

# A roadmap for a comparison framework for an adaptable software process improvement framework in small software companies

Micheal Tuape  
University of Botswana  
Department of Computer Science  
Gaborone Botswana  
tuapem@ub.ac.bw

Yirsaw Ayalew  
University of Botswana  
Department of Computer Science  
Gaborone Botswana  
ayalew@mopipi.ub.bw

□ *Abstract*— Poor software quality due to failure to organize development processes using SPI is visible in small software companies, although these companies are significant to industry and world economy; the challenge of quality needs much more attention. The development of different frameworks to sort the development process in SSC continues to leave a lot to be desired. This work leads us to the development of a comparison framework which will ultimately end with an adaptable SPA and SPI framework for SSC. From a SLR to identify the factors affecting SSC and mapping them to the specific processes in which they occur, we analyze the software development environment and identify SPI frameworks that have registered success in similar environments of SSC to help us adopt best practices from which comparison is made to generate requirements for an adaptable SPI framework within context for small companies.

## I. INTRODUCTION

Most of software development establishments face teething troubles in their projects due to deficiency of utilization of best practices and standards[1-3]. According to Khokhar et al [4] 50% of all application do not encounter trade goals while 40% of software development ventures do not deliver intended return on investment due to lack of best practices. Most software process assessment (SPA) models and standards pay attention primarily on medium and large companies; this obscures the helpfulness and proficiency of small software companies (SSC) perhaps because of their particular characteristics and limitations [5-7].

The entire software engineering SE industry appreciates the value of SSCs in terms of their link in producing cherished products and services to the industry. Although software process improvement (SPI) offers SSC prospects and trials[8], the bulk of software companies in industry fall within the SSC size category. It is essential to note that for SSC to offer customers with quality products that meet client needs within time and budget, the SSC must address daily trials, which involve probing software processes to ensuring production of quality products[5, 9]. This is a dreadful task to SSC probably due to time, resources and actually it ends up not being the case.

The apparent value of a simple practice in SPI is easily done using low cost practices, that makes SPI inexpensive for SSC

with inadequate resources, facilitating them to achieve significant paybacks minus the utilization of a disproportionate quantity of resources[10]. This situation has stifled quality in SSC especially in Africa that SPI is unheard of. An organization that is interested in improving quality by undertaking SPI will require to conduct a quick and inexpensive procedure which is a nightmare for SSC, and this explains why software practitioners in such situations should figure out which SPI frameworks for the varied available framework is the most relevant. Sizeable achievement has been recorded by larger software companies leaving SSC behind hence the need to pay attention to the SSC. In this spirit, SPI has been adapted to fit in specific contexts[11, 12]. Accordingly, this work attempts to develop a roadmap for a SPI comparison framework (SPICOF) that compares frameworks used to solve trials in similar software development environment (SDE), so as to adopt improvement actions (IA) that will be applied to solve the trials in the SDE for which the framework is being developed. The outcome of this work will eventually be used to formulate requirements for an adoptable SPA framework for SSC based on the characteristic factors that affect SSC for example those from Africa.

The rest of this paper is organized as follows, Section II is related work, section III covers our approach, section IV covers what will makes our work different, section V covers our own analysis and chapter VI concludes the paper with what works will follow.

## II. RELATED WORK

### A. Comparing SPI frameworks

The Comparison of SPI frameworks is not common perhaps because of low uptake of SPI and maybe most of the frameworks that exist have been designed to suit specific SDE[13, 14]. This comparison of these frameworks is such a challenge because they are comprehensive and have particular description. In choosing SPI framework, practitioners choose frameworks as favorite based on subjective reasons. In cases of SSC we have seen frameworks developed also to suite specific SDE and cases of comparison are also limited to renowned frameworks like CMM, ISO 15504 and ISO 9000 perhaps this is explained by the fact that choosing a framework against another is

□ This work was not supported by any organization

complicated. Comparisons majorly for two different reasons;

First, authors of new frameworks compare what has already been developed with what is being authored to determine the differences and the similarities. In [1] Zarour highlights what he describes as authors point of view, in which authors of new frameworks compare with existent frameworks. For example in [6], Anacleto et al compares RAPID, SPINI, FAME, TOPS and MARES. As authors of MARES they seem to check the difference and similarity of MARES and the other 4 frameworks using requirements. In another example McCaffrey et al [15] compares their proposed assessment model for SSC to other lightweight assessment methods.

Secondly, organizations without SPI strategy or with little knowledge on SPI can prudently compare different SPI frameworks to learn about other frameworks so as to conduct self-assessment to evaluate capability levels. Zarour et al [1] and Halvorlsen et al [2] discuss organizations need to compare frameworks to facilitate choice of SPI frameworks. For instance in [13] Tingey makes a thorough comparison of CMM, ISO 15504 and the Malcom Balgride National Quality Award (MBA) to expedite companies capacity to take knowledgeable choices. We also see in [16] Paulk comparing ISO 9000 and CMM to help organization make a clear distinction on the underlying philosophies, although he also highlights the fact that they address common concerns of quality and process management (PRM).

The two scenarios have often led to the use of the four types of comparison methods, as discussed by Halvorlsen in his taxonomy [2]. In his classification, he highlights, needs mapping comparison method, framework mapping comparison method, characteristics comparison method, and bilateral comparison method. Authors [1, 2, 6, 13, 15-17] have compared SPI frameworks with the stated comparison methods argued as follows:

#### 1) Needs Mapping Comparison Method

Needs mapping is a technique which contemplates organizational and environmental requirements when picking which SPI framework to embrace. Here are some examples as cited by [2]: Certification desires, for example of ISO 9000, is regularly enforced on subcontractors; Management may oblige that the preferred SPI framework must be incorporated in a total quality management methodology. In alternative occurrences, [1] uses this technique to compare TOPS, Micro-Evaluation (OWPL), MARES, SPM, RAPID, FAME, and EAP as different frameworks. And similarly [6] compares MARES to RAPID, SPINI, TOPS, and FAME. This method is not a straight comparison between frameworks, in that requirements are considered to be of paramount significance and must be cautiously considered into since they are challenging and can utterly limit the select of framework. For the reason that evolution in organizational environment is perpetual, we see the requirements keep changing instigating variation from one organization to the other.

#### 2) Framework Mapping Comparison Method

In [13] we appreciate that every single framework consists of a conventional statements or necessities allocating the content and emphasis of the framework as deliberated by [2]. Framework mapping is the procedure of constructing a map from statements or conceptions of one framework to those of the alternative. There are two divergent ways to do this: Mapping prevailing frameworks into the suggested framework being exclusively for the resolve of comparing, this delivers a mutual argument of reference from which the frameworks can be appraised, compared and contrasted; in the characteristics method the objective is to pronounce crucial traits of each SPI framework. Nevertheless, the purpose of mapping is to identify intersections and interactions between frameworks and produce a map of these statements or relationships. There can occur solid, weak or certainly not correlation as seen in Tingey's comparison of the 3 frameworks [13]. He, maps CMM, ISO9000 and the MBA on 2 stages, high and low reliant on the quantity of fact every one encompasses. This also decrees how the mapping outcomes can be revealed, for example in a matrix or some kind of graphic illustrations like Venn diagrams.

This is principally handy when an organization engages two or more distinct SPI frameworks, as analogous statements can be recognized and redundancy eliminated. The additional exertion required to employ an extra framework can be decreased. It is definitely a more low-level and detailed comparison method than characteristics. Because mapping goes into the specifics of each framework, it is not suitable for a broad impression. On the other hand, mapping into a base framework and supplementing with a quantitative analysis can point to overall focus and content. Framework mapping necessitates that some streamlining assumptions are put into thought. The outcomes are unavoidably prejudiced by these expectations as observed in [13].

#### 3) Characteristics Comparison Method

Grounded on a far-reaching literature inspection, [2] and [1] compare various frameworks in a taxonomy by defining an extensive list of relevant characteristics. Each framework is then described in terms of these characteristics and the outcome is presented in a tabular format. This type of comparison is well suited for a general overview of the frameworks and it can be used as a foundation for other comparison methods. A lot of information can be concluded from a table listing the characteristics of a number of frameworks. However, the comparison is on a high level and details must be collected elsewhere. For this purpose one of the other comparison methods can be used.

In [1] we see five frameworks of RAPID, SPINI, FAME, TOPS and their new method MARES compared by Anacleto et al. the characteristics listed are objective, measurable and comparable. Although Halvorsen [2], also uses this framework in his taxonomy he argues that being objective is almost unmanageable in the firm sense, determinate is often a matter of human ruling, and comparable means that a practitioner will benefit from the comparison. He claims that a kind of dimension scale should



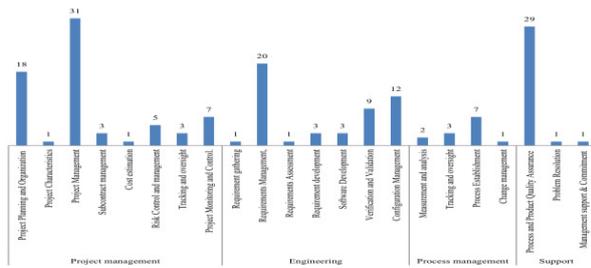


Fig 3. Graph representing frequencies of factors affecting SSC in specific processes

The 69 frameworks used for SSC identified from the SLR give us a picture of how much work has been done despite the continued slow pace of improvement in quality of software especially in SSC.

### B. Analysis of different frameworks and how they are used in different environment

Based on the analysis of the SPI frameworks for SSC we single out 7 based their characteristics and the development environment from which they operate i.e. Moprosoft, Mares, iSPA, RAPID, SPINI, MPS-BR and MECA see table III. We study the activities involved the improvement process of each of the identified SPI frameworks and discuss them as follows;

#### 1) MoProSoft

This framework was established in Mexico together with the UNAM, the Mexican Association for the Quality in Software Engineering (AMCIS) and the Ministry of Economy[20]. The drive of the manuscript was to present a Software Industry Processes Model (MoProSoft) in Mexico supporting the regularization of its procedure through the integration of best practices in software management and Eng.

Adopting the model was meant to permit nurturing the capability of organizations to offer high quality services and reach international level of competitiveness.

It has 6 processes; business management; PRM; PM; resource management; administration of definite assignments and Software development centered on the ISO/IEC/12207, CMM, ISO9001[19]

#### 2) The MARES Process Assessment Model

Brazilian software industry has grown at a fast rate yet playing a pivotal role in the economy. However, the SSC were face quality and productivity problems caused by organizational and administrative deficiencies, which affected their competitiveness.

MARES as a methodology for SPA in SSC was designed to support software process considering specific characteristics and limitations[7, 20]. SSC in Brazil found it particularly difficult to run assessments in conformance with international standards or models, such as, ISO/IEC 15504, CMMI for software or ISO 9001 (including 9000-3) due the major emphasis of these models and standards on medium or

large companies. Nevertheless, characteristics and confines typical for SSC in conformance with ISO/IEC 15504 by integrating context –process model in order to support the selection of relevant processes and process-risk model to support the identification of potential risks and improvement suggestions [21].

The MARES process assessment framework was also enhanced by the description of a context-process relationship model, which mimics the association between specific characteristics known difficulties and business objectives[6]. It was assembled by researchers from the UNIVALI University in collaboration with CenPra research center in Brazil[22].

MARES is divided into five core parts which are planning, contextualization, execution, monitoring and control and post-mortem[6]. Methodology for process assessment in SSC entail a model which is based on the exemplar model of part 5 of ISO/IEC 15504, [20] an assessment process that encounters the requirements of the assessment process defined in ISO/IEC 15504-2, an assessor accreditation method which outlines procedures and an assessment monitoring method which enables the constant monitoring of assessment methodology. [7, 22]

#### 3) iSPA Integrated Software Process Assessment

The Malaysian software industry was facing difficulties in their software projects due to lack of putting into practice the best practices and standards. According [23] 50% of all application are unsuccessful and did not achieve business goals whereas 40% of software projects failed to deliver intended return on investment due to lack of best practices. Among the causes of lack of best practices was level of awareness and understanding amongst software development companies. There were also issues of costs involved in the direction of the execution of standards or benchmark requirements which had become a significant deterrent factors [24]. This gave rise to the development of the SPA framework iSPI as a starting point from the ISO/IEC 15504 (Information Technology – Process Assessment) as an all-inclusive standards or SPA [25]. ISO/IEC 15504 is seen as an internationally recognized standard by ISO organization, and also embraced as a Malaysian standard. The framework has 3 layers to the system which are presentation, application and data layers with 6 processes Project/Process profiling gap analysis; Strength/Weakness generator; Improvement plan; Customization of PRM and Reporting[25].

#### 4) RAPID (Rapid Assessment or Process Improvement for software Development)

This is a light weight ISO/IEC 15504 or SPICE complaint assessment methodology with a range of 8 processes[26]. Requirement elicitation, software development, configuration management, quality assurance, problem resolution, PM, risk management, and process establishment. This methodology has been preferred for evaluation to its semblance with SPICE[27] probably because of its addition of threat management process within its activities.

TABLE II.  
SELECTED SPI FRAMEWORKS AND CORRESPONDING PROCESSES ADDRESSED DURING ASSESSMENT AND IMPROVEMENT

software process areas	Software processes	CMMI	RAPID	iSPA	MARES	SPINI	MOPROSOFT	MPS BR	MECA
Project management	Project Planning and Organization	x		x	x	No specific number of processes		x	No specific number of processes
	Project Characteristics	x					x	x	
	Project Management	x	x				x	x	
	Subcontract management	x						x	
	Cost estimation	x						x	
	Risk Control and management	x	x	x				X	
	Tracking and oversight	x						X	
	Project Monitoring and Control.	x		x	x		x	X	
Engineering	Requirement gathering	x	x					X	
	Requirements Management,	x		x				X	
	Requirements Assessment	x			x			X	
	Requirement development	x						X	
	Software Development	x	x		x		x	x	
	Verification and Validation	x						x	
	Configuration Management	x	x					x	
Process management	Measurement and analysis	x		x				x	
	Tracking and oversight	x			x		x		
	Process Establishment	x	x			x	x		
	Change management	x					x		
Support	Process and Product Quality Assurance	x	x				x		
	Problem Resolution	x	x	x			x		
	Management support & Commitment	x				x	x		

The key aim of RAPID is to pinpoint the strong points of companies and also make certain threats and improvement prospects[7]. The Rapid assessment encompasses 210 questions which will be appraised concentrating on risk management practice and analysis of the consequence of the process.

The RAPID assessment was constructed following the findings from a technical report, Much of the assessment is done through an interview it places its significant importance on the competence of the person doing the assessment, the assessor assumes dual roles which are team leader and support assessor, in each of the roles the assessor will have a specific task to carry out[28] RAPID was established by the Australian Software Quality Institute [29].

5) SPINI (An approach for SPI Initiation)

SPINI is an ISO/IEC 15504 or SPICE compatibility assessment method it was recognized as part of the SataSPIN project it was introduced in 1998 in the Satakunta region Western Finland[7]. This initiative was established to advantage software companies most of which were part of SMEs to grow their operations with known standard frameworks. SPINI encompasses 3 agendas; First of all, the organization desires to appreciate the abilities of SPI in achieving its business objectives. Secondly, an assessment of the current software processes desires to be complete and

lastly the SPI actions require be planned and supported. The framework comprises of events that are assembled from peripheral advisors; points of view steps; understanding priorities, carrying out assessment and supporting SPI. [14] The method articulates the important actions that were established as beneficial in starting up SPI using external support [7].

6) MPS-BR

MPS.BR Program is a countrywide mobilization drive that was envisioned for the improvement of software process in Brazil which created the MPS Model.

The MPS.BR program acronym is in Portuguese meaning “Melhoria de Processo do Software Brasileiro” was created in late 2003 by an Association for Promoting the Brazilian Software Excellence (SOFTEX). This association is composed of organizations in some cities of Brazilian states. SOFTEX has about 1,600 conglomerated companies and 70% are SSC.

The MPS-BR model constitutes of three key components: MPS Reference Model (MR-MPS); MPS Assessment Method (MA-MPS); and MPS Business Model (MN-MPS). MR-MPS and MA-MPS are ISO/IEC 12207 and ISO/IEC 15504, CMMI-DEV compatible, based on SE best practices, and in harmony with the Brazilian companies reality.

The three key components consist of 23 processes organization, innovation and deployment; Causal analysis and resolution; Organizational process performance; Qualitative PM Risk management; Decision analysis and resolution; Requirement development; Technical solution; Validation; Verification; Software integration; Software installation; Product release; Training; Process establishment; Process assessment and improvement; Tailoring process for PM; Configuration management; QA; Acquisition; Measurement; PM and Requirement management.

### 7) MECA

In Pakistan most large, medium and SSC had challenges in establishing quality improvement programmes. Contrary to others, Khokhar et al [4] perceive large organizations as distinct to SSC finding them generally complex to implement and adjust to quality improvement strategies. This proved complex since most quality models like CMMI and SPICE primarily addressed the requirements of large companies only. The MECA model offers a continuous monitoring approach for software processes and provides basis to commence improvement.

MECA is motivated by the Plan, Do, Check Act (PDCA) model. The PDCA was established by Walter Shewhart, it is responsible for the basic viewpoint of a controlled, cyclic approach to continuous improvement. MECA aids companies which continuously struggle to attain CMMI.

Thereafter the output of the SPI framework designed will also be used to compare the existent outcome of adopted SPI frameworks. We use two different SPI comparison methods, the characteristics method and the needs mapping comparison method. Fig.5 is a UML notation of a proposed dynamic SPICOF which has 3 main components being Analysis, Mapping IA and requirements and has 8 processes. The requirements are then used for the design and development of a SPI framework which is external to this work to see fig 6.

### C. The proposed SPICOF

#### 1) Analysis

In the first component denoted as 1 the proposed framework has 2 steps one is to analysis of SDE to elicit the challenges and the environmental context in which software is being developed. Then the second step identifies software frameworks that have been used in similar SDE to study the IA from which the existent frameworks being compared. These are denoted with numbers 1,2,3,...n in fig. 4. This serves to elicit the potential IA on the challenges that were addressed and in the SDE. The number of frameworks to select for comparison depends on existent frameworks chosen, preferably a choice of frameworks that addresses at list one of the challenges and SDE issues elicited. The output of step one and two in the first component of the proposed SPICOF are then tabulated in a matrix as input in the second component of the proposed SPICOF.

#### 2) Mapping Improvement action to challenges

This component of the proposed SPICOF has 2 processes, 3 and 4, the first process identifies which potential IAs should be taken, this IA is then mapped to a corresponding challenge and SDE issues listed in the matrix. This potential IA in the matrix is entered so as to give a basis from which choice will be made for the best. In same step the different potential IAs identified in the matrix are compared to choose the best option that can solve the challenge and the SDE issues identified.

Challenges and SDE issues that have not been mapped to a potential IA are put aside in the 4<sup>th</sup> process for other challenges that have not been mapped to IAs in the matrix to be identified as a candidate challenges for further scrutiny whose IA are sought from other best practices. The mapping is done at low level so as to scrutinise all significant detail required to solve particular challenge.

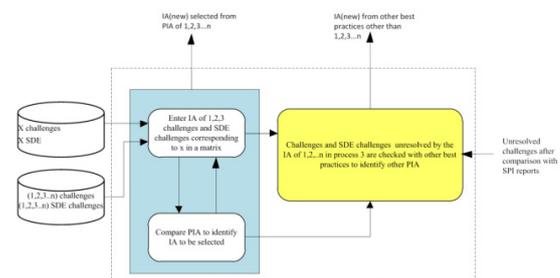


Fig 4. Mapping improvement action to challenges

#### 3) Requirements for SPI

Component 3 has three processes, 5 through 7 with iteration between step 7 back to the 4<sup>th</sup> process in the second component and then to the 5<sup>th</sup>, 6<sup>th</sup> and 7<sup>th</sup> in the third component. In step 5 candidate improvement actions are identified as requirements using the characteristics comparison method the requirements are scrutinized for presentation as measurable, comparable and extremely objective characteristics being input for construction of the SPI framework. An analysis is performed and results are mapped to output of step 1 and 2 for consistency and traceability. In this step again uses characteristics comparison method which is based on extensive literature survey in steps 1 and 2. The requirements that are defined in terms of the chosen potential IAs and input for step 6.

The actual construction of an adaptable SPI framework is then done using data from this stage SPA and SPI is done outside this scope see fig 6. All chosen IAs from different frameworks are put together to address the challenges identified, environmental issues elicited and results of the assessment from the SPA conducted.

In proses 7 the final comparison of the report of the SPI results highlighting the impact is compared with impacts of IAs conducted from the candidate actions adopted. After this assessment the gaps identified are iterated through step 4, 5, 6, and 7 until marginal gaps that can be negligible are identified then we stop here.

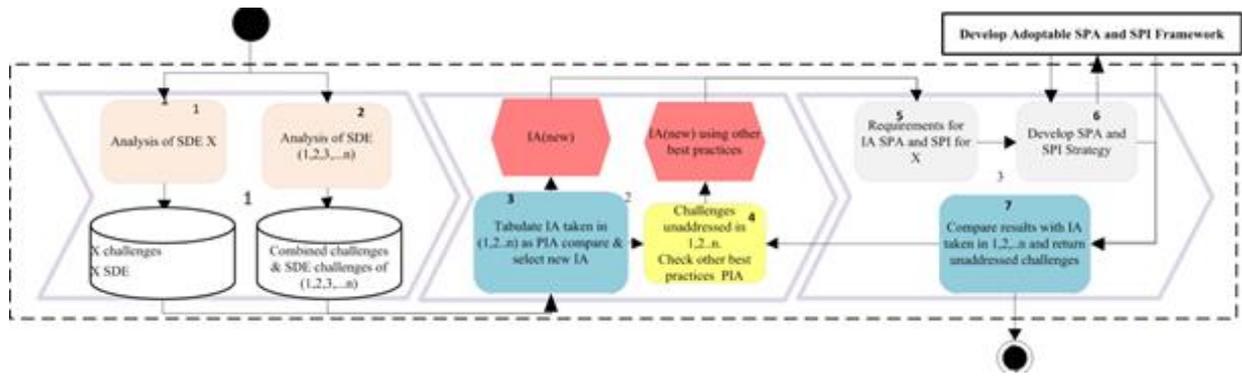


Fig 5. Dynamic SPI comparison framework for generating requirements for an adoptable SPI framework

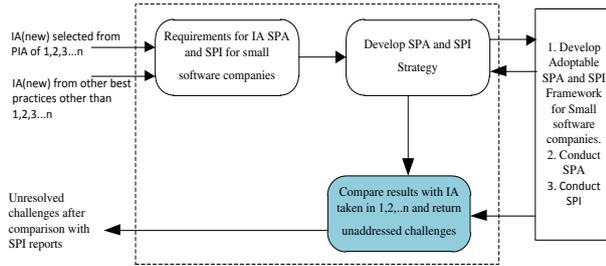


Fig 6. Requirements Improvement actions

IV. WHAT MAKES OUR FRAMEWORK DIFFERENT

A. Overview

SPA and SPI frameworks are significant in software process modeling and improvement to the extent that where software development is being practiced without SPI has a lot to be desired. This framework paves way for simplifying SPA and API for SSC. Unlike other frameworks this proposed framework compares SPI frameworks for the purpose of benchmarking to design and develop an adaptable SPI framework. In table II we compare the characteristics of our framework with the frameworks of [1, 2, 6].

The proposed framework has a twofold comparison; this makes it unique from the other SPICOFs that look at comparison only in perspective of what has been done. We compare for assessment before process improvement and for assessment of our SPI framework after application.

B. Assessment of proposed framework

Once characteristics of the company and the SDE in which they operate are identified, IA are identified and improvement is conducted then an evaluation based on test of safeguards in operational planning, the flexibility, informal management styles and risk management to check the changes in the safeguards still affect the processes. And then back to the loop.

V. OUR OWN ANALYSIS

A dilemma came into SE around 1965-1985 when failure of projects was increasing and the main cause of this crisis was the overall complexity of the software process, with projects over shooting budgets; and schedule; Low quality software; Requirement are not met; generally, projects were troublesome [30]. Accordingly, different solutions came up including SPI with frameworks and models like CMMI and SPICE.

Frameworks are reusable design for all or parts of development that is represented by a set of abstract processes, describing how software development is broken down into a set of interacting processes and the interaction with in the sub processes. If only SSC could effectively use frameworks, There is a promising avenue in this for achieving best practices in software development particularly by reducing the cost and ultimately improving the quality of software. Frameworks have played a pivotal role in software process modeling, evaluation and improvement; they have

TABLE III.

A COMPARISON OF CHARACTERISTICS OF PROPOSED COMPARISON FRAMEWORK TO OTHER COMPARISON FRAMEWORK

Zarour et al. Characteristics[1]	Halvorsen Characteristics[2]	Anacleto et al. Characteristics[6]	Proposed Characteristics
-Total of evaluated processes -Assessed processes. -Number of processes to be enhanced -Assessment period	-Geographic origin/spread -Scientific origin -Development/firmness -reputation -Analysis procedures	-Budget -Guidance for process choice -Support for identification of threat and perfection proposals -Requirement for precise SE awareness from the company representative -Tool provision -Public handiness	-Numbers of challenges being addressed. -Similarity of SDE in terms of skill, customer awareness, PM practices. -How challenges addressed vs adopted of tools. -Impact on quality of software. -Timeliness to market, team size vs project completion time

also enabled developers to rapidly produce new applications. However, the absences of an adaptable framework for different contexts in SSC make it difficult to improve the processes. An understanding of the characteristics of SSC will fill the gap that has made comparison of different SDE an attainable.

Most organizations fail to adopt such frameworks due to the costs involved, [31], and time factor and yet the other SPI frameworks continue to adopt the already marked as complicated frameworks. Although performing the CMMI based improvement is difficult and expensive for SSC [14]. Small companies often fail to afford the cost overrun, which is necessary for the popular frameworks.

In Africa for example, the case is different although the SSC face the same challenges of lack of implementation of best practices, lack of adopted standards and generally disorganized software processes; there is also no evidence in literature, of attempts to localize the existing frameworks for SPI[32-36].

Countries like Brazil, Mexico, Pakistan, Australia and Malaysia have developed local standards like MPS-BR, Moprosoft, iSPA, MARES, SPINI and MECA [9, 37, 38]. Because the existing international frameworks like CMMI and ISO 15505 have not registered substantial successes with SSC due to different reasons, all attempts to improve software process have not been ceased.

The success of a better software industry in places like Africa will be dependent on how much effort can be put to improve the status quo[36]. It is therefore high time since the industry is growing rapidly that we began to assess the software processes in Africa, in order to improve on quality of software.

Software process is a set of related activities that split software development efforts into separate stages to improve design, product management, and PM; it leads to the production of software. These activities may involve the development of the software from the basic idea, or transforming existing software. Processes are significant for providing avenue for efficiency, effectiveness, and reduce waste in an organization. Ineffective processes cause frustration, delays, and increased costs, however they can be improved.

SPI frameworks in SSC face numerous challenges, most of all is the ability to demonstrate the expected business results as suggested by Wangenheim et al. [39] and Cater-Steel et al. [40]. These authors suggest the reduction in process assessment expenses and the time essential to make the SPI benefits noticeable. Accordingly Staples et al. [41], also proposes that SPI approaches dedicated to SSC are better cheaper and within a reasonable time for SSC.

## VI. CONCLUSION AND FUTURE WORK

Our findings indicate that contextualization of SPI require each specific context to be treated differently and hence the need for an adaptable framework. We have noted that the SDE for SSC significantly differ and the difficulty in one

place may also occur differently and this calls for tailoring the solution differently.

We have observed that the application of the CMMI, SPICE and other standard frameworks and there adoption for the purpose of design of SPI frameworks may not necessarily take us to successful results for SSC. The software development processes in SSC and the factors that affect them do not occur in the same pattern project management process area may have less issues compared to PRM process area in two software different environments owing to different reasons.

These observations make the road map for the development of a SPICOF clearer giving us good insights into the development of the adaptable SPA and SPI framework for SSC which is context independent. Therefore the framework being developed will draw lessons from the above observations and other findings next studies to identify characteristics of SSC. In the near future we plan to make comparison of different SDE in relation to the SSC characteristics and how much it affects quality before we validate the SPICOF being developed for SPI in SSC.

## REFERENCES

- [1] M. Zarour, J.-M. Desharnais, and A. Abran, "A framework to compare SPAt methods dedicated to small and very small organizations," *International Conference on Software Quality-ICSQ*, 2007, pp. 16-17.
- [2] C. P. Halvorsen and R. Conradi, "A taxonomy to compare SPI frameworks," *European Workshop on Software Process Technology*, 2001, pp. 217-235. [http://dx.doi.org/10.1007/3-540-45752-6\\_17](http://dx.doi.org/10.1007/3-540-45752-6_17)
- [3] A. M. AL-Ashmori, B. B. Rad, and I. Suhaimi, "SPI frameworks as alternative of CMMI for SMEs: a literature review," *Journal of SE*, vol. 11, pp. 123-133, 2017. <http://dx.doi.org/10.3923/jse.2017.123.133>
- [4] M. N. Khokhar, A. Mansoor, K. Muhammad Nadeem, R. Shafiq Ur, and A. Rauf, "MECA: SPI for small organizations," *International Conference on Information and Emerging Technologies*, 2010, pp. 1-6. <http://dx.doi.org/10.1109/iciet.2010.5625678>
- [5] A. A. Khan, J. Keung, M. Niazi, S. Hussain, and M. Shameem, "GSEPIIM: A roadmap for SPA and SPI in the domain of global software development," *Journal of software: Evolution and Process*, vol. 31, p. e1988, 2019. <http://dx.doi.org/10.1002/smr.1988>
- [6] R. Anacleto, C. G. Von Wangenheim, C. F. Salviano, R. Savi, S. Brasil, and C. S. Brasil, "A method for process assessment in small software companies," 2004.
- [7] C. G. von Wangenheim, A. Anacleto, and C. F. Salviano, "Mares-a methodology for software process assessment in small software companies," Technical Report LPQS0012004.
- [8] X. Larrucea, R. V. O'Connor, R. Colomo-Palacios, and C. Y. Laporte, "Software process improvement in very small organizations," *IEEE Software*, vol. 33, pp. 85-89, 2016. <http://dx.doi.org/10.1109/ms.2016.42>
- [9] T. Tahir, G. Rasool, and M. Noman, "A systematic mapping study on software measurement programs in SMEs," *e-Informatica Software Engineering Journal*, vol. 12, 2018. <http://dx.doi.org/10.1109/icacs.2018.8333491>
- [10] C. Y. Laporte, J.-M. Desharnais, M. M. Abouelfattah, J.-C. Bamba, A. Renault, and N. Habra, "Initiating SPI in Small Enterprises: Experiments with CETIC's Micro-Evaluation Framework," *International Conference on Software Development (SWDCREK 2005) Reykjavik, Iceland*, 2005.

- [11] E. Mnkandla, "About software engineering frameworks and methodologies," in *AFRICON 2009*, 2009, pp. 1-5. <http://dx.doi.org/10.1109/afcon.2009.5308117>
- [12] J. Nakatumba-Nabende, B. Kanagwa, R. Hebig, R. Heldal, and E. Knauss, "Hybrid Software and Systems Development in Practice: Perspectives from Sweden and Uganda," *International Conference on Product-Focused SPI*, 2017, pp. 413-419. [http://dx.doi.org/10.1007/978-3-319-69926-4\\_30](http://dx.doi.org/10.1007/978-3-319-69926-4_30)
- [13] M. O. Tingey, *Comparing ISO 9000, Malcolm Balbridge, and the SEI CMM for Software: A Reference and Selection Guide*: Prentice Hall PTR, 1996.
- [14] C. Y. Laporte, S. Alexandre, and R. V. O'Connor, "A software engineering lifecycle standard for very small enterprises," *European Conference on SPI*, 2008, pp. 129-141. [http://dx.doi.org/10.1007/978-3-540-85936-9\\_12](http://dx.doi.org/10.1007/978-3-540-85936-9_12)
- [15] F. McCaffery, D. McFall, and F. G. Wilkie, "Improving the express process appraisal method," *International Conference on Product Focused SPI*, 2005, pp. 286-298. [http://dx.doi.org/10.1007/11497455\\_24](http://dx.doi.org/10.1007/11497455_24)
- [16] M. C. Paulk, "How ISO 9001 compares with the CMM," *IEEE software*, vol. 12, pp. 74-83, 1995. <http://dx.doi.org/10.1109/52.363163>
- [17] K. E. Emam, W. Melo, and J.-N. Drouin, *SPICE: The theory and practice of SPI and capability determination*: IEEE Computer Society Press, 1997.
- [18] M. Tuape and Y. Ayalew "Factors affecting development process in small software companies," *IEEE/ACM Symposium on Software Engineering in Africa (SEiA) 2019*. pp. 16-25 <http://dx.doi.org/10.1109/SEiA.2019.00011>
- [19] M. E. Morales-Trujillo, H. Oktaba, T. Ventura, and R. Torres, "From MoProSoft Level 2 to ISO/IEC 29110 Basic Profile: Bridging the Gap," *Clel Electronic Journal*, vol. 16, 2013. <http://dx.doi.org/10.19153/cleiej.16.1.2>
- [20] V. Ribaud and P. Saliou, "Process assessment issues of the ISO/IEC 29110 emerging standard," in *Proceedings of the 11<sup>th</sup> International Conference on Product Focused Software*, 2010, pp. 24-27. <http://dx.doi.org/10.1145/1961258.1961264>
- [21] C. G. von Wangenheim, A. Anacleto, and C. F. Salviano, "Helping small companies assess software processes," *IEEE software*, vol. 23, pp. 91-98, 2006. <https://doi.org/10.1109/ms.2006.13>
- [22] F. J. Pino, C. Pardo, F. Garcia, and M. Piattini, "Assessment methodology for SPI in small organizations," *Information and Software Technology*, vol. 52, pp. 1044-1061, 2010. <https://doi.org/10.1016/j.infsof.2010.04.004>
- [23] R. Z. R. M. Ali and S. Ibrahim, "An application tool to support the implementation of integrated SPIt for Malaysia's SME," *Malaysian Conference in Software Engineering*, 2011, pp. 177-182. <https://doi.org/10.1109/mysec.2011.6140665>
- [24] R. Ali and S. Ibrahim, "An Integrated SPA for Malaysia's SME," *Postgraduate Annual Research Seminar (PARS) in UTM Skudai*, 2009.
- [25] 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," *2nd International Conference on Software Engineering and Data Mining*, 2010, pp. 225-230.
- [26] A. Cater-Steel, M. Toleman, and T. Rout, "Process improvement for small firms: An evaluation of the RAPID assessment-based method," *Information and Software Technology*, vol. 48, pp. 323-334, 2006. <https://doi.org/10.1016/j.infsof.2005.09.012>
- [27] T. P. Rout, "ISO/IEC 15504 and Spice," *Encyclopedia of Software Engineering*, 2002. <https://doi.org/10.1002/0471028959.sof171>
- [28] T. Rout, "Evolving SPICE—the Future for ISO 15504," *1<sup>st</sup> International Conference on SPI and Capability Determination*, Limerick, Ireland, 2000, pp. 1-8.
- [29] H. Oktaba and M. Piattini, "SPI for Small and Medium Enterprises: Techniques and Case Studies," 2008. <https://doi.org/10.4018/978-1-59904-906-9>
- [30] E. Klotins, M. Unterkalmsteiner, and T. Gorschek, "Software engineering in start-up companies: An analysis of 88 experience reports," *Empirical Software Engineering*, vol. 24, pp. 68-102, 2019. <https://doi.org/10.1007/s10664-018-9620-y>
- [31] T. Dybå, "Enabling SPIt: an investigation of the importance of organizational issues," *Empirical Software Engineering*, vol. 7, pp. 387-390, 2002. <https://doi.org/10.1023/a:1020535725648>
- [32] C. Ferreira and J. Cohen, "Agile systems development and stakeholder satisfaction: a South African empirical study," *South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, 2008, pp. 48-55. <https://doi.org/10.1145/1456659.1456666>
- [33] B. De Wet and J. Visser, "An evaluation of software project risk management in South Africa," *South African Journal of Industrial Engineering*, vol. 24, pp. 14-29, 2013. <https://doi.org/10.7166/24-1-497>
- [34] E. Kabaale, G. Mayoka Kituyi, and I. Mbarika, "Requirements engineering process improvement challenges faced by software SMEs in Uganda," 2014. <https://doi.org/10.5120/15349-3690>
- [35] J. K. Balikuddembe and M. Tuape, "An Ambiguity Minimization Technique during Requirements Elicitation Phase," *International Conference on Computational Science and Computational Intelligence (CSCI)*, 2017, pp. 945-950 <https://doi.org/10.1109/csci.2017.164>.
- [36] Y. Ayalew and K. Motlhala, "Software Process Practices in Small Software Companies in Botswana," *14th International Conference on Computational Science and Its Applications*, 2014, pp. 49-57. <https://doi.org/10.1109/iccsa.2014.20>
- [37] A. A. Khan, J. Keung, S. Hussain, M. Niazi, and M. M. I. Tamimy, "Understanding SPIt in global software development: a theoretical framework of human factors," *ACM SIGAPP Applied Computing Review*, vol. 17, pp. 5-15, 2017. <https://doi.org/10.1145/3131080.3131081>
- [38] V. Garousi, M. Felderer, and T. Hacaloğlu, "What we know about software test maturity and test process improvement," *IEEE Software*, vol. 35, pp. 84-92, 2018. <https://doi.org/10.1109/ms.2017.4541043>
- [39] C. G. v. Wangenheim, A. Anacleto, and C. F. Salviano, "Helping small companies assess software processes," *IEEE Software*, vol. 23, pp. 91-98, 2006. <https://doi.org/10.1109/ms.2006.13>
- [40] A. P. Cater-Steel, "Process improvement in four small software companies," *Australian Software Engineering Conference*, 2001, pp. 262-272. <https://doi.org/10.1109/aswec.2001.948520>
- [41] M. Staples and M. Niazi, "Two case studies on small enterprise motivation and readiness for CMMI," *11<sup>th</sup> International Conference on Product Focused Software*, Limerick, Ireland, 2010. <https://doi.org/10.1145/1961258.1961274>