EPiC Series in Computing

Volume 97, 2024, Pages 20–29

Proceedings of 36th International Conference on Computer Applications in Industry and Engineering

A Process Improvement Model for Software Organizations during COVID-19 Pandemic

Hasnain Haneef[1], Assad Abbas[1], Samee U. Khan[2]

[1] Department of Computer Science, COMSATS University Islamabad, Islamabad 44000,Pakistan

[2]Department of Electrical and Computer Engineering, Mississippi State University, USA

husnain.hanif24@gmail.com, assadabbas@comsats.edu.pk,skhan@ece.msstate.edu

Abstract

Software Process Improvement (SPI) aims to achieve quality in software products for software organizations, as it helps to manage and improve the development processes. The success of software products highly depends on the right execution of software processes. The current pandemic (COVID-19) has highly affected the workflow of software organizations around the distributed geographical locations, resulting in difficulties in process execution which is a threat to software process improvement activity. The primary objective of this research is to provide a process improvement model for software development organizations for better management and improvement of the software development processes during the COVID-19 pandemic. Our proposed model is based on the objectives of the 'Team Software Process' (TSP)and 'Personal Software Process' (PSP) models to effectively manage the software development processes for both the teams and individuals involved in the remote development duringthe COVID-19 pandemic. The proposed model can also be applied in any uncertain situation other than COVID-19 to assist software organizations during remote work.

1. Introduction

Software Process Improvement (SPI) helps organizations improve their software development processes [1,2]. Improving software processes leads to better software products [3]. The SPI, according to Zahran [4], is "the discipline of defining, characterizing, improving, and measuring software management, better product innovation, faster cycle times, higher product quality, and lower development costs all at the same time". According to Khan and Keung [5], SPI was introduced to help software development companies manage processes to improve quality and efficiency. As a result, numerous SPI models have emerged, including the Software Process Improvements and Capability Determination Model (SPICE) [6,7], the ISO 9000, and Six Sigma [8]. These models and strategies can help an organization build a high-quality product while reducing the development costs and time and increasing the user satisfaction [5,9,10,11].

The current Coronavirus Disease (COVID-19) outbreak is the latest risk to global health. [12]. Currently, the COVID-19 pandemic is considered the biggest threat to humankind and is marked as themost dangerous global health disaster after World War II [13]. It has had a huge impact on how we liveand interact with one

another. Aside from the financial aspects, companies were forced to supportand (in some cases) equip office workers with the facilities to work remotely who quickly transitionedto a work-from-home set-up because of the pandemic. Organizations face challenges such as remote groups, telecommuting, lack of infrastructure, and fear of COVID-19.

The existing models of SPI [6-8] allow organizations to enhance process quality but only under normal circumstances. Existing SPI models are limited and can negatively affect software quality and efficiency, leading to decreased profitability during the uncertain conditions. Thus, to overcome the above issue, we have presented a process improvement model based on well-known software process improvement practices namely, Personal Software Process (PSP) and Team Software Process (TSP). The PSP supports engineers in measuring, optimizing, and growing their skills on a personal level as well as their planning and estimate processes [14–15]. Early TSP experience demonstrates that its application enhances engineering teams' quality and efficiency while assisting them in more closely adhering to cost and time obligations [17]. We have created a technique that software organizations can use to better manage and improve their current development activities during the current pandemic and even after that, when some emergency requires shifting to remote work instead of on-premises work, by integrating the practices of these two models with some innovations. The proposed work's primary contributions to research include a process improvement model for software organizations during the COVID-19 pandemic to assist software development organizations in remote development, particularly under uncertain conditions such as the sudden shift to remote work in any pandemic or emergency. We performed descriptive modeling to comprehend the execution of the current software development practices of the organizations. Based on expert opinion, we identified the software development challenges faced by the software organizations during the pandemic, which are subsequently addressed by the proposed model. We validated the model through the empirical study and collected feedback from industry experts about the model and performed statistical analysis of the collected data. The survey results demonstrate the effectiveness of the model. To further strengthen the analysis and to enhance the confidence in the pro- posed model, we simulated the proposed process model. The simulation model endorses the findings of the analysis conducted in the empirical study.

1.1 Motivation

SPI is an important concept for software organizations as it plays a vital role in producing a quality product and generating huge revenues and profits [3]. Several SPI models have emerged during the past years [6-8]; however, the models are designed to be executed under normal and routine circumstances because they do not deal explicitly with unstable circumstances, such as sudden shifting to remote development during the COVID-19 pandemic

Consequently, there is a need to develop a model that can explicitly deal with unanticipated situations, like the COVID-19 pandemic. To that end, in this paper we proposed a model for software organizations to facilitate them in effectively managing their remote work activitiesin the wake of COVID-19 pandemic and other similar situations where on-premises development activities are not possible.

1.2 Research questions

Our Study intend to explore the following research questions:

- RQ1: What are the challenges of process improvement to software organizations during the COVID-19 pandemic?
- RQ2: How does the proposed solution overcome the challenges of process improvement during the COVID-19 pandemic?

The rest of the paper is organized as follows: The related work is presented in section 2. The overall research methodology, which we have employed, is stated in Section 3. The proposed model, along with its explanation, is presented in section 4. Section 5 explains results and analysis,and finally, sections 6 and 7 explain threats to the validity and conclusion of the proposed work, respectively.

2 Related Work

This section highlights the previously proposed SPI models and approaches related to software process management and improvement. Additionally, some major limitations of the existing SPI models were highlighted.

Lee et al. [18] presented a study to identify determinants of successful process improvement activities in software development using a dynamic capability view approach. The authors focused on dynamic capability and its con- textual influence to identify and evaluate an organization's ability to implement software process improvement successfully. A model is presented in the proposed study to evaluate the impact of cohesion, innovation, and autonomy climates on SPI, butit still lacks some other challenges of process improvement. At the same time, no relevant criticalfactors discussed or highlighted in the proposed work can influence the success of SPI.

In the study conducted by Poth et al. [19] the main objective is to identify existing lean and agile approaches that can inherit continuous software process improvement for software development organizations. According to the authors, when using an established SPI approach, whether in a traditional or Lean/Agile, experienced SPICE or CMMI managers are expected to be much more function-specific in their solutions. Although the authors have presented a new dimension of SPI, but it is only limited to lean and agile-based software organizations.

Takahashi et al. [20] stated that a mature software process model could lead to high-quality products within the original schedule. The descriptive-analytic hierarchy process (Descriptive AHP), a new AHP model that explains the rank reversal phenomena, is used in this work to propose a methodology for improving software engineering procedures that can considerably satisfy system engineers. The waterfall model serves as the foundation of the suggested evaluationmodel. However, the authors of the proposed work do not cover some of the key SPI elements, such as process evaluation and performance measurement of the software development processes.

Sharma et al. [21] presented a study to investigate the critical factors having an impact on SPI initiatives in small and medium software organizations. The authors stated that it is important for software practitioners to have familiarity with SPI inhibitors to initiate SPI activities within an organization successfully. The primary objectives of the proposed work are to investigate the factors that affect SPI implementation initiatives in software SMEs, to synthesize the available evidence, to identify gaps in the taxonomy of the variables and to develop novel techniques to fill those gaps. Still,the proposed work lacks in dealing with the vital SPI concepts, such as management of the software development processes and improvement of the software development processes, which can ultimately affect the implementation of software process improvement.

Sun et al. [22] presented a few explicit rules to evaluate software processes to get quality products through software process improvement. The authors proposed a strategy for determiningexplicit guidelines for software process assessment based on assessment histories. Each standard issimply a combination of a subset of characteristics in an interaction execution that depicts whether theexecution is typical or unusual. Stakeholders could use the investigated rules as information andexperience to avoid future mistakes, thereby improving software process quality; they could alsobe used to gather a classifier to naturally survey future interaction execution. However, the authorshave not considered the other aspects of process improvement, for example, continuous monitoringof software development processes to successfully execute the development processes accordingto expected outputs.

Farooq et al. [23] presented a block chain-based technique to assist organizations in improving the software development processes. The approach has been developed for small and medium-scale organizations. The proposed model focuses on optimizing SPI related activities that are mainly reducing expenses of software process improvement, minimizing time consumption of process improvement activities, and reducing the resources required in SPI activities and defining a structure to allowsoftware development organizations to perform SPI related activities feasibly without changing their culture. However, the proposed work does not

assess process deviations from the expected output of thesoftware development processes, which can affect the success and failure of the SPI.

According to our best knowledge, based on an analysis of the state-of-the-art, we have found that there is no model that has been proposed in the literature that can effectively manage improvement of software development processes in any unstable and uncertain environment, such as a sudden switch toremote development during the COVID-19 pandemic.

3 Research Methodology

The research methodology adopted to conduct this research consists of seven steps, starting from analyzing the problem domain to reporting the results, as discussed below.

3.1 Selection of Problem Domain

The existing literature on SPI was explored to start the targeted research process. A comprehensive analysis of the multiple studies and challenges of SPI during the COVID-19 pandemic revealed that software development process requires more attention towards managing and improving the software development activities during unstable and uncertain circumstances requiring the shifting the operations to remote development during the COVID- 19 pandemic.

3.2 Analysis of Process Execution during COVID-19

To analyze the current software development processes, we performed descriptive modeling of the organizational processes. The industrial experts from different software organizations provided theinputs to help model the existing software processes of the organizations. In addition, we also analyzed the software organizational strategies to continue the development activities in situations like COVID- 19. We included 12 small and medium-sized enterprises (SMEs) for descriptive process modelling. One individual from each software organization participated in the modelling process. The participants asked about the nature of processes, their execution method, and their general objective. Exploring the Existing SPI Approaches. Analyzing existing studies to identify challenges, conceptual models, and practices to improve software development processes.

3.3 Identification of Challenges

Challenges of process improvement and management during COVID-19 pandemic were identified such as, process improvement model not available to manage software processes in uncertain environment. No plan to improve software processes during COVID-19 pandemic. Effective management of processes and sub-processes during the COVID-19 pandemic is difficult. Difficultto continuously monitor the processes and sub-processes, hurdles in the measurement of performance of the processes against expected output etc.

3.4 Designing the Proposed Process Improvement Model

A process improvement model is proposed to improve software development processes in unstable and uncertain circumstances.

3.5 Validation of the proposed model

For validation of the proposed process model, we have used industrial-expert based validationtechnique, paired-sample t-test for hypothesis evaluation and simulations based validation.

4 Proposed Process Improvement Model

Due to the lockdown during COVID-19, most of the software organizations like other businesses

started operations remotely. However, it is important to mention that none of the organizations had anticipated the shift and hence, were not prepared to work remotely. As a result, the productivity of the software organizations and the quality of the products affected due to the non-existence of any process model and guidelines for such situations. To that end, we propose a model that helps software organizations manage and execute their software development processes effectively in uncertain situations like the on-going pandemic to achieve their organizational goals.Interestingly, the proposed model can also be used in any uncertain situation in future as well. Figure 1 presents the proposed process improvement model. The details of the model are discussed in the sub-sections below. Figure 5 presents the proposed process improvement model.

4.1 Detailed Description of Proposed Model

After performing gap analysis and analyzing existing process execution in software organizations during the COVID-19 pandemic, following challenges were extracted and highlighted in the study:

- Managing development process and their sub-process during the remote work.
- Execution of process improvement approach during the COVID-19 pandemic.
- Continuous monitoring of software development process.

The proposed model was designed to address the challenges and limitations found during the analysis of process execution in software organizations and review of the available literatureon software process improvement. The model is mainly based on the two processes:

- Personal software process (PSP) model.
- Team software process (TSP) model.

The PSP helps engineers improve their performance and skills [14-15], while the TSP guides engineering teams in developing software-intensive products [16]. The TSP enhances quality and efficiency while adhering to cost and time obligations [17]. These models were chosen to support distant teams and team members in creating software products using process improvement techniques. Due to the COVID-19 pandemic, organizations have shifted their workflows, leading to the need for a process improvement model to optimize development processes, use fewer resources, and generate high revenues. The proposed model consists of three phases.

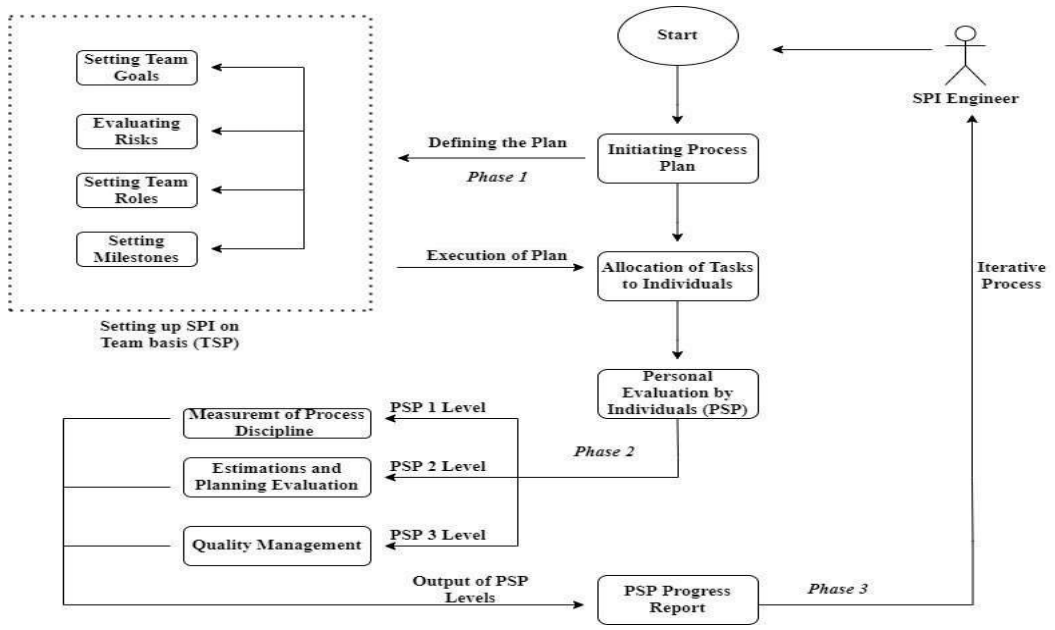

Figure 1 : Proposed Process Improvement Model

The first phase targets to plan and design schedules of tasks for the whole team. It involves team goals, evaluation of risks, setting team roles, and designing milestones. Once individualtasks are designed and allocated to team members, Phase 2 starts. This phase is based on the PSP practices. It aims to measure and evaluate the performance of individual team members.Three levels of PSP are defined in this phase. At PSP level 1, process discipline, such as finalizing plans of allocated tasks, their executions and the resources required to execute themare measured and analyzed. PSP level 2 demands the estimation of the remaining tasks of processes and evaluation of the plans and milestones. The last level of PSP is achieved by maintaining quality management details. It describes the nature of errors during the process execution, the effort and time consumed in fixing bugs, and the nature of bugs or errors that occurred during process executions. In Phase 3 of the methodology, a joint PSP report is designed and generated by a process improvement engineer. Analysis of the PSP report will help optimize the processes and generate a better TSP plan in the next iteration. Thus, thegoal of continuous process improvement will be achieved through correctly executing the proposed process improvement model.

4.2 Phases of the Proposed Model

Let us discuss the objectives of the three phases of the proposed process improvement model more concisely. Phase 1 aims to initiate the process plan for the whole team involved in the process development .Phase 2 of the proposed process improvement model targets to track the performanceof team members on an individual basis, they have to evaluate themselves at the following three levels. Phase 3 targets to prepare a PSP (personal software process) report for each individual according to the PSP levels defined in phase 2 and deliver them to the SPI engineer/leader. Each phase is based on several tasks, with each task having a particular purpose. All of the tasks involved in these three phases are explained in Table 1, Table 2, and Table 3 respectively.

If we conclude, the proposed process improvement model will enable the practitioners of software development organizations to continuously monitor the development processes through PSP plans.It will enable them to produce a software product that will be less error-prone. These software process improvement activities will ultimately enable the development organizations to avoid overcost and extra

time consumption caused by bugs and errors.

Table 1: Phase 1 of proposed model

Task	Purpose
1. Setting Team Goals	Team goals should be initiated at this step involving details of objectives to be achieved with maximum quality using available resources.
2. Evaluating Risks	All the direct and indirect risks should be well evaluated that can affect the team's performance.
3. Setting Team Roles	Team roles should be defined at this stage allocating the modules andsub-modules to different units of the team.
4. Setting Milestones	The last and crucial step of this phase is to set different milestones for theteam to get desired goals within the scheduled time.
5. Allocating Team Roles	To allocate the individuals of the team with respective roles and tasks

Table 2: Phase 2 of proposed model

Task	Purpose
1. Measurement of process discipline (PSP 0)	A baseline of the existing process should be developed, accounting for theamount of time spent on programming, the number of defects introduced and corrected, and the size of a program.
2. Estimations and planningevaluations (PSP 1)	Based on data collected at PSP 0 level, a report should be developed to estimate the actual and expected measurements for the time spent on programming, errors removing, and dealing with allocated tasks.
3. Quality Management (PSP2)	Defect prevention and removal of them are the focus at the PSP2 along with maintaining standards of the development and time spent on all quality maintaining activities.

Table 3: Phase 3 of the proposed model

Task	Purpose
1. Evaluation of PSP reports.	Evaluation of PSP reports for each individual of the team and using the analyzed data for next iteration of SPI plan.

Other than that, a TSP plan will be beneficial for the development organizations to produce a well-organized development plan based on process improvement objectives. The goal of a TSP plan is to produce and execute the development activities keeping in view the risks of current unwanted circumstances such as the COVID-19 pandemic etc. Other benefits of a TSP plan include setting milestones and maintaining schedules for the development teams.

5 Results and Analysis

Two case studies were conducted to compare the existing SPI approach and proposed process improvement model during the COVID-19 pandemic. Moreover, we used paired sample t-testand simulation-based validation techniques for further validation of the proposed work.

5.1 Hypothesis Formulation

On the theoretical groundwork mentioned earlier, we derived the following two hypotheses. The null hypothesis (H_0) states that "*Existing process improvement approaches allow managing and improving the*

software development processes during the COVID-19 pandemic". The alternate hypothesis (H_1) states that "*The proposed process improvement model assists to manage and improve the software development processes during the COVID-19 pandemic.*"

5.2 Feedback of Case Study 1 and 2 Participants

This case study aimed to validate existing process improvement approaches during the COVID-19 pandemic. Results showed that traditional Software Process Improvement (SPI) approaches failed to improve software process management and development during remote development. Experts disagreed with the claim that these approaches improved development processes. Respondents also disagreed with the statement that traditional SPI approaches helped in continuous process monitoring. They also disagreed with the ability of these approaches to help software organizations save resources like time and cost. The study concluded that existing SPI approaches did not minimize risks associated with sudden remote development shifts.

The second case study validated the proposed process improvement model in the context of the COVID-19 pandemic. The model focuses on process management, process improvement, continuous process monitoring, resource utilization, risk minimization, and performance tracking during remote development. A total of 38.23% of respondents agreed that the proposed model provides better management of software development processes during remote work. Around 85%agreed or strongly agreed that the model provides continuous process improvement. Industrial experts also agreed that the model helps utilize fewer resources, such as time and cost, during the pandemic. Additionally, 29.41% and 32.35% of respondents agreed that the model helps minimizerisks involved in developing software products during remote work. The model also helps track theprogress of processes during remote work. Statistical analysis was used to evaluate and understandthe data gathered from business professionals, transforming it into valuable information for the audience.

5.3 Paired sample t-test and Simulation based validation

The t-test [24-25] was used to assess the hypothesis, comparing data from two population groups. The paired t-test was used, as both datasets were from the same population [26-27]. The results showed a mean difference of 1.60 between the typical SPI and the suggested process model. The Ptwo-tail value was used for results evaluation, as it is crucial for determining statistical significance.The two-tail p-value was less than the 0.05 level, indicating a statistically significant difference between population means. Therefore, the null hypothesis was rejected, and an alternative hypothesis adopted.

Table 4: Paired Sample T-test for Null and Alternate Hypothesis Evaluation

N	Mean Difference	α	p-value	Null Hypothesis	Alternate Hypothesis
34	1.60	0.05	0.001	Rejected	Accepted

The proposed process model was also validated using simulations using the ExtendSim simulator. The model was designed to monitor processes and track performance on a continuous basis. Each process passed throughthe TSP and PSP phases, generating a SPI report. Data was collected from a software industry case study withtwo senior developers. The model's output showed a linear graph with straight lines, indicating that SPI reportsare directly proportional to input processes. The simulation results also endorse the results obtained throughexpert surveys for the proposed model. The model generates SPI reports at each iteration of the process, ensuring continuous monitoring and improvement.

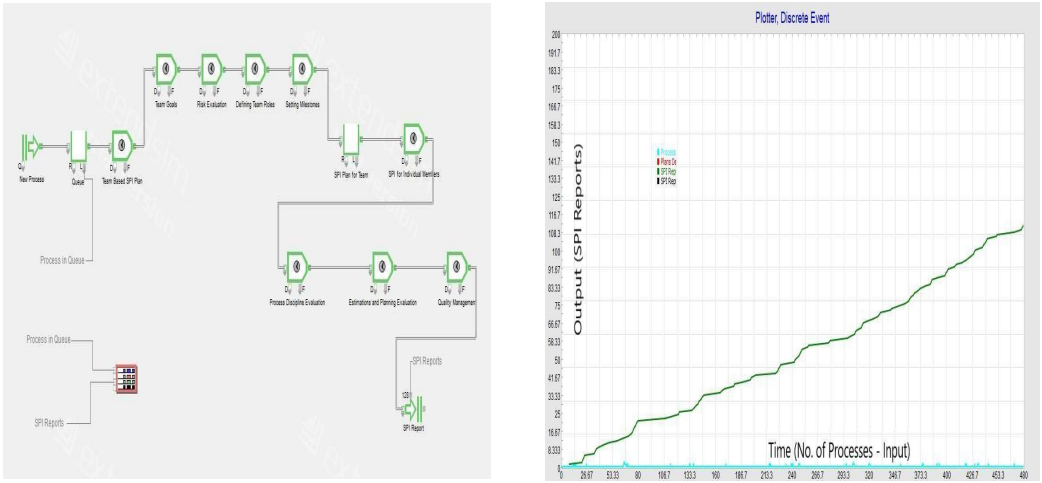

Figure 2: Simulations Assembly using ExtendSim Simulator and Output of Simulations

6 Threats to validity

Threats to Validity assesses the degree to which a study genuinely investigates the assertions made by the author [28]. In our scenario, a number of variables could affect how well the suggested technique performs. For instance, the results are based on a small group of organizations of varied sizes. As a result, depending on the size and context of the organizations, the proposed strategy may have a varying effect on software organizations. Furthermore, the findings of this study are based on an industrial evaluation of our suggested strategy from 34 participants; variations in sample size may have an impact on the findings.

7 Conclusion

Software process improvement (SPI) is crucial for software organizations to manage and improve their processes. A model has been proposed to help manage and execute development processes effectively during the pandemic. This model allows organizations to manage risks, plan iterations, monitor processes, and track developer performance. It can help achieve process improvement goalslike improved quality, productivity, revenues, and profits. The study is only applicable to small andmedium-sized software organizations but applying it to larger organizations may result in new output dimensions.

References

[1] J. A. DeFeo en J. M. Juran, Juran's quality handbook: The complete guide to performance excellence, 6thed. New York, NY: McGraw-Hill Professional, 2010.
[2] Deming, W. E. (2000). Out of the crisis (1. MIT Press ed.).
[3] S. Srinivasan en M. A. N. Murthy, "Process maturity model can help give a business an edge", SixSigma Newsletter, vol 1, pp 1–7, 2010.
[4] S. Zahran, Software process improvement: practical guidelines for business susccess.Addison-WesleyLongman Ltd, 1998.
[5] A. A. Khan en J. Keung, "Systematic review of success factors and barriers for software processimprovementin global software development", IET Softw., vol 10, no 5, pp125–135, Okt 2016.
[6] Emam, K. E. (1997). SPICE: The theory and practice of software process improvement and capability determination. IEEE Computer Society Press.
[7] A. Dorling, "SPICE: Software process improvement and capability dEtermination", Inf.Softw. Technol.,vol35,

no 6–7, pp 404–406, Jun 1993.
[8] M. J. Harry, “Six Sigma: a breakthrough strategy for profitability”, Quality progress, vol 31, 1998.
[9] M. Niazi, D. Wilson, en D. Zowghi, “A maturity model for the implementation of software process improvement: an empirical study”, J. Syst. Softw., vol 74, no 2, pp 155–172, Jan 2005.
[10] A. A. Khan, J. Keung, M. Niazi, S. Hussain, en A. Ahmad, “Systematic literature review and empirical investigation of barriers to process improvement in global software development: Client–vendor perspective”, Inf. Softw. Technol., vol 87, pp 180–205, Jul 2017.
[11] N. Ramasubbu, “Governing software process improvementsin globally distributedproduct development”, IEEE trans. softw. eng., vol 40, no 3, pp 235–250, Mrt 2014.
[12] A. S. Fauci, H. C. Lane, en R. R. Redfield, “Covid-19-navigating the uncharted”, New England Journal of Medicine, vol 382, no 13, pp 1268–1269, 2020.
[13] I. Chakraborty en P. Maity, “COVID-19 outbreak: Migration, effects on society,global environment, and prevention”, Science of the Total Environment, vol 728, 2020.
[14] Humphrey, W. S. (2000). *The personal software process (psp)*. CARNEGIE-MELLON UNIVPITTSBURGH PA SOFTWARE ENGINEERING INST.
[15] Prechelt, L., & Unger, B. (2001). An experiment measuring the effects of personal software process (PSP) training. *IEEE transactions on software engineering*, *27*(5), 465-472.
[16] Humphrey, W. S. (2000). *The team software process (TSP)*. CARNEGIE-MELLON UNIVPITTSBURGHPA SOFTWARE ENGINEERING INST.
[17] McAndrews, D. R. (2000). The Team Software Process (TSP): An Overview and Preliminary Results of Using Disciplined Practices.
[18] J.-C. Lee en C.-Y. Chen, “Exploring the determinants of software process improvement success: Adynamic capability view”, Inf. dev., vol 35, no 1, pp 6–20, Jan 2019.
[19] A. Poth, S. Sasabe, A. Mas, en A.-L. Mesquida, “Lean and agile software process improvement intraditional and agile environments”, J. Softw. (Malden), vol 31, no 1, bl e1986, Jan 2019.
[20] Takahashi, S., Komaya, K., Tamura, H. SOFTWARE PROCESS IMPROVEMENT PLANNING BY DESCRIPTIVE AHP, pp469-474, 2017
[21] P. Sharma en A. L. Sangal, “Investigating the factors which impact SPI implementation initiatives in software SMEs—A systematic map and review”, J. Softw. (Malden), vol 31, no 7, bl e2183, Jul 2019.
[22] C. Sun, J. Du, N. Chen, S.-C. Khoo, en Y. Yang, “Mining explicit rules for softwareprocess evaluation”, in Proceedings of the 2013 International Conference on Software andSystem Process - ICSSP 2013, San Francisco, CA, USA, 2013.
[23] U. Farooq, M. Ahmed, S. Hussain, F. Hussain, A. Naseem, en K. Aslam, “Blockchainbased software process improvement (BBSPI): An approach for SMEs to perform process improvement”, IEEEAccess, vol9, pp 10426–10442, 2021.
[24] E. McCrum-Gardner, “Which is the correct statistical test to use?” Brit. J. Oral Maxillofacial Surg., vol.46,no. 1, pp. 3841, 2008.
[25] N. Cressie and H. J. Whitford, “How to use the two sample t-test,” Biometrical J., vol. 28, no. 2, pp.131148, 1986.
[26] M. Xu, D. Fralick, J. Z. Zheng, B. Wang, X. M. Tu, and C. Feng, “The differences and similaritiesbetween two-sample t-test and paired ttest,” Shanghai Arch. Psychiatry, vol. 29, no. 3, p. 184, 2017.
[27] H. Hsu and P. A. Lachenbruch, “Paired t test,” in Encyclopedia of Bio- statistics,vol. 6. Hoboken, NJ, USA: Wiley, 2005.
[28] J. Siegmund, N. Siegmund, and S. Apel, “Views on internal and external validity in empirical software engineering,” in Proc. IEEE/ACM37th IEEE Int. Conf. Softw. Eng., May 2015, pp. 919.