

Software process improvement initiative in medium size IT organization: a case study

Mohammad Zarour, Mamdouh Alenezi

College of Computer and Information Sciences (CCIS) Prince Sultan University, Riyadh, Kingdom of Saudi Arabia

Article Info

Article history:

Received Oct 6, 2020

Revised Apr 15, 2021

Accepted May 6, 2021

Keywords:

CMMI

Cultural issues

SCAMPI

Software process improvement

ABSTRACT

IT organizations are striving to develop software solutions that meet customer needs as well as their business goals. Software organizations continuously improve their software development practices for faster time-to-market and better software quality. The success factors to conduct process improvement initiatives are documented in the literature, but their effect and existences differ from one place to another. Hence, as the culture and organizational behavior in developing software varies across the world, it becomes interesting to report and discuss local experiences in different regions. The experience gained in this empirical study differs from that gained in the previous work in the sense that it was the organization that initiated this initiative and this ensures the top management commitment in conducting the process improvement initiative. We have used CMMI continuous representation and SCAMPI class C appraisal method to conduct the appraisal. In this empirical study, most of the success factors are met to conduct the process improvement initiative, including: Staff involvement, and high staff morale. Several process pitfalls have been identified; an interesting one is that adopting powerful case tools is not enough to control the development process if they are not fully used.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Mohammad Zarour

Department of Software Engineering

Prince Sultan University

P.O. Box No. 66833 Rafha Street, Riyadh 11586 Saudi Arabia

Email: mzarour@psu.edu.sa

1. INTRODUCTION

IT organizations are striving to develop software solutions that meet customer needs as well as their business goals. As the culture and organizational behavior in doing business in general and developing software, in particular, varies across the world, it becomes interesting to report and discuss local experiences in different regions and how the best practices in developing software are implemented. Such reporting in the IT domain is scarce in the literature, which will provide researchers with abundant resources to look at the cultural issues and how they affect the development process and business. For instance, cultural issues influence doing business in China [1], [2], especially for multinational organizations that are not uncommon in the IT sector. Organizational success demands an appreciation for the culture's role in their planning and should embed culture in their traditional operation [3].

Meanwhile, IT organizations worldwide are increasingly acknowledging the significance of reducing software development duration but without compromising the quality. Hence, organizations must respond to their customer needs with high-quality products as fast as possible. To reach this level of continuous software delivery, many organizations think to enhance their software development processes to meet customer expectations. Moreover, software developers give much attention to their development

methods to ensure that the resultant software runs optimally on the new hardware [4]. Accordingly, Software organizations continuously improve their software development practices for faster time-to-market and better software quality [5]. Typically, organizations improve their software development methodologies based on perceived industry trends. DevOps, a new development philosophy, is very popular nowadays, and different companies work to adopt it [6]-[8].

On the other hand, other organizations with proven practices work to enhance their adopted software process rather than replacing it. To make sure that there is no deviation in the adopted proven practices and any deviation, rectifying actions should occur. For instance, many organizations apply standards and models such as capability maturity model integration (CMMI) [9] or ISO 15504 [10] to assess and improve their adopted processes. Several empirical studies have been documented in this regard, see for example [11]-[18]. Recently, data mining techniques are used in what is called process mining to support the process improvement initiatives, see for example [19], [20]. Exploring, assessing and understanding the organizational cultural and its effect on software process improvement (SPI) initiatives is crucial, see for example [21]-[25]. Employees are usually hesitant to participate in SPI initiatives due to their work load, deadline pressures, and their feeling of being offended and criticized during the SPI assessment. Hence, enhancing personal motivation in specific SPI initiatives is also crucial for its success [26]. This paper presents an empirical study in this regard and reports the experience and lessons learned by conducting a model-based software process improvement (SPI) initiative in a national IT organization in Saudi Arabia.

The experience gained in this empirical study differs from that gained in a previous work documented in [13]. The main difference is that the software engineering process group (SEPG) team approached the organizations in the previous experience and convinced them to conduct SPI initiatives. In contrast, in the current experience, the organization initiated the process and approached us to conduct the SPI initiative. This difference greatly influenced the findings; in the previous experience, the employees were reluctant to share information about process quality and supporting evidence with the SEPG team, which was the main obstacle. On the contrary, most of the current empirical study employees were collaborative due to the top management involvement and commitment to the SPI initiative. The commitment to the SPI initiative of both managerial levels and software developers has been studied in [17]. A behavior-based Commitment Model is proposed to raise awareness about the people issues in improving processes. Such factors, among others, are crucial to the success of any SPI initiative.

A detailed study of the factors that hamper pursuing the SPI initiative has been documented in [27]. Both studies revealed that there are both similarities and differences between practitioners in rating success and failure factors. The rating was affected by cultural issues in different countries. Moreover, these studies have shown that different practitioners are aware of what is imperative for the implementation of SPI programs. In the case study reported here, most of the success factors are met to conduct the SPI initiative, including top management commitment, Staff involvement, and high staff morale. These factors help the SEPG team conduct a successful SPI initiative and identify interesting factors that obstruct the development process and delay the delivery to customers.

The rest of the paper is organized as follows: Section 2 discusses the methodology and research drivers for this research paper. It describes the organization under study and why they want to start an SPI initiative. Section 3 presents the initial findings after the first day meetings based on which the SPI plan is drawn. Section 4 presents the detailed findings and analysis, and section 5 discusses the lessons learned. Section 6 discusses the validity threats, and section 7 presents the conclusion and future work.

2. METHODOLOGY AND ORGANIZATION'S DRIVERS

Organization X, whose anonymous name refers to the IT organization under study, is hired by a public agency to develop and run an online application that provides services the users in Saudi Arabia. Organization X is of medium size with around eighty employees. As the system is put in use for several years now, more services are being continuously requested to be added or updated in the system. Recently, significant-quality issues were discovered in the system that resulted in a significant delay in delivering the requested services. The organization's executive director and project managers are interested in assuring that these quality issues have been successfully dealt with to ensure the quality of services provided and delivered on time. To support that effort, the executive director sought an independent assessment of the adopted development process to determine what quality problems exist and solve them. Therefore, the executive director has contacted the software engineering process group (SEPG) at PSU, who are the authors of this paper, to conduct the appraisal and provide recommendations.

The organization is not interested in recertification or reaching a certain maturity level, but are more interested in identifying the process areas with significant weakness, the CMMI v1.3 [9] continuous model is

used for this purpose, and standard CMMI appraisal method for process improvement (SCAMP) C is adopted as an appraisal method. The main objectives of the assessment can be summarized as follows:

- Provide an accurate appraisal compliant with SCAMPI C appraisal.
- Conduct the appraisal within three calendars' workdays to minimize cost and disruptions to the site operations.
- Provide the organization with more insight into findings and recommended actions.

The appraisal team spent the first day of the assessment interviewing with various team members working on the various software processes from business analysts to deployment and operation teams. By the end of the first day, the assessment team met together and identified the main issue and information sources from where we need to collect more data, see Table 1 for types of information and sources.

Table 1. Types of information sources

Information/ Information Source	Interviews	Observations	Documents
Business Analysts	Yes		Yes
System Analysts	Yes	Yes	Yes
Project Managers	Yes	Yes	
Developers	Yes	Yes	
Operation team members	Yes		
Executive Directors	Yes		

Based on the identified issues and the time frame available for this phase of assessment, four process areas are chosen for detailed assessment as they have major issues. The selected process areas are: requirement management, requirement development, configuration management, and product integration. In the next phase, the initial appraisal of the four selected process areas, has been conducted. A detailed appraisal, based on CMMI v1.3, followed in the initial appraisal to entail the findings and describe in detail the weakness of each process area. Finally, the results and recommendations have been presented and discussed with the organization's team members, while the scientific analysis of this experience is documented in this paper. CMMI class C appraisals do not rate the process maturity. Instead, it gives a quick view of the process capability and shed light on possible internal improvements for the process(s) under investigation. Table 2 shows the coloring scheme used to rate each process area and its practices.

Table 2. Labels used to characterize appraisal results based on the assessment

Label Color	Description
	Practices are not appropriately addressed, which will harm some aspects of project performance or product quality in subsequent phases.
	Practices are partially addressed. They may negatively impact project cost and schedule in subsequent phases of the project.
	Practices are adequately addressed and are not likely to negatively impact project cost and schedule in subsequent project phases.

3. APPRAISAL INITIAL RESULTS

It is noticed that the received requests by the development team are either change requests, a production issue, or a quality assurance (QA) deployment issue. Two-thirds of the change requests and QA deployment issues are of normal priority, while the other third is high to critical priority. Around sixty percent of the production issues are of high and critical priorities, while around 40 percent is of normal to low priority.

It has been noticed that the time needed, measured in working days, to resolve a change request of low priority is double the time needed for resolving production issues of the same priority and around twelve times more than that to resolve the QA deployment issues. The same pattern holds for other priorities of the different issues. The long-time it takes to resolve a change request and the variation in time among different requests with different priorities are among the main drivers to conduct the appraisal. After the on-site appraisal, the main strengths in the current processes that are worth mentioning are as follows:

- Focusing on the front-end was a big success story for the organization without worrying about the infrastructure part. Changing this momentum will hurt the team and the progress of the team
- The organization's team is divided into small sub-teams. This should continue since smaller teams reduce complexity and maximize efficiency. Smaller team structures keep teams from getting bogged down and losing momentum.

- Adding New employees should be studied as it does not always benefit teams. Based on our observations, the team needs one or two bilingual system analysts and one or two in the development team to build automation.
- Automating continuous delivery with a delivery pipeline enhances the speed of delivery.
Moreover, several weaknesses have also been noticed that includes the following:
- Requirements Issues. There is a gap between Business requirement specifications (BRS) and software requirements specifications (SRS).
- SRS is not appropriately reviewed, and not all concerned team members are involved in it.
- The build process is ad hoc and fully manual.
- Environment inconsistencies. Every environment is different, which delay thing.
- The mocking backend is not available for test purposes.
- Test data is a big issue. There should be a mechanism for the development team to create their test data.
- Communication issues are everywhere between teams.
- Lack of resources/Staff for some process areas.
- The project manager has neither a proper tool nor measures to monitor and track projects and CRs.
- Release management is missing.
- The process for project initiation and opening a new change request is unclear.
- Security procedures are taking a lot of time, which hinders the progress.

All these issues were acknowledged verbally by the team, and the collected questionnaires also acknowledged them. Each one of these issues can be solved in different ways. The recommendations for solving these issues should be made in a different phase to assess the possible solutions' pros and cons before adopting any.

4. APPRAISAL DETAILED RESULTS AND ANALYSIS

Due to the time designated for the appraisal, the appraisal's team and the organization's team agreed to divide the assessment initiative into phases. The following subsections discuss in detail the assessed processes and the results of the assessment for each of these processes.

4.1. Assessed software process areas

The first phase covers four processes: requirement management, requirement development, configuration management, and product integration.

4.1.1. Requirement management

The purpose of requirements management (REQM) is to manage the project's products and product components' requirements and identify inconsistencies between those requirements and the project's plans and work products. The client requests new services or requirements. The business analyst communicates with the client to arrange for meetings to understand and complete the requirements definition. Business requirements are delivered to the system analyst to develop the SRS document submitted to the development team to develop the requested service and the QA team to develop the test cases.

4.1.2. Requirement development

Requirements are the basis for design. The development of requirements includes the following activities: elicitation, analysis, validation, and communication of customer needs, expectations, and constraints to obtain customer requirements. Collection and coordination of stakeholder needs. Development of the lifecycle requirements of the product. Establishment of the customer requirements. Establishment of the initial product and product component requirements consistent with customer requirements.

The business requirement specifications (BRS) are documented using a predefined template. The system analyst translates the BRS into system specifications using another predefined template to produce the SRS document. Non-functional requirements were identified. This became the set of customer requirements. Requirements for product components were established by functional area via scenarios and use cases, including those for interfaces. Use cases and scenarios evolved via joint application development (JAD) sessions into a final functional design that was jointly reviewed and approved as the basis for further work.

4.1.3. Configuration management

The purpose of configuration management (CM) is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits. Configuration items are identified basically as part of the change requests. Client data, environments, and documents are treated as CIs in addition to code. The collaborative lifecycle management

(CLM) is used as a tool to manage and control work products. Reports are generated from CLM and used by management to monitor project progress, risks, issues, and track changes to closure. Status on configuration work products is in weekly/monthly reports. Deliverables log tracks details of each deliverable. Risks/issues logs are maintained.

4.1.4. Product integration

The purpose of product integration is to assemble the product from the product components, ensure that the product, as integrated, functions properly, and deliver the product. Integration environments were established. Integration scripts are run in a defined sequence to build the system. After developers test the components via a unit test, the code is assembled in the pre-build server before moving into the test environment. Upon completing verification activities, the product is delivered to the Product team for production acceptance test (PAT) environment.

4.2. Appraisal findings

Figure 1 shows the appraisals' results. Several specific practices are found to have no evidence that supports their existence. The specific practices that are labeled red followed by those labeled yellow need more attention to be improved. For instance, a good practice when managing the requirements is to assess the impact of coming requirements on the existing commitments. Organization X has no evidence that such practice is adopted hence this may result in unexpected consequences on the current commitments when implementing the new requirements. Moreover, the commitments themselves are not negotiated or recorded anywhere! Other practices that are related to the requirement management and are missed include: evaluating the impact of requirement changes from the standpoint of relevant stakeholders, generating requirement traceability matrix and more. With regard to the requirement development process, the best practices missed include: prioritizing customers' functional and non-functional requirements, maintaining relationships between requirements that would be useful during change management and requirements allocation, allocating requirements to product components and the architecture and more.

The configuration management process has, also, various practices that are not adopted in the organization, this includes: Assigning a unique identifier for each configuration item, specify the relationship among various configuration items, obtaining authorization from the change control board (CCB) before creating or releasing baselines of configuration items and specifying the latest version of baselines. Similarly, the product integration process has also been found to miss various best practices that includes: periodically review and revise the product integration strategy, recording the rationale for decisions made and deferred, ensure the compatibility of the interfaces throughout the life of the product and more. For more details about the specific practices per process area please refer to CMMI guide [9]. A sample of the assessed practices are shown in Table 3.

Table 3. Sample for the red-labeled and yellow-labeled colors

Assessed Process Area	Red-Labeled Practices	Yellow-Labeled Practices
Requirements management	Assess the impact of requirements on existing commitments Negotiate and record commitments Generate a requirements traceability matrix	Establish objective criteria for the evaluation and acceptance of requirements Analyze requirements to ensure that the established criteria are met
Requirement Development	Establish and maintain prioritization of customer functional and quality attribute requirements Establish and maintain relationships between requirements for consideration during change management and requirements allocation Document relationships among allocated requirements	Translate stakeholder needs, expectations, constraints, and interfaces into documented customer requirements Allocate requirements to delivery increments Use proven models, simulations, and prototyping to analyze the balance of stakeholder needs and constraints
Configuration Management	Assign unique identifiers to configuration items Specify the essential characteristics of each configuration item Specify the latest version of baselines	Categorize and prioritize change requests Make the current set of baselines readily available Revise the configuration management structure as necessary
Product Integration	Identify alternative product component integration strategies Periodically review the product integration strategy and revise as needed Record the rationale for decisions made and deferred	Select the best integration strategy Identify verification procedures and criteria for the product integration environment Establish and maintain criteria for validation and delivery of the integrated product

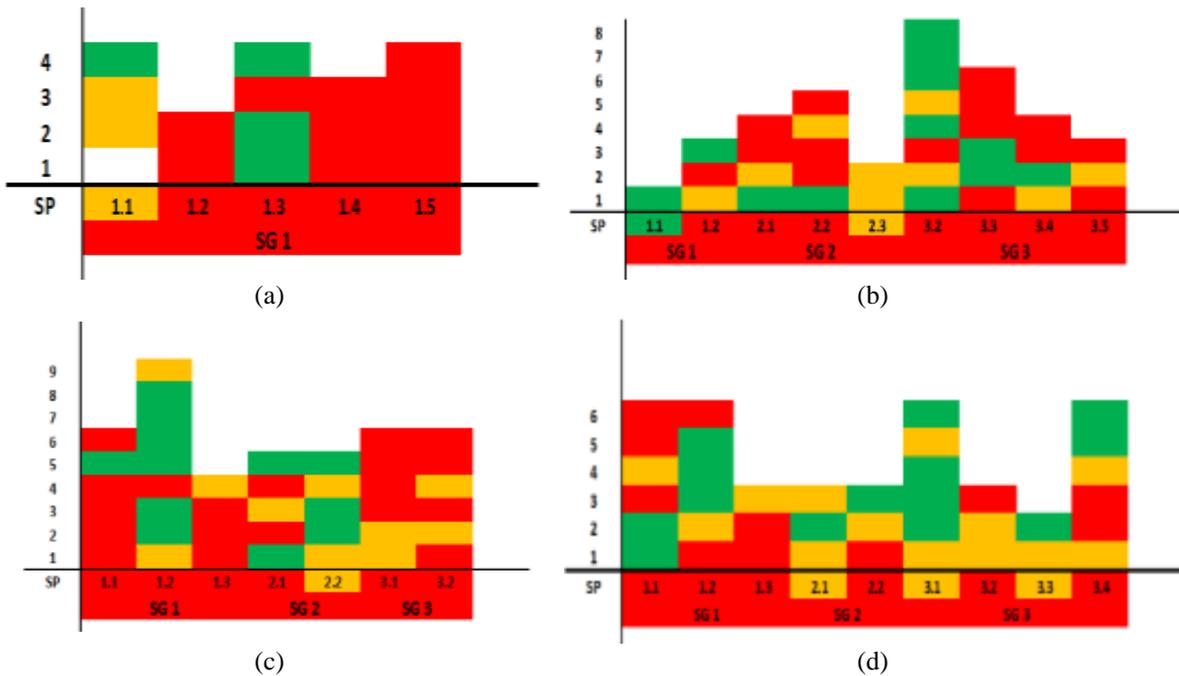


Figure 1. The results of appraisal for the selected processes; (a) requirement management appraisal results, (b) requirement development appraisal results, (c) configuration management appraisal results, and (d) product integration appraisal results

4.3. Discussion

The organization's technical team has several opportunities to improve the requirement management process. One of the main findings is that the business analyst (BA) starts collecting the requirements without the project manager's consent or knowledge. The BA gets the request to collect the requirements from a specific authority, and he obeys that request. As the project manager is aware of project commitments and is not involved in the requirements negotiations with clients, this ends up with assessing the impact of requirements on existing commitments and negotiating commitments to be unfulfilled.

Moreover, evaluating the impact of changing requirements during the developing process is not done as well as tracing requirements to their requesters and dependencies between requirements; this may affect the performance of the development process when a change is approved to be made for a specific requirement and the related or linked ones. Hence, practices related to evaluating the impact of requirement changes from stakeholders' point of view, generating and maintaining requirements traceability, and identifying changes to plans and work products resulting from changing requirements are unfulfilled.

It is also noticed that project management involvement in the requirement management process is feeble. Hence, reviewing the project plans and activities as a result of requirement changes and inconsistencies is weak. This resulted in having practices related to reviewing the project's plans and activities for inconsistencies after adopting changes to requirements, identifying sources of such inconsistencies, and initiating any corrective actions unfulfilled.

Quality requirements are inadequately addressed in the BRS and accordingly in the SRS. Requirements that affect the design and architecture are not addressed. It can be because the system is already developed, and most of the work is to add new services. Despite that, some new requirements may still impact the design, mainly the quality requirements addressing performance issues. Hence, analyst(s) should keep their eyes open to catch such requirements. This has resulted in having some practices unfulfilled that include practices related to establishing and maintaining prioritization of customer functional and quality attribute requirements, developing architectural requirements capturing critical quality attributes along with their measures, determining architecturally significant quality attributes based on the critical mission and business drivers and assess the impact of the architecturally significant-quality attribute requirements on the product and product development costs and risks. On the other hand, other practices are partially fulfilled; this includes practices related to translating stakeholder needs into documented customer requirements, deriving requirements that result from design decisions, identifying and developing internal and external product interfaces, identifying desirable functionality as well as sub-functionality and quality attributes, and develop and use proven models and prototypes to analyze stakeholders needs and constraints.

Requirement prioritization and risk analysis are very critical when developing the requirements documents or SRS. Analysts should work closely with the clients to prioritize the requirements and determine what should be included in the SRS document. No evidence is found that supports the analysis and prioritization of requirements in the organization's project that resulted in having some practices unfulfilled that are related to prioritizing customer functional and quality attribute requirements, managing relationships between requirements to support change management and requirements allocation, identifying essential requirements that have a strong influence on cost, schedule, performance, or risk, examining product lifecycle concepts for impacts of requirements on risks, and analyzing the requirements to determine the risk that the resulting product will not perform appropriately in its intended use environment.

Allocation of requirements to a specific production component and architecture is missed when developing requirements. This task is delegated to the development team. To ensure that all requirements are met, they should be assigned to specific product components. No evidence is found that supports the allocation of requirements to product components in the organization's project that resulted in having some practices unfulfilled that includes allocating requirements to product components and the architecture, documenting relationships among allocated requirements, and analyzing stakeholder needs, expectations, constraints, and external interfaces to organize them into related subjects and remove conflicts. The practice of allocating requirements to delivery increments is partially fulfilled. Surprisingly, the collaborative lifecycle management CLM case tool was bought by the organization but is not used effectively to manage the development process in detail. Hence, the CLM features related to requirements allocation should be put to use.

Technical performance measures to track the performance of the development effort are missed. Such measures are helpful to assess the current situation during project development. Other practices such as analyzing scenarios to better understand customer needs and constraints and the design's analysis as it matures in requirements validation should be explicitly defined and adopted. Several practices are found to be unfulfilled in the configuration management process. For instance, what criteria are used to select configuration items that belong to a specific work product? How are configuration items identified (via an identifier?) what are the main characteristics and relationships among configuration items? Is there a configuration control board (CCB)? Are baselines defined (approved configuration items at a certain point in time)?

Various versions of baselines are not managed to identify the latest version of the baselines, the difference between consecutive baselines, and assess baselines integrity. When a request to change is made, its impact should be analyzed before adding it to the next baseline. Moreover, it is unclear what standards and procedures the adopted configuration management are following. The full use of the CLM for configuration management will help track and document all the activities discussed above. It also helps in complying with well-known standards and procedures for configuration management.

The strategy adopted for product integration is not documented to show the current strategy in detail, why it is adopted, is there other alternative strategies, when was the last time the strategy was reviewed? What are the requirements for the production environment? Are the portions of the environment that are not useful being exposed? This resulted in having some practices unfulfilled that include: identifying alternative product component integration strategies, reviewing the product integration strategy and revise it as needed, recording the rationale for decisions made and deferred, identifying the requirements for the product integration environment, disposing of those portions of the environment that are no longer useful, establishing and maintaining product integration procedures for the product components, establishing and maintaining criteria for product component integration and evaluation, ensuring the compatibility of the interfaces throughout the life of the product, and revising the product integration strategy, procedures, and criteria as appropriate.

The method used for packing and delivering the assembled product is not documented. It does not show the requirements and standards—no measures of its effectiveness or benchmarking with other methods effectiveness is provided. After completing the appraisal and analyzing the results, another meeting was conducted to verify and validate the collected answers. Results were also shown and explained to the software team.

5. LESSONS LEARNED

In this empirical study, the CMMI model's continuous representation and the SCAMPI C appraisal method have been used to conduct the SPI initiative. By the end of this case study, we observed the following:

- One of the main lessons learned by the end of this initiative is the importance of top management commitment to undertake the SPI initiative. The team members are overwhelmed with tasks to deliver on

- time without noticing or giving attention to the quality drawbacks sensed by the executive director who asked for and committed to the SPI initiative.
- Although some team members claimed, at the first meeting, that they adopt the agile process, we found after the assessment that the adopted process model is a modified waterfall model. This means that the team members need more awareness and training of various process models to enhance their ability to select and decide which process model to use.
 - Although powerful tools are bought by the organization, including the IBM CLM tool, we found that the powerful features of such tools are not used effectively. Some of these features are not used at all. Hence, we stress the importance of using and activating these tools to manage the development process effectively.
 - Concerning the previous point, process automation is mal-functioning, which means that continuous delivery and continuous integration is not done effectively, resulting in a long time to deliver and integrate update or new services to the current system. Continuous delivery and continuous integration are critical terms in the new software process model known as DevOps.
 - The QA environment and its dependencies are found to have a lot of differences compared to the deployment environment which caused a lot of delays in the delivery of new features.

6. VALIDITY THREATS

One of the possible threats that may affect the conducted case study is the construct validity, which is concerned with practitioners' interpretation of various practices and terminologies used in the appraisal. We solved this issue by briefly introducing each process area that defines the main terminologies. All the appraisals have been conducted face-to-face, and any questions related to any practice have been explained to the participants accordingly. We do not have any evidence that proves this limitation as none of the participants reported it as a problem. Another possible threat is related to external validity; the generalization of the results to other situations and people requires further research; hence we cannot generalize it for the time being, which makes external validity in process.

7. CONCLUSION

The experience gained in this process assessment initiative differs from that gained in the previous work, where the organization has initiated this initiative and this ensured the top management commitment. We have used CMMI continuous representation and SCAMPI Class C appraisal method to conduct the appraisal. Most of the documented success factors to conduct a process improvement initiative are met. This includes: Staff involvement, and high staff morale. Several process pitfalls have been identified; an interesting one is that adopting powerful case tools is not enough to control the development process if they are not fully used. Based on the observations of this case study, we suggest the following processes to be considered for the second phase of appraisal, as they have several unresolved issues:

- Technical solution: The technical solution is concerned with the design, development, and implementation of solutions to requirements.
- Project monitoring and control: The purpose of project monitoring and control is to provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.
- Process and product quality assurance: The purpose of this process is to provide staff and management with objective insight into processes and associated work products
- Organizational process definition: The purpose of this process is to plan, implement, and deploy organizational process improvements based on a thorough understanding of the organization's processes and process assets' current strengths and weaknesses.

ACKNOWLEDGMENT

The authors would like to thank Prince Sultan University for their support.

REFERENCES

- [1] C. ju Flora Hung, "Cultural influence on relationship cultivation strategies: Multinational companies in China," *J. Commun. Manag.*, vol. 8, no. 3, pp. 264–281, Jul. 2004, doi: 10.1108/13632540410807682.
- [2] J. Yao, A. Crupi, A. Di Minin, and X. Zhang, "Knowledge sharing and technological innovation capabilities of Chinese software SMEs," *J. Knowl. Manag.*, vol. 24, no. 3, pp. 607–634, May 2020, doi: 10.1108/JKM-08-2019-

- 0445.
- [3] T. Lawrence, "Global Leadership Communication: A Strategic Proposal," *Creight. J. Interdiscip. Leadersh.*, vol. 1, no. 1, p. 51, May 2015, doi: 10.17062/cjil.v1i1.8.
 - [4] K. A. Buragga and N. Zaman, "Software development techniques for constructive information systems design," 1 edition. IGI Global, 2013.
 - [5] L. Lwakatare, P. Kuvaja, and M. Oivo, "An Exploratory Study of DevOps: Extending the Dimensions of DevOps with Practices," in *The Eleventh International Conference on Software Engineering Advances*, 2016.
 - [6] S. Rafi, W. Yu, M. A. Akbar, S. Mahmood, A. Alsanad, and A. Gumaei, "Readiness model for DevOps implementation in software organizations," *J. Softw. Evol. Process*, vol. 33, no. 4, p. e2323, Apr. 2021, doi: 10.1002/smr.2323.
 - [7] K. Maroukian and S. R. Gulliver, "Leading DevOps Practice and Principle Adoption," in *9th International Conference on Information Technology Convergence and Services*, 2020, pp. 41–56.
 - [8] M. Zarour, N. Alhammad, M. Alenezi, and K. Alsarayrah, "Devops Process Model Adoption in Saudi Arabia: An Empirical Study," *Artic. Jordanian J. Comput. Inf. Technol.*, vol. 06, no. 03, 2020, doi: 10.5455/jjcit.71-1580581874.
 - [9] CMMI Product Team, "CMMI for Development, Version 1.3," 2010.
 - [10] ISO/IEC, "ISO/IEC TR 15504, Information Technology - Software Process Assessment - Parts 1-9," 1998.
 - [11] A. P. Cater-Steel, "Process Improvement in Four Small Software Companies," *13th Australian Software Engineering Conference (ASWEC'01)*. IEEE, Canberra, Australia, 2001, pp. 262–272, doi: 10.1109/ASWEC.2001.948520.
 - [12] F. J. Pino, C. Pardo, F. García, and M. Piattini, "Assessment methodology for software process improvement in small organizations," *Inf. Softw. Technol.*, vol. 52, no. 10, pp. 1044–1061, 2010, doi: 10.1016/j.infsof.2010.04.004.
 - [13] Z. Alshaikh, M. Alsaleh, A. Alarifi, and M. Zarour, "Process improvement in governmental agencies: Toward CMMI certification," *2015 IEEE Int. Symp. Softw. Reliab. Eng. Work. ISSREW 2015*, 2016, pp. 168–173, doi: 10.1109/ISSREW.2015.7392063.
 - [14] M. Zarour, A. Abran, and J.-M. Desharnais, "Software process improvement for small and very small enterprises," *Software Design and Development*, 2013, doi: 10.4018/978-1-4666-4301-7.ch066.
 - [15] Gleison Santos *et al.*, "Implementing Software Process Improvement Initiatives in Small and Medium-Size Enterprises in Brazil," *Sixth International Conference on the Quality of Information and Communications Technology*. IEEE, Lisbon New University, Lisbon, Portugal, 2007, doi: 10.1109/QUATIC.2007.22.
 - [16] D. Chevers, "Software Process Improvement: Awareness, Use, and Benefits in Canadian Software Development Firms," *Rev. Adm. Empres.*, vol. 57, no. 2, pp. 170–177, Apr. 2017, doi: 10.1590/S0034-759020170206.
 - [17] P. Abrahamsson, "Commitment to Software Process improvement Development of Diagnostic Tool to Facilitate Improvement," *Software Quality Journal*, vol. 8, pp. 63–76, 2019.
 - [18] P. Allen, M. Ramachandran, and H. Abushama, "PRISMS: an Approach to Software Process Improvement for Small to Medium Enterprises," *Third International Conference on Quality Software, 2003. Proceedings*, 2015, doi: 10.1109/QSIC.2003.1319105.
 - [19] C. Matthies, F. Dobrigkeit, and G. Hesse, "Mining for Process Improvements: Analyzing Software Repositories in Agile Retrospectives," in *Proceedings - 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW 2020*, 2020, pp. 189–190.
 - [20] T. Graafmans, O. Turetken, H. Poppelaars, and D. Fahland, "Process Mining for Six Sigma: A Guideline and Tool Support," *Bus. Inf. Syst. Eng.*, vol. 63, no. 3, pp. 277–300, Jun. 2020, doi: 10.1007/s12599-020-00649-w.
 - [21] O. M. Passos, A. C. Dias-Neto, and R. Da Silva Barreto, "Organizational culture and success in SPI initiatives," *IEEE Softw.*, vol. 29, no. 3, pp. 97–99, May 2012, doi: 10.1109/MS.2012.52.
 - [22] C. C. Shih and S. J. Huang, "Exploring the relationship between organizational culture and software process improvement deployment," *Inf. Manag.*, vol. 47, no. 5–6, pp. 271–281, Aug. 2010, doi: 10.1016/j.im.2010.06.001Get.
 - [23] O. Passos, A. Dias-Neto, R. da S. B.- ClbSE, and U. 2012, "Assessing the Relevance of Organizational Culture in Software Process Improvement Initiatives.," *researchgate.net*, 2012.
 - [24] S. D. Müller and P. A. Nielsen, "Competing values in software process improvement: A study of cultural profiles," *Inf. Technol. People*, vol. 26, no. 2, pp. 146–171, May 2013, doi: 10.1108/ITP-03-2013-0052.
 - [25] J. C. Lee, Y. C. Shiue, and C. Y. Chen, "Examining the impacts of organizational culture and top management support of knowledge sharing on the success of software process improvement," *Comput. Human Behav.*, vol. 54, pp. 462–474, Jan. 2016, doi: 10.1016/j.chb.2015.08.030.
 - [26] E. Herranz, J. G. Guzmán, A. De Amescua-Seco, and X. Larrucea, "Gamification for software process improvement: A practical approach," *IET Softw.*, vol. 13, no. 2, pp. 112–121, Apr. 2019, doi: 10.1049/iet-sen.2018.5120.
 - [27] M. Zarour, "Empirical Investigation of Factors that Hamper Pursuing Software Process Improvement: Analyses of Saudi Practitioners' Views," *J. Comput. Sci. 2018, Vol. 14, Page 122*, vol. 14, no. 1, pp. 122–131, Jan. 2018, doi: 10.3844/jcssp.2018.122.131.

BIOGRAPHIES OF AUTHORS

Dr. Zarour holds a Ph.D. in Software Engineering (2009) from– University of Quebec (Montréal, Canada) and a master's degree in Computer Science (1998) from the University of Jordan. He is currently a faculty member in the college of computer and information sciences (CCIS) at Prince Sultan University Riyadh, Saudi Arabia. He has more than 15 years of teaching and research experience in university and academic environments and has several years of industry experience in software development and quality assurance. His research interests include software process assessment and improvement, software quality and security, cost estimation, and user experience. He has many peer-reviewed publications



Dr. Mamdouh Alenezi is currently the Dean of Quality Assurance and Development at Prince Sultan University. Dr. Alenezi received his MS and Ph.D. degrees from DePaul University and North Dakota State University in 2011 and 2014, respectively. Dr. Alenezi is an associate professor in software engineering with a teaching emphasis on software engineering and software security. He participates in organizing several international scientific conferences and editorial boards of well-reputed journals. He has extensive experience in applying data mining and machine learning techniques to solve software engineering problems. He published more than 90 papers. He conducted several research areas and development of predictive models using machine learning to predict fault-prone classes, comprehend source code, and predict the appropriate developer to be assigned to a newly reported bug. His research focuses on Software Engineering, Software Security, Machine Learning, and Open-Source Software Systems. Dr. Alenezi served as Chair of the Computer Science Department, the Chief Information Technology Officer, and Dean of Educational Services before he was appointed Dean of Quality Assurance and Development in September 2020.