codes as attributes generated from the themes of the data (transcripts). We used the 6C coding framework to structure the 21 categories presented herein. The attributes were compared for relationships to generate 16 attributes presented

in this subsection. After comparison, we found 3 attributes connected to financial risk and owing to the relationships, we merged them as one. Similarly, organisation learning had 2; structural risk had 2; differentiation advantage also had 2 attributes with identifiable relationships to have an ultimate total of the 16 attributes presented in Figure 1.

1) CONTEXT

From the description of the companies and how they operate as presented by the participants, we can fully understand the circumstances surrounding the SSCs. We used context to identify one attribute that is the structure of the companies presented below.

a: STRUCTURE

Describes the structure of the companies that produce software for the market and have between 3 and 46 employees. In these descriptions, the events that happen within the companies are correlated with the definition of the companies. In the descriptions, a set of conditions is placed where and when software is developed, enabling an overview of how software is developed.

"...our company currently employs 16 staff ... and in some instances, we are overwhelmed and we have to call in some developers if we have something pressing to accomplish in a short time... we also once in a while bring in some expertise from out of the company with highly specialized situation when there is in need ... the cost implication of a bigger team is complex at the moment".(PF3)

An example of this is the number of staff in the company being responsible for the relationship with inherent challenges, such as having a manageable number of staff in a company and the cost implication of having more staff presenting a risk factor faced by the companies.

2) CONDITION

In this session, the participants discuss what challenges affect the success of software development. Participants identified five attributes influencing their companies' development process: strategy, finance, differentiation advantage, flexibility, and organization.

a: STRATEGY

The participants believe that they are often surrounded by dilemmas that make it challenging to define their company's strategic directions. During the interviews, some participants described situations where they faced difficult decisions, such as entering a new market. With limited experience, such dilemmas may raise many questions about how to begin, what steps to take, and what strategies to employ. The SSCs may require unique approaches [26], according to Richardson and Wangeheim [31]. Technology change [27], new regulations and competitors [28] can also threaten SSCs; technology can create new opportunities or render the current process obsolete, affecting how SSCs function [33]. A participant expressed a scenario the company faces, as captioned in the quote.

Open coding	Selective coding	Theoretical coding
How do we start? What to do next, what strategies to take up, Invest in life time savings to get the company off the ground, where will the money come from, difficulty of having money to run the company since you may start with out clients, small number of staff that is manageable, fewer people in the company,	Financial risk Strategic risk Reputation Structural risk	Risk
Need for a good team, efficiency in producing software, adjust accordingly to meet demands of the market, better understanding of the tools, clear process, ultimate product satisfaction, produce software at reduced cost, clients getting value for money	Cost Advantage Differentiation Advantage	Competitive advantage
Changing market, software products perceived to be expensive, software market tends to be urban based market, increasing demand for software product, similarity of products, tools demanded by customers, take advantage of favorable government policies,	Market structure Competitive structure Regulation and policy	Resilience
Internal sharing of challenges and knowledge, collaboration with other organisations, we are better with the tools we know, afraid of making mistakes with new tools and techniques, this works for us, we are too busy we cant start with new tools, desire to transform practice, accept new ideas, process improvement strategies,	Organisation learning Knowledge Absorbtion	Innovative capacity
Organizational goals, clear project objectives, difficult to follow plan, schedule and budgeting complexity, task allocation, prioritization, motivate staff, failure to meet targets, everyone knows his or her role, persuasion in acting on common goals, cohesion, multitasking, individual style of work, free style flexibility in tasks, quick response to change, limiting documentation to save time, encouragement of self control in a free style work approach	Leadership Organisation Planning Control Flexibility	Management Ability

FIGURE 1. Outcome of the coding process, open codes are excerpts of themes from transcripts, descriptive and interpretive strategies applied while making a constant comparison for developing the selective codes and the theoretical codes generated from the constant comparison and relationships of the selective codes.

“..... yes, there are difficult situations that come by of course you know being a small company and at the start, we had to self-finance we started in a situation that we had to convince clients who most times are not very forthcoming..... so we found ourselves in a dilemma of not knowing what to do and sometimes even how to start on a new product is challenging...”. (PG1)

b: FINANCE

Below is a caption describing a participant’s concern about financing. Concerning company financing, staff retention is mentioned as an expensive endeavour, and the participants also mentioned client insecurities. However, the company must continue operating without financing other than their savings.

“.....Financing is a challenge, especially when you cannot get access or support from commercial banks, and sometimes the venture capital opportunities are not readily available due to the risk of a new product in the market... These guys want to see you on your feet before they come in ... we managed to pull through although it was not easy surely it was a big risk, yet money to run the company is very important” (PF3)

c: DIFFERENTIATION ADVANTAGE

The participants raised concerns that affected the ability of the company to gain a differentiation advantage. A few concerns

include understanding the tools and processes to develop software, achieving production efficiency, and making necessary adjustments to meet market demands. Consequently, some participants had to switch from traditional software development to SaaS so that their company could gain an advantage over its competitors. Captioned below is an interesting case of one of the participants.

“... we started with bespoke software and had to adjust accordingly so, we decided to venture in SaaS after about two products in the market it seemed obvious that we needed more time on developing and maintaining our products to satisfy our customers” (PF2)

d: FLEXIBILITY

Hoch et al. [23] suggests that flexibility is a good trait for SSCs that constantly change; flexibility is a crucial pillar of success. Participant descriptions of the factors that contribute to and explain flexibility during software development in their companies include freedom of work, allowing employees to pursue their styles and ways of completing tasks, taking on multiple tasks assigned to one person, and choosing individual work hours. Flexible employees will take on more work because they are willing to do what is required to accomplish tasks. Similarly, multi-taskers can perform more tasks [29], do more responsibilities, and offer more than their less

flexible counterparts. Management must be flexible when dealing with employees [30]. Workers who have flexible managers achieve goals more quickly while getting feedback, guidance, and recognition to maximize performance. To support these viewpoints, here is a quote from one of the participants:

“.....it becomes easier for us ... mostly because we encourage our developers to employ diverse styles and methods of work to the extent that it is okay to have flexible work hours.... what is important is that the work should be done. This flexibility has worked for us...” (PT 4)

e: ORGANIZATION

There are some challenges that the companies' form might not address internally. The participants describe a complex role structure, budgetary limitations, and insufficient software development tools. Participants further believed that internal arrangements might not change the effects of these concerns, although mitigative action can be put in place to improve customer satisfaction. Managers need to consider the organization as a sum of its processes [31]. The occurrence of unstructured processes and poor planning is typical of SSC, as observed by Tuape and Ayalew [1]. These 'unstructured' processes are discussed severally by many other researchers who propose Software process improvement (SPI) as remedies to ensure quality software processes, especially in SSC [32], [33], [34], [35], [36]. In the ever-changing environment, some participants believe that small teams encourage freedom of work, which they can exploit.

“... we use the methods that work best. we encourage the team to work with flexibility so that some discussions can be made on the spot depending on the task at hand.” (PN2)

Additionally, participants believe these challenges can expose the organization to financial and strategic risks, making leadership and management even more difficult due to laissez-faire attitudes in the workplace. On the other hand, participants also believe that free and individual work styles help them respond quickly to challenges that require change. These assumptions describe that a company responds successfully to unforeseen challenges, resulting in a company producing more cost-effective and high-quality products than its competitors. This condition creates a cost and differentiation advantage for the company.

3) CAUSE

To determine the situation that led to the attributes that influence the practice of the SSCs, we apply two coding strategies first, using descriptive coding and interpretational coding to generate 4 attributes: budgetary constraints, challenges in meeting time-to-market, competitive structure, and the possibility of tapping new knowledge from the challenges faced during practice.

a: BUDGETARY CONSTRAINTS

Budget constraints lead to a challenge in software practice. As evidence of this challenge, lack of finance is responsible for the complexity of funding quality assurance procedures and methods since these can be expensive when

a company has budgetary constraints. Additional description of challenges in practice resulting from budgetary constraints to finance staff training and development for further staff skilling and the inability to use tools that require monetary commitment. One of the participants describes this situation by stating:

“... the cost implications of running the company especially when you have few clients make things very difficult the money is needed for all sorts of activities, yet the budget is limited..... training and staff development, for example, becomes very difficult to prioritize”. (PT 3)

b: CHALLENGES IN MEETING TIME-TO-MARKET

The participants described the challenges they faced within their companies, affecting product delivery within a feasible timeframe. These challenges include the pressure caused by customers seeking a working product in a relatively short time, the competitive environment in which they operate, and the bad experience with software development tools, approaches and methods. Companies can save resources by analyzing the market structure and focusing on easily accessible markets; for example, in [37] and [38], customer involvement in software production is highlighted as a significant factor affecting quality and causing failure.

“..... the customers are much more concerned with a working product, and the methods are least of the concerns to them.... It is sometimes difficult when they are not available when needed to sort out some of the issues associated with unclear requirements”. (PG1)

c: NEW KNOWLEDGE FROM THE CHALLENGES

The changing interests of customers affect the project's requirements and the technical delivery of the product. As a result, there is an opportunity for organizational learning due to challenges. The participants believe that their companies fail to leverage this opportunity due to limited documentation. In support of this, an example of this is the lost opportunity in an undocumented try and error procedure attempt that was successful when it recurs. Reflection and creativity are required of SSCs, as they generate, retain, and transfer knowledge to gain experience from challenges and improve over time. Organizational learning includes conscious and unconscious elements and the acquisition of knowledge, access to information, and evaluation of information that affect organizational processes as described by Lyles and Easterby-Smith [39]. The study's results demonstrate that the organizational learning capacity influences the innovative performance of Small and Medium Enterprises (SMEs) which actually covers a sizable proportion of SSCs Ibert [40]; Their study further provides evidence for these relations and demonstrates that they are significant and positive in the context of SSCs. Surprisingly, Ibert confirms that very few organizations claim to have lowered software production costs or increased software quality [40].

Nevertheless, many claims to have improved the work situation for software developers and managers. While Gasston and Halloran [41] suggest that to achieve

optimal benefits from implementing process improvement programs, organizations must move towards becoming what is termed “a learning organization.” Software process assessment “leads to the identification and selection of vital activities for improvement and the continuous application of improvements to match business needs”. A participant is quoted in the excerpt below as saying.

“...we are always busy, and experience sharing opportunities could help us learn from how we overcome mistakes... but it is not easy because of our schedules ...and because of our agile approach and limited documentation, we could be missing out on some improvement and learning opportunity”. (PT4)

d: COMPETITIVE STRUCTURE OF THE MARKET

External competition shapes the structure of the market to the extent that it can dictate the company’s internal dynamics. The participants describe changes in internal behaviour because of external competition to counter the changes in the market structure and remain relevant in the business. Although this situation occurs in all sizes of software companies in SSCs it may involve adopting different styles of developing software products particularly the use of ad hoc practices. Ojala [42] studied six Finnish SSCs and their entry into the Japanese market despite the cultural differences; the author cites a market size and a sophisticated industry structure as complex situations to deal with. Software companies often target large markets with many segments to support business. In addition, they specialize in a single segment that most closely matches their ideal customer, such as Fin-tech, gaming, or mobile apps. In areas where the target market is small, most SSCs focus on several segments to achieve the volume needed. For example, SSCs venture into games, Fin-tech, and websites simultaneously. This makes the target market structure of the SSCs a critical attribute for the characteristics of a software company since this attribute shapes quality dimensions that the company can achieve. In the participants’ words, one states;

“...many software companies in the market are switching to producing their software and just sell services to clients this has changed the behaviours and preference of our customers ...especially with reliable and cheaper cloud services” (PF4)

4) CONSEQUENCES

We employed different coding strategies to identify and describe the pleasant and unpleasant results of the conditions affecting the software development processes. We established connections or relationships between some codes to develop and describe six categories: financial risks, business interruption, planning, control, knowledge absorption, and organizational learning.

a: FINANCIAL RISK

Participants describe their companies as having negative consequences of financial challenges, such as investing years of savings under challenging circumstances to finance the company’s operations, although finding customers may take some time. Several participants described their companies as

synonymous with financial challenges, adversely affecting normal operations and their products. The situation described here is risky due to the compromise involved. Financial risk creates great pressure to succeed, given that financial stability is essential because running a business, paying staff, and investing in growth costs money Majchrowski *et al.* [27]. Additionally, Latham [43] argues that start-ups are likelier to fail than larger established companies because financial risk influences decisions, such as the number of employees to hire and sometimes the nature of tools and processes a company can use. For example, one of the participants is quoted in an excerpt below as saying.

“...the cost of performing some tasks make it complex for us so we have to let go of some things and settle them after because we can have the money to take up more staff ... this ultimately slows us down because the task has to wait for the next available staff to take it up” (PG3)

b: BUSINESS INTERRUPTION

Through the relationships, the description of the negative impacts of the small number of staff explains structural risk and leadership difficulties as an attribute that defines risk in SSCs. This risk attribute is more prominent in smaller companies. Participant PT2 relates this to the difficulty leaders go through, yet they need to motivate the team, who may be very exhausted to the extent that motivating the team becomes difficult. References [44], [45] Such situations affect the team’s ability for processes adoption.

“...we are very few in numbers, and most of the time, we have to take on more than one role, which makes it difficult to motivate the team to do a good job. Some projects become so exhausting that we all leave when we are exhausted at the end of the day.”(PT2)

Another participant expresses a difference with the bigger companies in dealing with situations of sickness or absence of any team member, saying they currently treat such situations differently and business remains interrupted. However, the participant PG2 adds that

“...we started as a team of 5 and before increasing the number of staff we had challenges of taking extra responsibility that could render the team unproductive”.

c: CONTROL

The participants describe practices that prevent the detection of errors and deviations from procedures and take corrective measures when necessary. Some of the descriptions identified include the conditions, such as assumed self-control due to limited staff monitoring to achieve independence due to a small team, and limiting documentation to save time. However, documentation is the basis for comparing attained results against planned work. A relationship in all these was combined to describe control. Controlling is the process of monitoring organizational progress in achieving its goals. It measures, compares, finds deviations, and corrects organizational activities [46]. It involves quantifying performance, comparing existing standards, finding nonconformities, and fixing them. As a result, results are controlled by monitoring how people act. When results differ from the plan, engaging

those in charge is essential. A company's ability to control internal action gives the company credence to growing, although excessive control can also affect the organization; therefore, understanding the extent of control in an organization is essential. One of the excerpts from participants was that:

"...some of our clients are not interested in documentation ... this is why we give it little attention although at a certain point we got into trouble to adjust to make documentation because this other client wanted it....sure at the end it was imperative, especially for tracing requirements and quality control" (PN2)

d: PLANNING

The participants described actions of ad hoc practices based on assumptions that everyone knows his or her role. Overlapping tasks have consequences for the regular planning in SSCs. Planning is an ongoing step and can be highly specialized based on organizational and team goals [48]. Planning is when the manager creates a detailed action plan aimed at achieving some organizational goals [47]. It is up to the manager to recognize that planning goals are essential within their area. This is the second managerial function; this step requires one to decide how resources will be distributed and organize personnel according to the plan. It requires categorizing varied roles and assuring that they have assigned the correct number of employees to carry out the plan [47]. This will also require delegating authority, allocating work, and providing direction so the team can accomplish the planned work. In one of the cases, the description provided by a participant underscores the importance of planning as a time-wasting process with the justification that the small number may not require excessive planning as quoted:

"...sticking to the plan sometimes is complicated especially that we are few and have lots of tasks ... in some cases, the clients are not very cooperative, so we try to be flexible and just have to ensure things are done ... so long as it works" (PN4)

e: KNOWLEDGE ABSORPTION

The description of actions that occur in the SSCs with an impact on the capacity of the company to absorb new knowledge, some of the participants' expressions that fit this description include how they look out for new tools and methods, the desire for new ways to transform practice, process and adapting to change when new tools or methods have been put in place. Studies involving a firm's innovation performance, aspiration level, and organizational learning are said to be innovative. Zahra and George [49] proposed a reconceptualization of ACAP, defining it as "a set of organizational routines and processes by which a company acquires, assimilates, transforms and exploits knowledge to produce a dynamic organizational capability". As further illustrated in the quote below, these effects describe a company's appetite for absorbing knowledge:

"...we do not use scum as prescribed in the book ... we have adopted our own approach to fit within our context. For example, our team size cannot allow us to do as in the book, but in our way... we go to the sprint meeting with a

predesigned feature, and our meetings are motioned to evaluate and review the design ... we do this to help us move first and we started by following the book, but it was not working so we continued to adjust until we found what works for us" (PG5)

f: ORGANIZATION LEARNING

Organizational learning is described from the participant's perception of the adverse effects of the ever-changing market structure. The perception that software is expensive as expressed by the clients and the increasing demand for software. The negative impacts came from analyzing the developers' concepts being too busy to learn new techniques and methods and being better contented with the tools already known to the developers. This established the basis of describing the high appetite for learning from the challenges faced by the SSCs.

"As developers, we are keen on what we know works... We are hesitant to bring in a new method since we don't have the time to learn new things ... to meet our targets, especially since we are working on our new payment solution ... we have many new services to roll out, so there is no time to learn new things." (PG4)

5) CONTINGENCIES

The described eventuality affects software processes both positively and negatively. Software companies keen on what contingencies are in place can put up mechanisms to overcome the adverse effects and strategies to tap into the opportunities that come with the positives. We identified 3 categories from the description that relate to this, reputation and leadership provided in an organization, government regulations and policy. This describes the possible occurrences that cannot be predicted with certainties, such as the regulatory requirements that affect SSCs and the challenges that affect the reputation of the companies.

a: LEADERSHIP

Leadership is responsible for providing a positive influence towards achieving the organization's goals, for example, motivating the team towards teamwork and developing a positive attitude towards efficiency and effectiveness. In cases where leadership is bad morale, teamwork and the appropriate attitude to take the company forward may hinder the attainment of the organizational goals. Coordination and motivation are essential in achieving harmony and encouraging individuals to achieve the company's goals. Efficient managers need to be influential leaders since leadership implies communion, and people tend to follow those who offer a means of satisfying their needs, hopes and aspirations. Participants describe a case where leadership has driven the company towards a positive direction.

"...we have achieved a lot even if we are a small team simply because we have always worked as a solid team ... we use a flat organization structure and our leaders encourage free communication to keep the team focused and motivated..." (PF5)

b: REGULATIONS AND POLICY

Government regulations and policies that tend to create unpredictable situations in the software market confuse

the SSCs. The participants describe how new government regulations, such as a requirement for fintech companies to implement security standardization, took them off the balance because they did not know how to start and what strategies to use. Taking the steps often caused uncertainty that exposed the company to risks after affecting regular business. Evidence of this is seen in countries like Brazil [50], Malaysia [51], and Pakistan [52], where the government's favourite policy has helped software businesses grow and avoid fatalities. Government regulations and policy influences control several markets. Suppose suppliers in a market are supported, or there is control in market prices; in this case, SSCs must acclimatize to trade schemes to take advantage of probable government support and act according to regulatory requirements. In Brazil, for example, the software industry has experienced enormous growth researchers Montoni *et al.* [50] relate the successful growth of the Brazilian software industry to the MPS BR, a Softex program supported by the Ministry of Science, Technology in Brazil.

"...the government made it a requirement for all fintech software companies to standardize with the ISO, and this made the business a little slow because we needed this standardization to get business with most agencies." (PG4)

c: REPUTATION

The reputation of the company can be compromised, which can lead to loss of business. The participants describe unpredictable customer behaviour and limited customer knowledge as situations that may pose a reputation risk to the company. The participants also are not very bothered about this, given that they are unpredictable. Laporte *et al.* [35] discuss image management while encouraging feedback and responding to customer challenges in an empathetic and grateful manner. It is crucial to maintain a reputation regardless of negative feedback. Lawsuits and fines for employee injuries, property damage, and failure to meet contractual obligations can tarnish the reputation of SSCs.

Moreover, due to the small number of staff, SSCs are vulnerable to liability obligations, resulting in reputational damage. SSCs may be unable to pay for damages without compromising their cash flow. Outsourcing software development to SSCs is a common practice, but their products may not be defect-free, and as a result, the SSCs may be at risk, negatively affecting their reputations. These unexpected situations breed confusion as quoted in one of the excerpts below:

"...at a certain point what market to target was not easy we tried to build a market base by developing software for SMEs, but the market was not forthcoming... we had to change and focus to a product which worked so well for us... after a year then the SMEs started approaching us... yet we had changed our business strategy... sure it was disappointing, but we had no choice". (PF1)

6) COVARIANCE

Covariance describes how situations compare to each other and identify the possible changes that may occur out of

comparing the identified concepts. The use of interpretation-focused coding strategies was necessary in identifying and describing attributes like cost advantage and differentiation advantage.

a: COST ADVANTAGE

The participants describe the need for a good software development team and the efficiency of the product to be produced by the software team. Comparing the two interests describes the need for the software development teams to strive toward the attainment of effective means of producing software, some of which include a reduced cost by enhancing the team's ability to produce a better-quality product at a reduced cost, increasing customers happiness with the product to attain cost advantage which has a significant cost implication on the company.

"..... putting together and keeping a good team ... has been a challenge. As soon as we hire a new employee, he or she will need to know the rest of the team. When a member of the team leaves, it is not a pleasant experience. Yet, the employee will already be well-versed in working for the company. This is not an easy one..... some of our worries are how efficient the team would be to develop a product of good quality, especially when we have just put together a new team to accomplish specific tasks" (PG2)

b: DIFFERENTIATION ADVANTAGE

The combination of factors such as the team's challenges in the production of bespoke software and the inability to meet customer expectations affecting ultimate product satisfaction describes the urge for the companies seeking differentiation advantage because of the influence to shift to product-based software after frustrations from bespoke software. These relationships have given the SSCs premises of developing unique products of their own. This is also visible in the excerpt of one of the participants who is quoted saying.

"... maintenance of the customer's software sometimes is endless and yet the client is not willing to pay extra for unforeseen features... this makes us exhausted with tasks which keep on resurfacing when we ventured into our own products it's easy to see the benefits because we have a unique product, and our customers keep growing we have integrated many online payment products, especially with the growth of eCommerce" (PF4)

C. THE NON-TECHNICAL CHARACTERISTICS

This subsection presents the five characteristics that emerged by constantly comparing the 16 attributes generated while establishing relationships between them to identify the 5 characteristics during the GT. We present the definitions of the characteristics and the respective attributes that were merged to form the characteristic.

1) RISK

Risk orientation is an expression of a preference for a threat versus an unknown outcome which depends upon the probabilistic framing of gains and losses and the company status-quo position relative to expected achievements and

setbacks [81]. Risk attributes are financial, strategic, reputation and structural risks. These attributes are identifiable with the SSCs and can be aggregated to form an overall risk to describe the risk exposure of the SSCs. This explains why risk management is crucial for most organizations; if all the parameters of risk are well managed, alignment and improvement of essential organizational and business processes becomes easier [53]. This, however, is a complication observed in most companies, although the extent of complexity is much severe in SSCs. The different size categories face different exposure, and the companies react differently to the risk parameters.

2) COMPETITIVE ADVANTAGE

The attributes that define competitive advantage are cost advantage and differentiation advantage. These factors make it possible for SSCs to produce better software products in terms of cost and to the clients' satisfaction better than other rivals. This is what is referred to as a competitive advantage. These factors allow the productive entity to generate more sales or superior margins than its market rivals. In SSCs, it is often believed that smaller teams could gain a competitive advantage by creating a unique team. A small team of highly qualified professionals with a common background is usually advantageous [54] for SSCs, despite the challenge of putting together a good team [46]. This will mean the competitive advantage will be anchored more on human capital, which may not be sustainable. Porter's two basic types of competitive advantage are cost advantage and differentiation advantage [55]. The competitive advantage philosophy suggests that everyone is better off if decisions are made based on the competitive edge at all stages nationwide, company, local, and individual. Such a competitive advantage helps larger companies cope with ever-changing environmental demands, such as changes in regulation and globalization [43]. According to Adler and Bartholomew (1), this is like demanding optimal resources and globalization of manufacturing and services. Since politicians and thick boundaries delineate nations' territory, the issue of offshoring and the use of protectionist measures arises.

3) RESILIENCE

The forces that shape the market of the SSCs can sometimes be destructive, especially when the SSCs have not yet built sufficient resilience to overcome whatever challenges that come along with competition, the market's structure and its characteristics and the regulatory and policy infrastructure in place. Once a SSC has not built sufficient resilience, it is subject to a failure to cope with market effects' destructive waves, and adopting processes may be challenging.

4) INNOVATIVE CAPACITY

This relates to the SSCs' ability to engage in innovation, that is, the introduction of new processes, tools, or methods for software practice in the organization. The ability to innovate is among the most vital factors that impact

a company's performance. The significant attributes in defining innovative capacity are the ability to learn and the knowledge absorption capacity. A broad definition of organizational learning is how organizations acquire new knowledge. It is essential in organizations operating in turbulent environments, where knowledge acts as a critical resource [56]. It offers an opportunity to create, retain, and transfer knowledge from the creators to the users within an organization. Organizations can improve over time as it gains experience [41], a process responsible for creating knowledge to better an organization. Authors [48], [57] define organizational learning as detecting errors and fixing processes, and organizational learning capacities are improving firm performance over time. Learning capacity is a critical factor in the growth of innovation and organizational effectiveness [58]. In today's global economy, the world of business must develop the capacity to innovate products and processes to maintain a competitive advantage and survive.

5) MANAGEMENT ABILITY

The SSCs should have specific attributes that fulfil organizational objectives: organizational management ability is vital in an organization, and any failure in management can cause the organization to fail. A company should structure its leadership, organize its affairs, ensure systematic planning, and be able to adjust the daily activity to control while exercising sufficient flexibility.

The importance of management in software engineering and how it impacts SSC success cannot be overstated [43], [59]. Managing involves planning, organizing, leading, and controlling [60], enabling the completion of a project on time and within budget [46]. Similarly, the ISO/IEC 29110-5-1-2 standard 7 provides a management and engineering guide to most SSCs, as observed by Mesquida and Mas [46]. The main attributes that shape the management ability of the SSCs are organizational processes, planning and organizing, leading, controlling and the extent of flexibility in the organization.

IV. THEORY ON NON-TECHNICAL CHARACTERISTICS OF SSCs AND HOW THEY AFFECT PROCESS ADOPTION

In Section III, we presented the characteristics generated from grounded theory in which we answered the first research question asking; what non-technical characteristics influenced by the number of employees can be identified from the experience of software practitioners in SSCs? In this section, we present the hypotheses that emerge from the non-technical characteristics in the previous research question to answer the second research question, which was to find out what hypothesis can emerge to explain the non-technical characteristics and how they affect the adaptability of process tools during software development in SSCs? The grounded theory presents five hypotheses of the non-technical characteristics (core categories) and how they affect process adoption during software practice in SSCs, as shown below. We attempted to illustrate how the theory emerged from the data connecting

the key relationship between the 16 categories as dimensions of the hypotheses.

To evaluate the intensity of the non-technical characteristics and the subsequent dimensions, we profiled the 18 companies according to the number of employees. Three profiles separate the 18 companies in 3 distinct profiles: less than 10, between 11 and 20, and between 21 and 50 employees. For this study, we refer to the profiles as initial, intermediate, and pivoting, respectively. Participants' experiences were correlated with the intensity of non-technical characteristics and the number of employees in their respective companies. We iterated into the data to adduce evidence to justify the intensity of how the non-technical characteristics occur differently in the smaller companies compared to their larger counterparts and hence the hypotheses.

A. HYPOTHESIS 1. HIGHER RISK EXPOSURE INCREASES THE PERCEIVED DIFFICULTY IN PROCESS ADOPTION IN THE SMALLER CATEGORY OF SSCs COMPARED TO THOSE IN THE LARGER CATEGORY

Hypothesis 1 has 5 dimensions: financial, strategic, reputation, structural and business interruption; each of the dimensions illustrate the different risks a company can be exposed to. Overall risk exposes SSCs to different forms of vulnerability, including the difficulty in adopting processes. The exposure effect is much more pronounced in the initial profile than in the pivoting profile. In this subsection, we discuss each of the dimensions of risk and relate them to the difficulty in process adoption while comparing the smaller companies to the larger ones.

Comments from participants show that the severity of financial risk on the development processes is hinged on affecting the processes used during software development. It is common for smaller companies to have a weak financial base, limiting their choices of methodologies and processes due to the expense of training staff and hiring a large number of employees. As mentioned by the participants in the initial profile, financial risk is the primary concern of the SSCs (n=18). That explains the limited use of standardization and certification in SSCs.

Unpredictable situations make it challenging to implement formalized decision-making processes. Discussions with the participants portray that a lack of a clear strategy is one of the most challenging aspects of SSCs, especially for the companies in the initial profile. This implies that the SSCs may not know what steps to take on essential choices like structure, product delivery and marketing strategies. Decisions of strategy expose the SSCs to strategic risks that may be difficult to make for those in the initial profile compared to the ones in the pivoting profile. As a result, the SSCs in the initial profile would then make hasty decisions to take any business opportunity available, sometimes not following specific processes.

Reputation risk is a significant dimension of the risk hypothesis, it occurs when activities that could support the building of a good reputation is ignored, although it is

supposed to be a company's important asset. The views of the participants are tagged to unpredictable situations. One of the participants (PN4) is quoted saying: "*sometimes we are better not putting this into thought and better deal with it when it comes*". This is often associated with the fact that the SSCs in the initial profile often lack the resources to deal with these unpredictable situations, creating an attitude to prefer ad hoc practices to avoid certain situations, including processes that may not be clear to them. Another participant (PT2) expressed their challenge by saying: "*being unfamiliar with government procedures made us lose business and affected our reputation*".

Business interruption is often caused by structural challenges like the number of staff in a company can cause a business to be interrupted in many ways. For example, if the company has a small team, an illness could impact the normal processes in the company. One of the participants (PG1) intermediate profile, whose company has now grown from 5 employees to 19, supports of this hypothesis by saying:

"... We started as a team of 5 and in situations where a team member would be unavailable maybe due to sickness... we had to reassign the responsibilities to one of the members on the team ... yes it has an effect on efficiency given that our developers use tools and processes at their comfort... it becomes difficult... because there was no money to hire many staff at that time ... but when we grew it became easier. the team is not overworked, and we make sure that our team is trained of the common practice of agile processes" (PG1)

The number of staff is crucial, affecting almost every aspect of the company. The significance of structural risk is illustrated by its severity. After financing, it is the second most mentioned dimension of the risk hypothesis. As lean business models [61], [62] become more prevalent, and companies must be more aware of their inputs and contingencies if they are delayed or lost. The SSCs should establish business continuity plans. Researchers [63], [33] suggest the need for contingency planning to settle human factors that impact SSCs. They propose solutions including assigning roles to respond quickly, minimizing interruptions, protecting the customers and ensuring business continuity.

B. HYPOTHESIS 2. LOW, COMPETITIVE ADVANTAGE INCREASES THE PERCEIVED DIFFICULTY IN PROCESS ADOPTION IN THE SMALLER CATEGORY OF THE SSCs COMPARED TO THOSE IN THE LARGER CATEGORY

This hypothesis has 2 dimensions to it: cost and differentiation advantage. If any of these are low, process adoption will be difficult for the SSCs of the initial profile that are more vulnerable than those in the intermediate and pivoting profiles. A software team's ability to produce quality software at a reduced cost in the SSCs mostly depends on human capital. Software development is human-intensive, although an organization should be able to put in place mechanisms to ensure quality and cost optimization, which unfortunately is not the case with SSCs. The struggle to put up a good team that can adjust to overcome the challenges of the volatile

software market is familiar to SSCs. It is not very easy for a company to succeed despite these challenges.

Participant (PT3) in the initial profile decried “*it is so challenging it is to deliver software products with the same benefits at a lower cost as customers demand ... it drains, and you begin to feel like it's not worth it*”. While another participant (PG5) from SSCs in the pivoting profile said:

“... it was very challenging to grow because our customers could not simply pay for the products, and we had to take up the projects to survive ... until we decided to change our business to provide services, then we managed to employ more staff and expand our services from fintech to Edu-tech services (PG5)

Decisions making and general growth of a company will depend on the company's ability to compete in the market, and cost advantage is a very significant factor. This means SSCs in the initial profile are left with no choice but to focus on revenue generation rather than cost reduction, which affects the ability to attain cost advantage, which is significant for the growth of SSCs. Unfortunately, most times at the expense of taking shortcuts by omitting processes to meet revenue targets.

Some participants believe that their challenges in attaining competitive advantage go beyond cost advantage. In the discussion, one participant expressed the desire to offer benefits beyond its competitors in the market, which depends on the context in which they operate. Positioning the companies to create superior value for customers and outstanding profits [55]. Without internal development of these approaches, they will not provide a sustained competitive advantage as competitors may counter their actions with similar ones. Regarding this, one of the participants disclosed that:

“... we started in the city, and we had clients coming from rural areas we thought we would make it easier for them... by moving to the rural areas, but when we were near, the clients wanted to pay less for the same work ... understanding value is a challenge with the rural clients so in such cases we transform our processes to enable them to have a better understanding and sometimes try to make the product cheaper” (PT2)

Using the people within an organization to create a benefit is one of the most overlooked methods in business today. In most successful SSCs, contemporaries with common backgrounds try to mold people to fit the business rather than create a business model that includes the strengths and weaknesses of its people. Most business strategies aim to achieve a sustainable competitive advantage [27], so practitioners should be keen while choosing methods and tools that give the company flexibility to attain the required competitive advantage.

C. HYPOTHESIS 3. REDUCED RESILIENCE INCREASES THE PERCEIVED DIFFICULTY IN PROCESS ADOPTION IN THE SMALLER CATEGORY OF SSCs COMPARED TO THOSE IN THE LARGER CATEGORY

Hypothesis 3 predicts how SSCs react to process adoption, given their resilience to the volatility of the context in which they operate. Participants disclose that their processes are

affected by context, including customers, competitors, and how the market is regulated. Although the SSCs try to change the market structure by adapting to the market situation changes to survive. The SSCs need to be helped to adapt by examining how the adaptation inflicts damage to a company and the products they produce, isolating the negative traits, and altering its attitude to the market to moderate them. This hypothesis has 3 dimensions: the market structure, competitive structure, and regulatory/policy influences.

A typical view of the participants is that SSCs need to build the required resilience to overcome the challenges brought by the volatility of the market structure. Although most times, the operations are at low costs, with customers equally interested in cheaper products. The difficulty in producing quality products to customers' satisfaction and remaining in business becomes visible when a company targets an out-of-the-way market and an erroneous market segment. The companies will have to compromise on effective use of processes to lower costs in the names of flexibility. Smaller companies get trapped in situations where they take up practices with shortcuts. The situation may be different with the bigger companies since in most cases, they already have established clientele and a solid financial capacity for such risks. One of the participants said:

“... it seemed easier to enter the market while targeting the low hanging fruits... at that time our customers wanted low-cost products ... it was because of the type of clients. We decided to stop making client products and start on our own products. this changed everything, and we had to employ more staff train our staff certify for standards and or processes became a critical issue” (PG2)

The essential elements of market structure are size and complexity. An example of how the elements of market structure interplay with the operations of the SSCs is seen in how urban markets tend to be large and diverse, while rural markets are often smaller and more homogeneous. This situation was experienced by a participant (PT2) who shared how fitting into a remote market was challenging. Market structures may offer different opportunities to the SSCs, depending on how prepared they are to face the challenges. The smaller categories usually are ill-prepared, and the bigger companies may concentrate their marketing efforts effectively as they avoid wasting resources, as suggested by [64]. This is echoed by a participant who recounts their journey to where they are now:

“... at the start, we had to make adjustments because the market was not easy, different clients wanted different things and because we prepared our team with a thinking of developing products for clients who were not always there, we could not let the clients interested in other products go ... when we managed to increase the team we seemed better prepared and we are not worried anymore.” (PF3)

Software demand and market trends shape the market's competitive structure, and its alterations can have substantial positive or negative effects depending on how established the SSCs are. It is typically unusual for the smaller companies in the starting face to have the resilience to see them through meeting the market's demand and competing favourably.

For this reason, the smaller companies try to create all shortcuts to fit the situation. A business can achieve higher prices and operate more profitably in markets with high barriers to entry and few competitors [65]. Additionally, it is sometimes difficult because of the customer's limited understanding of software and quality issues. How a company reacts to government policy and regulations shapes the company's resilience. To achieve this, the company's business objectives and operations must align with government regulations and policies a case in point may involve a company strategically targeting incentives provided by government with aim of promoting SME growth.

D. HYPOTHESIS 4. LOW INNOVATIVE CAPACITY INCREASES THE PERCEIVED DIFFICULTY IN PROCESS ADOPTION IN THE SMALLER CATEGORY OF SSCs COMPARED TO THOSE IN THE LARGER CATEGORY

Hypothesis 4 explains how process adoption is hindered when a company has limited innovative capacity. Innovative capacity has 2 dimensions to it: organizational learning and knowledge absorption capacity of the company. According to the participants, both dimensions affect the smaller companies most, with some participants in the category of larger companies expressing dissatisfaction with the extent of process adoption when they had few staff. One of the participants revealed that organizational learning and innovation inhibitors are critical for company performance and learning abilities which are critical for organization's growth and development.

The challenges of knowledge acquisition are more prominent in a company with less staff, and this is amplified by the structural challenges associated to learning, especially when it comes to new processes and process improvement. The smaller companies are not even up to the basics of learning from mistakes, given that there is insufficient documentation that could facilitate learning. One participant from the initial profile fronts an argument on what they feel about documentation.

“...why do we need the documents?... it is considered a waste of time which we do not have.... after all the clients are not interested in the documents anyway...” (PG4)

The other dimension of this hypothesis is knowledge absorption capacity; in this case, the companies are expected to recognize and identify value in information to benefit the organization. The processes for absorbing external knowledge become essential for innovation in firms and adapting to changes in the competitive environment.

E. HYPOTHESIS 5. LIMITED MANAGEMENT ABILITY INCREASES THE PERCEIVED DIFFICULTY IN PROCESS ADOPTION IN THE SMALLER CATEGORY SSCs COMPARED TO THOSE IN THE LARGER CATEGORY

Hypothesis 5 explains the importance of the strong ability of management in helping SSCs attain effective process adoption. This hypothesis has 4 dimensions: organizational processes, planning and organizing, leading, controlling and

the extent of flexibility in the organization. Reinforced by attributes that define how to implement management in the SSCs. The smaller companies have insufficiencies that limit management's ability to support the different attributes of management in enabling the organization to attain its expectations.

Most participants of the smaller companies tend to rely on unpredictable, unplanned, ad hoc, or complex and unstructured activities in SSCs. For instance, the decision to enter a new market is an unstructured business decision. Other ad hoc processes include gathering information, collaborating, negotiating with others, and making decisions. Having structured processes motivates incredible benefits for businesses; it also lifts the ability of management to ensure that processes are followed. Being organized transforms several things in a company. This is often complicated for the smaller companies because it is often a struggle to put up the required structures, let alone follow the processes. One of the participants from the initial profile hinted to this.

“Since we are just a few of us, most times we knew what to do because what is important to us is to produce a working software we don't need to be supervised all the time,..... why really.....I would think ... it creates unnecessary tension we are better working with the freedom.....” (PG3)

Informal management is when a company's management processes are not clearly defined, documented, or resolutely managed. The question is whether failure to formalize processes is enough to affect an organization? While others think it might not be time well spent for SSCs with a few employees to create formal, on the contrary, formalized processes are suitable for SSCs. However, the SSCs need to formalize differently. Consider that all the earlier processes were by word of mouth and not part of a defined framework. This would lead to unclear and overlapping requirements, making it impossible for the team to know what they are being measured against.

Another dimension that defines management ability is the ability to plan for the organization. However, the structure of SSCs makes it difficult to have sufficient planning and organizing because of ad-hoc practices and the pseudo confidence that come, especially with small teams on the assumption that everybody knows what to do. Because of their numbers, participants in smaller companies also believe that the planning process is a luxury. One of the participants who has gone through experience in both initial and intermediate profiles shares his thoughts about this.

“... we started as a very small team planning the processes to use was a little difficult, especially to deliberately plan our activities when we are under pressure to deliver products in a short time.....but overtime this had to change because we increased the number of staff it became easier to take up more projects and easier to use different processes... at this point we had no choice but to plan what processes.” (PT1)

Leadership is a management dimension responsible for the skills for influencing individuals to achieve a specific goal; it is the most well-thought-out and significant managerial attribute. Leaders are responsible for creating a positive

attitude towards work; they change employees' conduct to improve effectiveness and efficiency. Leadership motivates or instigates the team to work together in the company's interests. Motivating a small team may be challenging, especially since, in most cases the team get over-exhausted with extra work. Leadership involves several activities and processes, including directing, motivating, communicating, and coordinating the company's process system [30].

The participants from the small companies believe that controlling is essential as a dimension of management ability because it enables monitoring organizational progress in achieving set goals. However, most participants believe that control may not be required with a small structure. Control mechanisms should be sufficient for organizations to measure, compare, find deviations, and order to correct processes in the organization. As a result, results are controlled by monitoring how people act. When results differ from the plan, engaging those in charge is essential. As one of the participants from a SSC of intermediate profile category expressed

“... Planning without control is pointless, it is from controlling that we get to know that the plan has been successful. we did not care much about this, but we noticed how important it was when we started doing it.” (PF1)

A company needs flexibility from time to time, although it has both advantages and disadvantages. Smaller and more entrepreneurial firms can benefit from more flexible administrative structures and processes when responding to changes in the business environment [43]. On the other hand, it can have drawbacks that adversely affect an organization if unchecked. Management should look for such situations and take steps to prevent the adverse effects on the company's smooth operation.

V. RELATED WORK

Recent literature in software practice has indicated a growing interest in context and structuralism of software companies especially SSCs. Among the issues of concern are a concise definition of what describes software companies among others. The challenge has been voiced by different researchers like Sánchez-Gordón *et al.* [66]. The authors refer to “small” and “very small” enterprises as a challenging concept, as there is no commonly accepted definition of the terms. Richardson and von Wangenheim [26] also add that considering a sizable percentage of SSCs across the globe and that researchers often consider small organizations together with medium enterprises commonly referred to as SMEs without differentiating their specific characteristics can affect research results.

Additionally, the researchers appreciate the number of employees is the common denominator in all the definitions of companies perhaps because software engineering is a people intensive process and, in that respect, most processes are structured such that the people take a central role. This gives a unique context for the SSCs and that an attempt to solve the challenges of SSCs requires an additional input in identifying specific characteristics related to the

challenges brought about by the number of employees [59]. The literature recognizes this fact, and to this effect and the extent of software companies' characterization have been done to avert the limitations faced during SPI. Researchers Claudia *et al.* [45] focus on SME, McFall *et al.* [67] review SPI for all software companies. In contrast, Richardson [68], Richardson and von Wangenheim [26] focus on SSCs to mitigate foreseen SPI failures in both cases.

Addressing the limitations of implementing SPI in software organizations, McFall *et al.* [67], note that the priorities and concerns for organizations with fewer than 20 employees differ from larger ones. The authors further argue that not all software companies are the same and vary according to size, market sector, time in business, management style, product range, and geographical location. Additionally, Richardson and von Wangenheim [26] addressed similar challenges, although in the context of SSCs, stating that these companies often require different approaches because of specific business models and goals, resource availability (financial and human), process and management capability, organizational differences, among other things. Such situations have led to further complexity. On a similar note, Richardson [68], in longitudinal action research conducted in Ireland, argues that there has been little discussion about the characteristics that should be included in SPI models to make them useful for the SSCs. In his paper, the author proposes an SPI model for use in SSCs accompanied by eight characteristics that should be looked out for in the model, and the author further presents the justification for their successful inclusion. The author outlines the company's business goals, critical software processes, value for money, the maximum effect of improvement in as short a time, return on investment, process orientation, other software models and flexibility for ease of use. Our study recognizes the characteristics are indeed significant for process improvement and bridged this gap by proposing the characteristics and developing them into hypothesis to explain process adoption and the complexities that arise out of the context in which SSCs operate.

Claudia *et al.* [45] present characteristics of SMEs in a systematic review protocol that was implemented 6441 research papers from 4 software engineering databases, 28 papers are finally selected to establish a characterization of SMEs. Although the researcher's motivation in characterizing SMEs was the need for a well-defined SPI, the researchers further categorized SMEs into micro, small, and median using the number of staff. The respective categories are 1 to 9, 10 to 50 and 51 to 130. The researchers justify their categorization, stating that the term SME is used to classify the companies according to the number of workers, which differs depending on the country region. Moreover, up to 99% use, such categorization with less than 50 employees and 98% have less than 20 employees in Ireland and Australia, respectively, [45]. This implies that most SMEs are SSCs, although if not exclusively defined as SSCs, it accommodates ambiguity, as highlighted by Tuape *et al.* in two separate studies [59], [22].

TABLE 2. Comparison matrix for characteristics or features used to characterize Large Software organizations, SME and SSCs (A) Claudia [45], (B) Sánchez-Gordón [66], (C) Richardson [70], (D) Richardson [26].

No	Characteristics (Features)	A	B	C	D
1	Company's business goals,			x	x
2	Financial Resources	x			x
3	Flexibility for ease of use			x	
4	Geographical location		x		
5	Human Resources	x			x
6	Important software processes			x	
7	Management style		x		x
8	Market sector		x		
9	Maximum effect of improvement			x	
10	Models and/or Standards	x		x	x
11	Organization	x			x
12	Processes	x		x	x
13	Product range		x		
14	Projects	x			
15	Return on investment			x	
16	Size		x		
17	Time in business		x		
18	Value for money			x	

Adopting the SPI Model to achieve the SME's goals and vision is challenging. The authors describe SMEs based on their limitations in implementing SPI. Therefore, they claim that this characterization can help identify the specific problems of SMEs to enable the organization to select an appropriate implementation process improvement for effective software development. The authors list a high dependence on customers, a lack of knowledge about improving product quality as organizational limitations, a lack of economic resources to invest in process improvement, and a high dependence on external support as financial limitations. The authors add lack of personnel due to the minimal number of employees, lack of roles definition, so employees perform several functions, lack of knowledge about process as human resource limitations: undefined processes and high cost of SPI implementation.

Additionally, working with small-size projects, process implementation takes a long time, process implementation is based on customer's requirements as project-related limitations, and the models and standards limitations are bad experiences in the adoption of SPI models, SME is adapted to the selected model that will be applied. In all the cases reviewed as studies related to our study, we present a comparison of the SPI limitation characteristics in Table 2. With 18 features identified, there are more similarities in the presentations of Richardson [68] and Richardson [26]. Our work differs because we focus on the context of SSCs and also, we propose our characteristics to complement number of employees in affecting process adoption in SSCs. We have also gone further by developing hypothesis for the prediction and explanation of process adoption using these characteristics.

In a nutshell, the hypotheses proposed in this work furthers the understanding of the challenges of process adoption in

SSCs, advancing deeper insights beyond process improvement. This does not mean that SPI is not essential; instead, it offers an opportunity to focus on solving the challenges of limited usage of the processes that require a better understanding of why SSCs fail to adopt the available methods, procedures, and strategies in the first place.

VI. CONCLUSION AND FUTURE WORK

A. CONCLUSION

In the introduction, we mentioned that process adoption is a very significant to producing quality software products in software practice, especially for SSCs, because they produce the most software products in the market. This challenge is unfortunately minimally addressed in the software engineering literature. This paper began by investigating structuralism (number of employees) as a critical challenge affecting process adoption in software practice in SSCs. The investigation was done by conducting 18 interviews through a GT. 5 non-technical characteristics related to the number of employees significant for the theoretical advancement to predict and explain software process adoption in SSCs emerged.

This paper emphasizes that the non-technical characteristics (risk, competitive advantage, resilience, innovative capacity, and management ability) with their respective dimensions complement a number of employees as antecedents in affecting process adoption in SSCs. As a result of the GT, these characteristics become complex due to the challenge of the number of employees, and they affect practice in a complementary manner. Software processes are significantly affected in SSCs because they employ less than 50 employees. Companies with fewer employees tend to have unstable processes and practices compared to their larger counterparts. We attempted to demonstrate this by profiling the SSCs in 3 different profiles, and our findings indicated that, indeed, those in the profile with fewer employees experienced severer effects of the identified characteristics in affecting processes.

On the other hand, some companies in the profile with more staff reveal how they shared similar experiences while still in the profile with fewer staff. It is also important to note that the characteristics may seem associated with all software companies irrespective of a number of employees, however, we emphasize the severity of effects on the SSCs. Additionally, the larger companies have mechanisms for mitigating the challenges that arise out of these characteristics, as emphasized in literature [1], [22], [69].

There are important lessons for software engineering practitioners and researchers who should be aware that although the number of employees is significant in defining what SSCs are, it is paramount to go beyond the thinking that the number of staff will define everything Sánchez-Gordón *et al.* [66]. To solve the challenges SSCs in practice, researchers and practitioners should pay attention to the non-technical characteristics that are influenced by the number of employees and are responsible for the challenges and the operations of SSCs.

A case in point is that the company's risk exposure would make it challenging to implement SPI. Tools methods and processes introduced disregarding the management's ability to lead and motivate the team to implement a new tool, let alone control the effect of introducing the same tool, stand a risk of being unadoptable to the SSCs because they do not fit into the context [3]. The introduction of some processes may affect the operation of SSCs, and the required resilience may be necessary to consider. Else the conditions around the SSCs would change, affecting normal operations and business in general. This calls for the need to look at the inherent characteristics carefully lest the same mistakes are made by solving symptoms and not the holistic problem of what to consider when making decisions on the tools, processes, and methods for SSCs.

The contribution of this paper should be interpreted in consideration of these several limitations: firstly, we limited this study to interviewing 18 participants from 4 countries that is Finland, Tanzania, Namibia, and Ghana. We believe these countries represent a reasonable surrogate of the divergent context of the SSCs of both the mature and immature setup. Although the 18 participants having different roles may portray divergence to affect the validity of the study, however the participants who described their roles as project managers were also significantly involved in software development activity and their experience in software development contributed to the understanding of what actually occurs in the SSCs. Secondly, interviews may compromise the objectivist views that were expected to guide this study to ensure a value-free outcome. However, the data collection which occurred during the COVID time could not have the expected level of independent observations. So, the interviews were designed to give us the best possibility of value-free findings by ensuring that the researchers avoid influencing the participant's thoughts as much as possible.

Additionally, purposive (multi-variant) sampling could have given room for an element of sampling bias. However, we tried to be strict about a standard basic threshold on what type of company the participant was expected to come from. In the plight of these limitations, further theory building and testing in other areas using different methods would be required to add building blocks to this work.

VII. FUTURE WORK

Our next course of action is to undertake a detailed study; (1) to establish interdependencies of the proposed non-technical characteristics (2) to develop a classification taxonomy based on the hypotheses from this work. We intend to attach a metric so that this taxonomy will be used to evaluate companies and determine a specific class with scores. Based on this, informed choices of appropriate software development processes, tools, or methods can be made by the SSCs.

ACKNOWLEDGMENT

The authors sincerely thank all the participants who offered their valuable time to participate in the data collection

interviews. They also extend our gratitude to the editors and reviewers for their valuable time and honest rebuttal to improve this paper.

REFERENCES

- [1] M. Tuape and Y. Ayalew, "Factors affecting development process in small software companies," in *Proc. IEEE/ACM Symp. Softw. Eng. Afr. (SEiA)*, May 2019, pp. 16–23, doi: [10.1109/SEiA.2019.00011](https://doi.org/10.1109/SEiA.2019.00011).
- [2] C. G. von Wangenheim, A. Anacleto, and C. F. Salviano, "Helping small companies assess software processes," *IEEE Softw.*, vol. 23, no. 1, pp. 91–98, Jan. 2006, doi: [10.1109/MS.2006.13](https://doi.org/10.1109/MS.2006.13).
- [3] N. Tripathi, E. Annanperä, M. Oivo, and K. Liukkunen, "Exploring processes in small software companies: A systematic review," in *Software Process Improvement and Capability Determination (Communications in Computer and Information Science)*, vol. 609. Cham, Switzerland: Springer, 2016, pp. 150–165, doi: [10.1007/978-3-319-38980-6_12](https://doi.org/10.1007/978-3-319-38980-6_12).
- [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 *Proc. 4th Int. SPICE Conf. Process Assessment Improvement*, Lisbon, Portugal, 2004, pp. 33–37.
- [5] European Commission. (2020). *User Guide to the SME Definition*. Publications Office of the European Union. Accessed: May 30, 2021. [Online]. Available: <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2003:124:0036:0041:en:PDF>
- [6] B. G. Glaser, *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory 1978*. New York, NY, USA: Sociology Press, 1967.
- [7] B. G. Glaser and A. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. New York, NY, USA: Aldine de Gruyter, 1967.
- [8] M. Redman-MacLaren and J. Mills, "Transformational grounded theory: Theory, voice, and action," *Int. J. Qualitative Methods*, vol. 14, no. 3, pp. 1–12, 2015, doi: [10.1177/160940691501400301](https://doi.org/10.1177/160940691501400301).
- [9] J. D. Wolfe, B. Shawn, and C. L. Cain, "Research methods in medical sociology," in *Wiley Blackwell Companion to Medical Sociology*. 2021, pp. 45–61.
- [10] K. Bahari, A. T. Talosig, and J. B. Pizarro, "Nursing technologies creativity as an expression of caring: A grounded theory study," *Global Qualitative Nursing Res.*, vol. 8, Feb. 2021, Art. no. 2333393621997397.
- [11] M. Bianchi, L. Ghirrotto, and A. Bagnasco, "Tutor and student dyadic interactions in relation to interprofessional education and clinical care: A constructivist grounded theory study," *J. Adv. Nursing*, vol. 77, no. 2, pp. 922–933, 2021.
- [12] P. Zhu, M. Luke, and J. Bellini, "A grounded theory analysis of cultural humility in counseling and counselor education," *Counselor Educ. Supervision*, vol. 60, no. 1, pp. 73–89, Mar. 2021.
- [13] P. Goyal, S. Bhattacharya, and A. Gandhi, "Grounded theory in management research: Through the lens of gender-based pay disparity," *J. Adv. Manage. Res.*, vol. 19, no. 1, pp. 12–30, Jan. 2022.
- [14] F. Rahimi-Feyzabad, M. Yazdanpanah, S. Gholamrezai, and M. Ahmadvand, "Institutional constraints to groundwater resource management in arid and semi-arid regions: A straussian grounded theory study," *Hydrogeol. J.*, vol. 29, no. 3, pp. 925–947, May 2021.
- [15] R. Suddaby, "From the editors: What grounded theory is not," *Acad. Manage. J.*, vol. 49, no. 4, pp. 633–642, Aug. 2006.
- [16] S. Bragagnolo, N. Anquetil, S. Ducasse, A. Seriai, and M. Derras, "Software migration: A theoretical framework (a grounded theory approach on systematic literature review)," *Empirical Softw. Eng.*, 2021.
- [17] G. Coleman and R. O'Connor, "Investigating software process in practice: A grounded theory perspective," *J. Syst. Softw.*, vol. 81, no. 5, pp. 772–784, May 2008.
- [18] J. Kasurinen, O. Taipale, and K. Smolander, "Analysis of problems in testing practices," in *Proc. 16th Asia-Pacific Softw. Eng. Conf.*, Dec. 2009, pp. 309–315, doi: [10.1109/APSEC.2009.17](https://doi.org/10.1109/APSEC.2009.17).
- [19] Z. Masood, R. Hoda, and K. Blincoc, "Real world scrum a grounded theory of variations in practice," *IEEE Trans. Softw. Eng.*, vol. 48, no. 5, pp. 1579–1591, May 2022.
- [20] K.-J. Stol, P. Ralph, and B. Fitzgerald, "Grounded theory in software engineering research: A critical review and guidelines," in *Proc. 38th Int. Conf. Softw. Eng.*, May 2016, pp. 120–131, doi: [10.1145/2884781.2884833](https://doi.org/10.1145/2884781.2884833).
- [21] B. G. Glaser, "Advances in the methodology of grounded theory: Theoretical sensitivity," Univ. California, Mill Valley, CA, USA, 1978.
- [22] M. Tuape, V. T. 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*, vol. 9, pp. 130366–130388, 2021.

- [23] D. J. Hoch, C. Roeding, S. K. Lindner, and G. Purkert, *Secrets of Software Success*. Boston, MA, USA: Harvard Business School Press, 2000.
- [24] B. G. Glaser, *Basics of Grounded Theory Analysis: Emergence vs Forcing*. Mill Valley, CA, USA: Sociology Press, 1992.
- [25] R. Hoda, J. Noble, and S. Marshall, "The impact of inadequate customer collaboration on self-organizing agile teams," *Inf. Softw. Technol.*, vol. 53, no. 5, pp. 521–534, May 2011.
- [26] I. Richardson and C. G. Von Wangenheim, "Guest editors' introduction: Why are small software organizations different?" *IEEE Softw.*, vol. 24, no. 1, pp. 18–22, Jan. 2007.
- [27] 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.
- [28] J. Keung, R. Jeffery, and B. Kitchenham, "The challenge of introducing a new software cost estimation technology into a small software organisation," in *Proc. Austral. Softw. Eng. Conf.*, Melbourne, NSW, Australia, 2004, pp. 52–59, doi: [10.1109/ASWEC.2004.1290457](https://doi.org/10.1109/ASWEC.2004.1290457).
- [29] A. Ajitabh and K. Momaya, "Competitiveness of firms: Review of theory, frameworks and models," *Singapore Manage. Rev.*, vol. 26, no. 1, pp. 45–61, 2004.
- [30] T. Dybå, "Factors of software process improvement success in small and large organizations: An empirical study in the Scandinavian context," in *Proc. 9th Eur. Softw. Eng. Conf. Held Jointly 10th ACM SIGSOFT Int. Symp. Found. Softw. Eng. (ESEC/FSE)*, Helsinki, Finland, 2003, pp. 148–157, doi: [10.1145/940071.940092](https://doi.org/10.1145/940071.940092).
- [31] R. E. Miles, C. C. Snow, A. D. Meyer, and H. J. Coleman, Jr., "Organizational strategy, structure, and process," *Acad. Manage. Rev.*, vol. 3, no. 3, pp. 546–562, 1978.
- [32] I. Allison, "Organizational factors shaping software process improvement in small-medium sized software teams: A multi-case analysis," in *Proc. 7th Int. Conf. Quality Inf. Commun. Technol.*, Porto, Portugal, Sep. 2010, pp. 418–423, doi: [10.1109/QUATIC.2010.81](https://doi.org/10.1109/QUATIC.2010.81).
- [33] M. A. T. Almomani, S. Basri, A. K. B. Mahmood, and A. O. Bajeh, "Software development practices and problems in Malaysian small and medium software enterprises: A pilot study," in *Proc. 5th Int. Conf. IT Conver. Secur. (ICITCS)*, Aug. 2015, pp. 1–5.
- [34] S. B. Basri and R. V. O'Connor, "Organizational commitment towards software process improvement an Irish software VSEs case study," in *Proc. Int. Symp. Inf. Technol.*, vol. 3, Jun. 2010, pp. 1456–1461.
- [35] C. Y. Laporte, S. Alexandre, and R. O'Connor, "A software engineering lifecycle standard for very small enterprises," in *Proc. 15th Eur. Conf. Softw. Process Improvement (EuroSPI)*, vol. 16, Dublin, Ireland: Springer, Sep. 2008, pp. 129–141, doi: [10.1007/978-3-540-85936-9_12](https://doi.org/10.1007/978-3-540-85936-9_12).
- [36] C. Y. Laporte and R. V. O'Connor, "Implementing process improvement in very small enterprises with ISO/IEC 29110: A multiple case study analysis," in *Proc. 10th Int. Conf. Quality Inf. Commun. Technol. (QUATIC)*, Sep. 2016, pp. 125–130.
- [37] M. Harris, K. Aebischer, and T. Klaus, "The whitewater process: Software product development in small IT businesses," *Commun. ACM*, vol. 50, no. 5, pp. 89–93, May 2007, doi: [10.1145/1230819.1241669](https://doi.org/10.1145/1230819.1241669).
- [38] I. Keshta, M. Niazi, and M. Alshayeb, "Towards implementation of process and product quality assurance process area for Saudi Arabian small and medium sized software development organizations," *IEEE Access*, vol. 6, pp. 41643–41675, 2018, doi: [10.1109/ACCESS.2018.2859249](https://doi.org/10.1109/ACCESS.2018.2859249).
- [39] M. Lyles and M. P. V. Easterby-Smith, "Introduction: Watersheds of organizational learning and knowledge management," in *The Blackwell Handbook of Organizational Learning and Knowledge Management*. Oxford, U.K.: Blackwell, 2003.
- [40] O. Ibert, "Projects and firms as discordant complements: Organisational learning in the Munich software ecology," *Res. Policy*, vol. 33, no. 10, pp. 1529–1546, Dec. 2004.
- [41] J. Gasston and P. Halloran, "Continuous software process improvement requires organisational learning: An Australian case study," *Softw. Qual. J.*, vol. 8, no. 1, pp. 37–51, 1999.
- [42] M. Ruokonen, N. Nummela, K. Puumalainen, and S. Saarenketo, "Market orientation and internationalisation in small software firms," *Eur. J. Marketing*, vol. 42, nos. 11–12, pp. 1294–1315, Nov. 2008.
- [43] S. Latham, "Contrasting strategic response to economic recession in start-up versus established software firms," *J. Small Bus. Manage.*, vol. 47, no. 2, pp. 180–201, 2009.
- [44] D. A. M. Aguillar, I. Murakami, P. M. Junior, and P. T. Aquino, "IT governance program and improvements in Brazilian small business: Viability and case study," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2017, pp. 961–964.
- [45] V. Claudia, M. Mirna, and M. Jezreel, "Characterization of software processes improvement needs in SMEs," in *Proc. Int. Conf. Mechatronics, Electron. Automot. Eng.*, Nov. 2013, pp. 223–228.
- [46] A. Mesquida and A. Mas, "A project management improvement program according to ISO/IEC 29110 and PMBOK," *J. Softw., Evol. Process*, vol. 26, no. 9, pp. 846–854, 2014.
- [47] R. V. O'Connor and C. Y. Laporte, "Software project management in very small entities with ISO/IEC 29110," in *Proc. 19th Eur. Conf. Softw. Process Improvement (EuroSPI)*, vol. 301, Vienna, Austria: Springer, Jun. 2012, pp. 330–341, doi: [10.1007/978-3-642-31199-4_29](https://doi.org/10.1007/978-3-642-31199-4_29).
- [48] W. S. Humphrey, *A Discipline for Software Engineering*. London, U.K.: Pearson, 1995.
- [49] S. A. Zahra and G. George, "Absorptive capacity: A review, reconceptualization, and extension," *Acad. Manage. Rev.*, vol. 27, no. 2, pp. 185–203, Apr. 2002.
- [50] M. A. Montoni, A. R. Rocha, and K. C. Weber, "MPS. BR: A successful program for software process improvement in Brazil," *Softw. Process: Improvement Pract.*, vol. 14, no. 5, pp. 289–300, 2009.
- [51] R. Zurina, R. M. Ali, and S. Ibrahim, "An iSPA model evaluation based on critical success factors and selected criteria to support Malaysia's SME environment," in *Proc. 2nd Int. Conf. Softw. Eng. Data Mining*, 2010, pp. 225–230.
- [52] M. Nawazish Khokhar, A. Mansoor, M. Nadeem Khokhar, S. Ur Rehman, and A. Rauf, "MECA: Software process improvement for small organizations," in *Proc. Int. Conf. Emerg. Technol.*, Jun. 2010, pp. 1–6.
- [53] B. Barafort, A.-L. Mesquida, and A. Mas, "Integrated risk management process assessment model for IT organizations based on ISO 31000 in an ISO multi-standards context," *Comput. Standards Interfaces*, vol. 60, pp. 57–66, Nov. 2018.
- [54] C. Giardino, M. Unterkalmsteiner, N. Paternoster, T. Gorschek, and P. Abrahamsson, "What do we know about software development in startups?" *IEEE Softw.*, vol. 31, no. 5, pp. 28–32, Sep. 2014, doi: [10.1109/MS.2014.129](https://doi.org/10.1109/MS.2014.129).
- [55] M. E. Porter, "From competitive advantage to corporate strategy," in *Readings in Strategic Management*. Berlin, Germany: Springer, 1989, pp. 234–255.
- [56] D. Jiménez-Jiménez and J. G. Cegarra-Navarro, "The performance effect of organizational learning and market orientation," *Ind. Marketing Manage.*, vol. 36, no. 6, pp. 694–708, Aug. 2007.
- [57] T. Koc, "Organizational determinants of innovation capacity in software companies," *Comput. Ind. Eng.*, vol. 53, no. 3, pp. 373–385, Oct. 2007.
- [58] R. Chiva, "The learning organization and the level of consciousness," *Learn. Org.*, vol. 24, no. 3, pp. 150–158, 2017, doi: [10.1108/TLO-11-2016-0074](https://doi.org/10.1108/TLO-11-2016-0074).
- [59] 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 *Proc. Future Technol. Conf.*, 2020, pp. 341–356.
- [60] H. Kerzner, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Hoboken, NJ, USA: Wiley, 2017.
- [61] A. Ojala, "Business models and opportunity creation: How IT entrepreneurs create and develop business models under uncertainty," *Inf. Syst. J.*, vol. 26, no. 5, pp. 451–476, Sep. 2016.
- [62] M. Schief and P. Buxmann, "Business models in the software industry," in *Proc. 45th Hawaii Int. Conf. Syst. Sci.*, Jan. 2012, pp. 3328–3337.
- [63] M. A. Almomani, S. Basri, and A. R. Gilal, "Empirical study of software process improvement in Malaysian small and medium enterprises: The human aspects," *J. Softw., Evol. Process*, vol. 30, no. 10, p. e1953, Oct. 2018, doi: [10.1002/smr.1953](https://doi.org/10.1002/smr.1953).
- [64] A. Ojala, "Entry in a psychically distant market: Finnish small and medium-sized software firms in Japan," *Eur. Manage. J.*, vol. 26, no. 2, pp. 135–144, 2008.
- [65] E. Blake, "Software engineering in developing communities," in *Proc. ICSE Workshop Cooperat. Human Aspects Softw. Eng. (CHASE)*, 2010, pp. 1–4.
- [66] M.-L. Sánchez-Gordón, R. Colomo-Palacios, A. de Amescua Seco, and R. V. O'Connor, "The route to software process improvement in small- and medium-sized enterprises," in *Managing Software Process Evolution*. Cham, Switzerland: Springer, 2016, pp. 109–136.
- [67] D. McFall, F. G. Wilkie, F. McCaffery, N. Lester, and R. Sterrit, "Software processes and process improvement in Northern Ireland," in *Proc. 16th Int. Conf. Softw. Syst. Eng. Appl.*, Paris, France, Dec. 2003, pp. 1–10.
- [68] I. Richardson, "SPI models: What characteristics are required for small software development companies?" *Softw. Qual. J.*, vol. 10, no. 2, pp. 101–114, 2002, doi: [10.1023/A:1020519822806](https://doi.org/10.1023/A:1020519822806).

- [69] P. Clarke and R. V. O'Connor, "An approach to evaluating software process adaptation," in *Software Process Improvement and Capability Determination* (Computer and Information Science), vol. 155. Berlin, Germany: Springer, 2011, pp. 28–41, doi: [10.1007/978-3-642-21233-8](https://doi.org/10.1007/978-3-642-21233-8).
- [70] I. Richardson, "SPI models: What characteristics are required for small software development companies?" in *Proc. 7th Eur. Conf. Softw. Quality*, vol. 2349. Helsinki, Finland: Springer, Jun. 2002, pp. 100–113, doi: [10.1007/3-540-47984-8_14](https://doi.org/10.1007/3-540-47984-8_14).



MICHEAL TUAPE (Member, IEEE) received the B.Sc. degree (Hons.) in information technology from Uganda Christian University Mukono, Uganda, in 2010, and the M.Sc. degree in software engineering from Makerere University, Kampala, Uganda, in 2014. He is currently pursuing the D.Sc. degree in technology software engineering with the Lappeenranta-Lahti University of Technology, Lappeenranta, Finland. He is currently a Junior Researcher with the Lappeenranta-Lahti

University of Technology, since 2020, he has also been a Research Associate with the UIG Project, University of Botswana, Gaborone, Botswana, 2018–2020, a Research Associate at the Software Systems Center Makerere University, 2015–2018. His research interests include software engineering, requirements engineering, software development process small software companies, and open science. He is also a member of ACM and the PMI.



VICTORIA T. HASHEELA-MUFETI received the B.Sc. degree in computer science and economics from the University of Namibia, in 2005, and the B.Sc. degree (Hons.) in computer science from Stellenbosch University, South Africa, in 2007, and the M.Sc. degree in informatics from Mannheim University, Germany, in 2010, and the D.Sc. degree in technology from Lappeenranta University of Technology, Finland, in 2018. She was a Fulbright Scholar at the University of New Mexico, USA, in 2018. She is currently a Senior Lecturer with the University of Namibia. Her research interests include digital preservation of African indigenous knowledge and languages, software development for SMEs, and data analytics.



JUSSI KASURINEN received the master's degree in information technology, the D.Sc. degree in technology, specializing in software engineering and software testing, and the Adjunct Professor of entertainment software engineering from the Lappeenranta University of Technology (LUT), Lappeenranta, Finland, in 2007, 2011, and 2017, respectively. He is currently working as an Associate Professor and the Head of Degree Programs in software engineering with the LUT School of Engineering Sciences. During his career, he has authored over 50 scientific publications in various topics of software engineering, and four non-fiction books discussing programming languages and software testing. His current research interests include but are not limited to smart systems for software engineering, games from the viewpoint of software, software testing practices, and software process quality. He is also the current LUT University Representative of the Finnish Software Measurement Association (FiSMA), Computer Science Association of Finland, and Academic Engineers and Architects in Finland (TEK).

• • •