The Impact of Critical Factors for the Successful Software Process Improvement

Mohammed Alsuwat
Department of Computer Science, King Abdulaziz University, Jeddah, Saudi Arabia

Rizwan Qureshi
Department of Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

ABSTRACT

Software enterprises should develop high-quality products to thrive in the competitive markets amid continuous changes in requirements and situations while adapting to changes in the operational environment while improving the software development process. Most organizations embrace diverse strategies, including gamification, human factors, relevant skills, experts, and organizational structure, to enhance the quality of products and sustainability of corporate operations. In this regard, human factors' lack of sufficient understanding of the software improvement processes makes it challenging to achieve the desired software process improvement (SPI) results. The researcher in this project identified SPI as one of the critical initiatives for enhancing product quality and ensuring organizations accomplish software goals. However, the attainment of effective SPI requires a set of activities and factors initiated in the software development environment. The factors constitute critical success factors (CSFs), which are actions and activities completed by the project team to achieve established goals through effective implementation of SPI. In a literature analysis, the researcher identified SPI's main setbacks in software development because organizations and projects lack root causes analysis on why projects fail, review of continuous improvement models, and inputs of experienced project managers. The embedded survey affirmed that CSFs constitute the SPI solution in modern software development processes and projects.

General Terms

Software Engineering, Software Process Improvement (SPI)

Keywords

SPI, critical success factors, survey, cumulative analysis, quantitative research

1. INTRODUCTION

Software requirements should be consistent, concise, precise, verifiable, unambiguous, and complete to avoid misunderstanding in the development process. Moreover, the production process should involve regular measurement and monitoring of progress to determine commitment to improvement, quality assurance system, validation criteria, measurement model, and conformance to project tracking and oversight. Bayona and colleagues [1] indicate that numerous methods, models, and standards are emerging in the software industry to articulate software process improvement, but challenges in the deployment are prevalent. The methods focus on the technical aspects of software development, but the main software problems or the causes of software crisis such as delay, lack of quality, and budget overrun are managerial. Therefore, popular software process improvement models do not acknowledge that improving the quality of a software product requires the enhancement of the production process rather than altering underlying components and attributes.

Software firms' world is increasingly accepting the need to reduce software development duration without quality compromise. In this regard, enterprises need to respond to customers' needs with high-quality products swiftly. Zarour and Alenezi [2] indicate that the goal is attainable by improving software development processes while prioritizing customer expectations. Software process improvement requires software developers to continually monitor development methods to ensure products utilize hardware optimally, meaning that improvement of software development practices should deliver enhanced software quality and fast time-to-market. In contrast, the failure to forge effective software process improvement leads to poor quality, reactionary organization, routinely exceeding schedules and budgets, prevalent disregard of specific processes, and a tendency to improvise processes during project execution. Therefore, all modern software organizations operating in competitive markets need to undertake software process improvement initiatives with time and cost constraints.

Software process improvement methodologies are crucial processes that incessantly advance and regulate software quality and performance development. Sun and Nazir [3] indicate that process improvement is evident in software product quality, time, and change reduction. Therefore, the primary goal of embracing software process improvement is to increase the quality of software products and make the software process highly effective through continuous evaluation. Sulayman et al. [4] report that investment in the initiative for integrating software process improvement in development cycles leads to enhanced customer satisfaction, increased organizational flexibility, better productivity, reduced time to market, and improved product quality. Therefore, organizations need to increasingly invest in identifying and adopting success factors that induce new and reliable processes and improve the existing ones. Nonetheless, software process improvement is a repetitive activity requiring time, resources, steps, and iterations to achieve significant success.

The further paper is organized as follows. Section 2 contains the related work identifying the limitations of existing studies. The problem statement and proposed solution are discussed in sections 3 and 4 respectively. Section 5 illustrates the validation of the proposed solution.

2. RELATED WORK

Client's satisfaction is an important factor to be considered in the development of the software. As argued by Albuquerque et al. [5] in "Software Process Improvement Programs: What happens after official appraisal?" at every stage, the experts involved must ensure to meet the requirements so that software can meet the specifications of the target market after the official appraisal. A solution that the authors offer includes experts ensuring they have laid out procedures to meet the specific deliverables for software process improvement.

The development of a software occurs in stages that are paramount in satisfying the needs of customers. In literature by Herranz et al. [6] “Deploying a gamification framework for software process improvement: preliminary results” they argue that since the development of software is usually sequential, the deliverables at each stage must address the specific needs of the end users. They found out that deploying a gamification framework can aid in streamlining how the software improvement process is executed. Therefore, they recommend authors to adapt gamification frameworks to ensure they meet all specifications identified within the initial stages of improvement.

According to the Almomani et al. [7], “The Factors influencing the implementation of software process improvement in small and medium enterprises: an empirical study” they found out that human factors are the most significant factors that affect the process improvement of a software within different contexts. They argue that when such factors are identified early enough, errors can be significantly reduced in the long run. The authors suggest that experts should be focused on identifying any factors that may impede their software improvement processes and offer amicable solutions at different stages. Experts should analyze the mediating effect of critical success factors to produce efficient software. Chugh et al. [8] discuss the effect of knowledge management on perceived software process improvement: Mediating effects of critical success factors and moderating effect of the use of information technology reveal that there is a need to use relevant skills in developing the software. For instance, employees should have vast computer and programming skills to develop productive and a more secure software. Therefore, poor IT skills expose software to other external risks such as cyber-attacks. According to the Georgiadou [9], “The sociocultural dimension of the Software Process Improvement Manifesto: Pilot validation by experts,” addresses the social cultural elements that influence the software improvement process. The author argues that experts should be involved in the validation of the processes at all times. As a recommendation, they suggest that to ensure minimal errors, experts should evaluate the existing deliverables and devise better approaches to adopt new changes. Herran et al. [10-11] “Deploying a gamification framework for software process improvement: preliminary results” and “Gamification for software process improvement: a practical approach”, narrate the aspect of gamification in the software improvement process. They argue that the lack of gamification may result in considerable errors when making improvements. As a solution, they recommend that there should be an effective gamification framework to guide the experts in their improvement process.

Khan et al. [12] “Understanding software process improvement in global software development: a theoretical framework of human factors”, argue that the lack of sufficient understanding of the software improvement processes by human factors makes sit challenging to achieve the desired results in the long run. The identification of the aspects that are essential in the process of software improvement creates a comfortable environment for experts to make significant changes. Thus, they propose that having a better approach to understanding the human factors influencing improvement processes can help considerably.

Khan et al. [13] “GSEPIM: A roadmap for software process assessment and improvement in the domain of global software development.”, explored the topic on the road map to assessing software improvement processes on a global scale. They argued that having a clearly set roadmap for the initiation and execution of a software process improvement road map is an essential strategy to achieving success. They recommended that software professionals should adapt the roadmaps they explored so that they can effectively execute software process improvements.

Khan et al. [14] reviewed the literature on the topic of process improvements from other authors. In the tertiary study, the authors argued that experts can use various approaches to solve process improvement problems within the software engineering industry. The adaptation of the most effective strategy would be dependent on the specific needs of an organization or other institution. As a recommendation, the authors posited that software engineers ought to explore other tertiary materials that can be guided in their process of software process improvement.

As argued by Khan et al. [15] in “Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development” argue that having a taxonomy outlining the process by which particular processes should be accomplished is essential. They recommend that experts should be more oriented towards creating taxonomic processes that can aid in their software improvements and also developments.

Lee et al. [16] in “An integrated model of the knowledge antecedents for exploring software process improvement success” explore the knowledge antecedents that are required for experts to engage in software process improvements. They argue that for a successful process implementation, there is a need to understand the knowledge descendants for different deliverables. Thus, they offer a solution of always identifying the knowledge antecedents before initiating process improvements.

Niazi et al. [17] explore the thoughts of software experts when developing software and when involved n process improvement. They posit that the process of improving software revolves around the thoughts of the experts. They have to be oriented towards introducing newer measures to create products that satisfy the needs of the consumers. Thus, they recommend that experts should explore their thoughts while coming up with the most effective solutions.

Sharma and Sangal [18] in “Building a hierarchical structure model of enablers that affect the software process improvement in software SMEs” discover the hierarchical structures that are followed when engaging in software process execution and development. They are more concerned about what the experts should focus on while trying to meet the needs of the target consumers. They suggest that at all times, experts ought to have a hierarchical structure where they put processes that to follow in the execution of the software process improvement.

Table 1. Limitations of the related work

Paper Title	Limitation
[15] Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development.	More exploration is needed on the taxonomic process of software improvement.
[12] Understanding software process improvement in global software development: a theoretical framework of human factors	More human factors need to be explored to expound the topic further.
[10] Gamification for software process improvement: a practical approach.	The approach should consider a real life case study for clarifications.
[7] Empirical study of software process improvement in Malaysian small and medium enterprises: The human aspects	More aspects than just the human should be integrated in the study
[18] Building a hierarchical structure model of enablers that affect the software process improvement in software SMEs—A mixed method approach	The hierarchical model can be expounded further to provide a detailed explanation of the processes of improvement.
[14] Systematic literature reviews of software process improvement: A tertiary study	Not enough tertiary literature sources are explored in the study
[13] A roadmap for software process assessment and improvement in the domain of global software development	The road map explored gives details about the process of software development but lacks sufficient evidence showing the improvements.
[8] The effect of knowledge management on perceived software process improvement: Mediating effects of critical success factors and moderating effect of the use of information technology	Adequate information provided on management on perceived software process improvement.
[16] An integrated model of the knowledge antecedents for exploring software process improvement success	No limitation on this article was noted.
[9] The socio cultural dimension of the Software Process Improvement Manifesto: Pilot validation by experts.	More validation that is expounded should be included
[17] What do software practitioners really think about software process improvement project success?	More reasons should be added for experts to understand the importance of software process improvement.
[7] Factors influencing the implementation of software process improvement in small and medium enterprises	More factors should be included to explore why the implementation of software improvement is essential
[5] Software Process Improvement Programs: What happens after official appraisal?	The program explained does not go into detail which is a requirement for better understanding.
[19] Approaches to strategic alignment of software process improvement: A systematic literature review	The approaches should be added to provide the experts in the design and improvement field with more information.

Vasconcellos et al. [19] explain the Approaches to strategic alignment of software process improvement. In their exploration, they found out that a number of approaches should be adopted to create an alignment of software process improvement. They recommended that experts should be oriented towards exploring all the existing alternatives to improving the process of software process improvement on a global scale.

Sharma and Sangal [20] indicate that many software development organizations find it challenging to implement a process improvement that can help them complete their project within the set budgets and time without compromising on their initial project goals.

Garousi et al. [21] associate failure in the Turkish software industry with the inability to identify and articulate CSFs, such as project monitoring and controlling, teams with experience in software development methodologies, and team's expertise in project tasks.

Arias et al. [22] indicate that 83.8% of software development projects fail while 52.7% of completed projects are delayed, over the budget estimates, or entail reduced features and functions from the specifications. Meanwhile, the leading causes of failure include technology illiteracy (4.3%), lack of

IT management (6.2%), obsoleteness (7.5%), inadequate planning (8.1%), change of requirements and specifications (8.7%), insufficient executive support (9.3%), unrealistic expectations (9.9%), inadequate resources (10.6%), lack of user involvement (12.4%), and incomplete requirements (13.1%).

According to Nasir and Sahibuddin [23], the failure to articulate CSFs is responsible for fatal errors, such as defective software in Therac-25 that caused the death of four people, bugs in the baggage handling control system that delayed the opening of Denver International Airport for 16 months, and software specification and design errors in the European Space Agency rocket that caused its explosion.

Otoom et al. [24] indicate that a project is perceived successful when it meets functional and technical specifications and the desired budget where delivery is within the scheduled deadline. Thus, the software is a vital aspect of the modern world. Table 1 depicts the limitations of related work.

3. PROBLEM DEFINITION

Software enterprises should develop high-quality products to thrive in a competitive market amid continuous changes in requirements and situations. The enterprises should adapt to changes in the operational environment while improving the software development process. In this regard, software process improvement (SPI) is a critical initiative for enhancing product quality and improving the development process. SPI constitutes the critical efforts software enterprises adopt to enhance quality. Thus, it represents a series of activities integrated into software development to improve the final quality and ensure the delivery of the finished product to the customer is timely. Meanwhile, Critical success factors (CSFs) in the software development process describe a software project’s ability to deliver customer satisfaction and achieve established goals. CSFs represent the factors that the project team must complete to achieve established goals through the effective implementation of SPI. Therefore, the attainment of SPI in software enterprises requires the identification and continuous pursuit of CSFs.

Khan et al. [13] point out some issues that may prevent software development companies from meeting project requirements, including lack of proper documentation for the entire software development process, failure to control and manage issues related to software configuration management, lack of a straightforward software development plan, and lack of effective communication. In this regard, software that fails to accomplish established objectives lacks reliable and practical strategies for enforcing SPI. Therefore, SPI is unattainable in most software development organizations due to the inability to identify CSFs and strategies for enforcing them in software projects. Arias et al. [22] posit that effective identification and implementation of CSFs, such as senior management support, qualified project managers, adequate planning, key users’ involvement, requirement´s management, monitor and control of project execution, and development team management, constrain project failure by articulating SPI throughout the development process.

Software development does not feature a perfect process, despite applying software development and software engineering methodologies [25]. Meanwhile, Hussein [26] notes that projects are not always unique but involve diverse contextual factors and varying extents of the contextual factors at different stages of the implementation process. Therefore, project success relies on selecting contextual factors depending on the implementation stage and project characteristics. In this regard, Hussein [26] indicates that the growth of databases’ size relates to increasing reported challenges and problems due to a lack of complete awareness about the project and its operational context. Arguably, most clients and management perceive software projects as deliverable efforts within a specified time, budget, and scope constraints, regardless of the context [26]. However, the failure to consider CSFs leads to unaccounted transformations, constraints, complexity, and uncertainty. In this context, identifying and implementing CSFs unique to a project contributes to the success or reduces the reported challenges’ scope. CSFs drive organizational and project success by managing it as part of projects rather than the outcome.

4. THE PROPOSED SOLUTION

The software engineering process describes the regulated and organized approach that software development organizations use to develop, design, maintain, document, and test software. Software development organizations apply combined programming, computer science, project management, cost management, and engineering principles to deliver a reliable, functional, high-quality software product. The software engineering process seeks to streamline the software development process to create a reliable, high-quality product through initial customer inception, software production, and software maintenance. In turn, SPI describes the models, planning techniques, tools, and tasks applied to improve specific metrics in the software development process. Software firms apply SPI to reduce software development costs, achieve high product quality, and improve the speed of software development. Thus, SPI is the reengineering effort that software development organizations pursue to identify and resolve inefficiencies in the software development process.

Vasconcellos et al. [19] suggest that project managers, programmers, and software engineers desire to work under SPI that is strategically aligned to separate software design specifications from the associated functional specifications. This requirement is necessary to enhance the software testing process and clearly define all parts of the software system. In this regard, SPI creates the opportunity to deliver high-quality software, reduce project times, and lower project costs. Project teams can improve their current software development practices through iterative and continuous enhancements. However, organizations must identify the current weak practices in their software processes to create new additions and pursue continuous change. SPI initiatives can succeed if the software process has well-defined components, methods, and a roadmap. The components must include regular procedures, tools, and technologies to organize and manage the software process improvement initiative.

CSFs shape the success prospects for SPI by describing and measuring practical actions that software development organizations take to improve the developmental costs, product delivery time, software quality, and other software process metrics. For instance, organizations may pursue CSFs, such as delivering software products faster or generating high-quality software products, to monitor progress towards established goals. In this regard, Venczel et al. [27] note that CSFs allow the expansion of the iron triangle to include other project aspects that enhance success rate by inducing SPI, such as resource coordination, communication, and risk

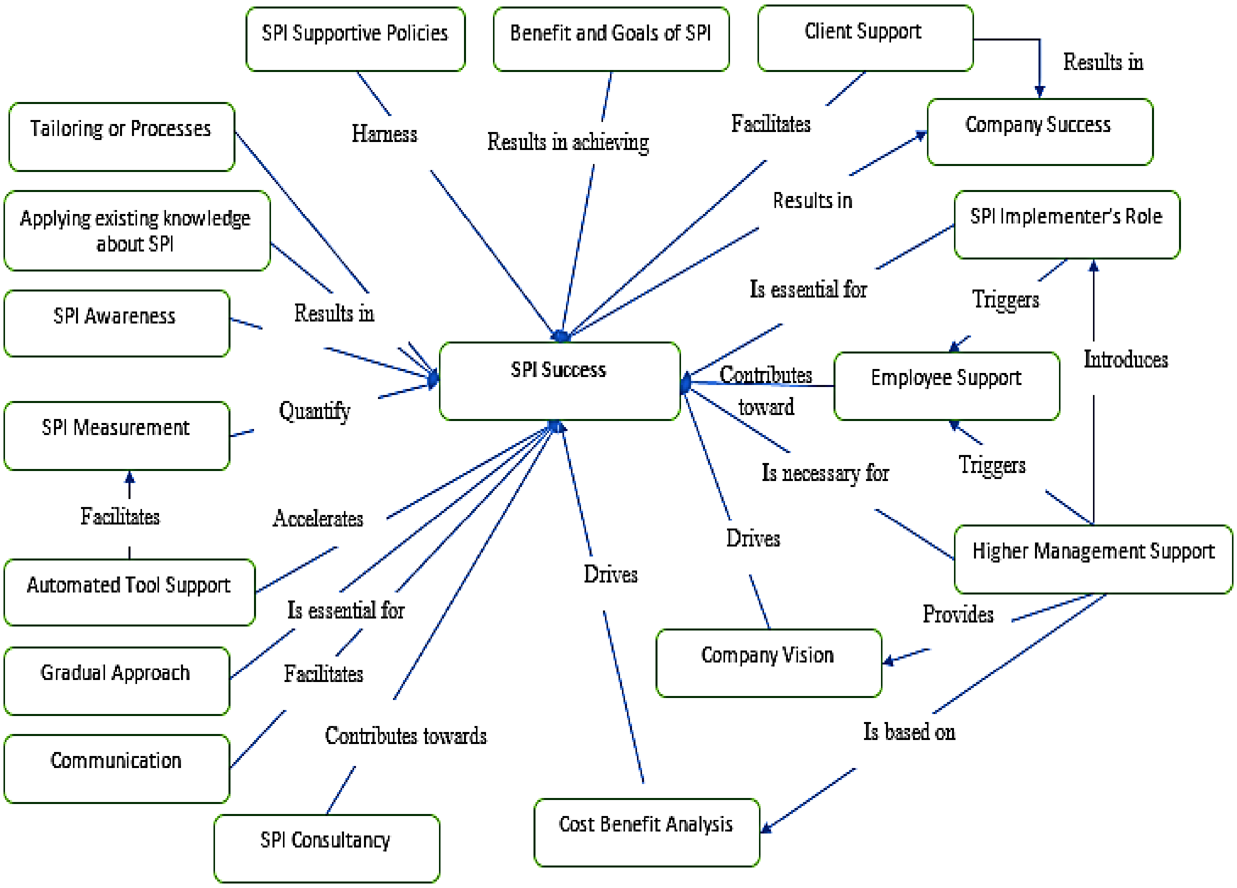

Fig 1: The effect of critical factors to achieve successful software process improvement

management. The concept focuses on enhancing SPI by monitoring change management, collaboration, quality of project team resources, relationship management, communication, project manager competencies, and project management practices and performance [27]. Consequently, software organizations institute project teams, project managers, clear and measurable project goals, top management support for the project, and authority of the project manager, as CSFs.

In this project, the introduction of CSFs will induce the SPI required to accomplish software engineering and management. Vrchota et al. [28] posit that CSFs articulate and monitor organizational sustainability, stakeholders' demands, and project life cycle, shifting project focus from time, quality, and budget to economic, environmental, and social long-term priorities. In this regard, software developers and project managers will increasingly concentrate on supporting the next generation of users or requirements rather than meeting deadlines while focusing on global wellness rather than the scope from project elements. Therefore, software projects will involve regular evaluation of failures, time pressure, and customer satisfaction with products besides the team's motivation and stimulation during software development [28]. Integrating CSFs will be eliminating project problems, employing project managers with the right qualifications, correcting financial budgets, endorsing effective communication channels, and supporting top management. Nonetheless, attaining sustainable SPI will require consideration of CSFs in all aspects of the software project lifecycle, including external environment, project implementation, project work procedures, project factors, and human factors.

Modern software development processes face unique challenges due to the diverse demand and dynamism in complexity and size. Garousi et al. [21] indicate that lean and agile software development models are reliable for inducing SPI but not comprehensive because software development projects continue to exhibit significant failure rates than traditional software projects. In this regard, Garousi et al. [21] posit that software developers and project managers need to enhance SPI by introducing CSFs, such as software development methodology, organizational culture and management style, project planning and controlling, customer skill, training and education, team building and team dynamics, and performance, usability, security, compatibility, maintainability, and transferability. The consideration of team, organizational, contingency, and customer factors foster SPI, which guarantees product-related factors, process success, and stakeholders' satisfaction. Therefore, CSFs provide the software development process with a multidimensional approach to project success and sustainability.

The attainment of a reliable and sustainable SPI requires identifying and considering numerous CSFs. Arias et al. [22] indicate that stakeholders in software should monitor CSFs, such as senior management support, adequate planning, key users' involvement, requirement management, and development team management. Nasir and Sahibuddin [23] recommend a realistic schedule, clear objectives and goals, and precise requirements and specifications that can be monitored through Personal Software Process, Proxy-Based Estimation, Delphi technique, Function Point Analysis, and

Constructive Cost Model. Otoom et al. [24] report that essential CSFs to consider in SPI based on software project organization structure includes leadership, team capability, and the right team. Vrchota et al. [28] posit that crucial CSFs for enhancing management of Industrial 4.0 projects entail finances, employees and flexibility, and leadership and experiences. Meanwhile, Sobieraj and Metelski [29] report that the CSFs that guarantee SPI includes experienced project team, client engagement, project value and uniqueness, information and communication channels, and planning and reviews. Chow and Cao [25] identified 36 CSFs surrounding processes, people, project, technical, organizational software development aspects, while Hussein [26] categorized CSFs depending on project characteristics into constraints, organizational complexity, uncertainty, transformation, and impact on business. In this regard, this SPI project involved CSFs with an impact on SPI or other CSFs, as illustrated in Figure 1. Software firms can effectively identify the impact of considering the effect of a given CSF on SPI and the project environment.

5. VALIDATION

The researcher validated the new theory to reflect the actual experience of software engineers with CSFs through an online survey. The researcher included a cover letter detailing the purpose of the survey, the specificity of required participants, and the need to understand the principles behind CSFs in SPI before accessing and answering the validation questionnaire. The researcher explained to the software engineers the improved conceptual framework for software engineering practices to institute successful SPI software projects.

The researcher utilized a sample with 40 respondents to validate the results. The participants in the questionnaire were academicians, software engineers, programmers, project managers, IT managers, and software designers working in different organizations distributed across Europe, Asia, Africa, and America. The researcher utilized an online questionnaire, which was shared with potential respondents through social media, including Facebook, Twitter, WhatsApp, Telegram, and LinkedIn. The distribution of the questionnaire and identification of respondents entailed snowballing sampling, where each respondent was allowed to invite other potential participants.

The questionnaire involved likert scale questions with five levels, 5 as strongly disagree, 4 as disagree, 3 as neutral, 2 as agree, and 1 as strongly agree. The questionnaire had 26 questions for evaluating the roles of CSFs in SPI.

Goal 1: the proposed SPI framework is efficient and effective in software development processes

Goal 2: the implementation of CSFs enhances SPI and software environment.

Goal 3: SPI is reliable and highly practical compared to the conventional management of software development

Goal 4: software development processes and firms should embrace CSFs in SPI

The researcher interpreted and condensed themes from the survey. The emergent themes were used to validate the SPI framework. The key dimensions and theoretical typologies reported by the participants were identified, mapped, and classified appropriately to conceptualize the validity and applicability of the SPI framework. The researcher utilized pie charts to report the findings from the question.

5.1 Cumulative Analysis of Goal 1

The researchers analyzed the explanatory accounts presented by the participants for the goal 1 as shown in figure 2. Figure 2 shows that The researcher assessed the role of CSFs in the enhancement of the software development process through stimulation of SPI. The cumulative findings show that 33% of the respondents strongly agree, 47% agree, 7% express neutrality, 9% disagree, and 4% strongly disagree that SPI framework is efficient and effective in software development processes.

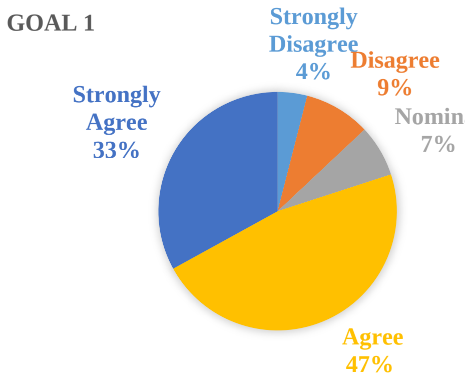

Fig 2: Cumulative analysis of goal 1

5.2 Cumulative Analysis of Goal 2

The researcher assessed the role of CSFs in the enhancement of the software development process through stimulation of SPI for goal 2. SPI needs to be advantageous to encourage replacement of the established software processes as shown in figure 3. Figure 3 indicates that 25% of the respondents strongly agree, 59% agree, 7% nominal, 7% disagree, and 2% strongly disagree that CSFs are responsible for improving the software development process by articulating SPI.

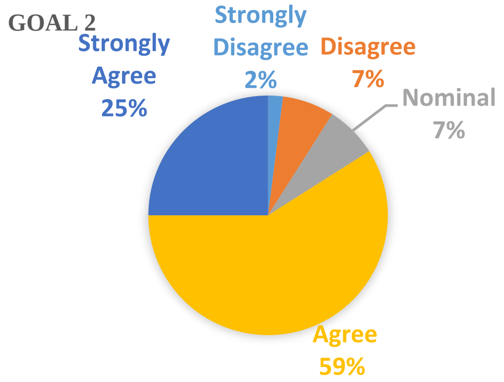

Fig 3: Cumulative analysis of goal 2

5.3 Cumulative Analysis of Goal 3

In goal 3, the researcher assessed the benefits of SPI compared to the conventional software development processes. SPI needs to be advantageous to encourage replacement of the established software processes as shown in figure 4. According to figure 4, 24% of the respondents strongly agreed, 63% agreed, 8% disagreed, and 5% expressed neutrality that SPI is highly beneficial to software projects and development processes compared to the conventional software development processes.

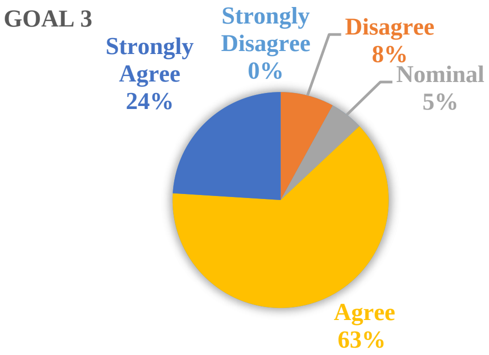

Fig 4: Cumulative analysis of goal 3

5.4 Cumulative Analysis of Goal 4

Goal 4 evaluated the feasibility of software development processes and firms embracing CSFs in SPI. With minimal switching costs and guaranteed benefits, software stakeholders will embrace CSFs in SPI as shown in figure 5. Figure 5 displays that 52% of the survey participants strongly agreed, 39% agreed, 2% were nominal, 4% disagreed, and 3% strongly disagreed that software development processes and firms should embrace CSFs in SPI.

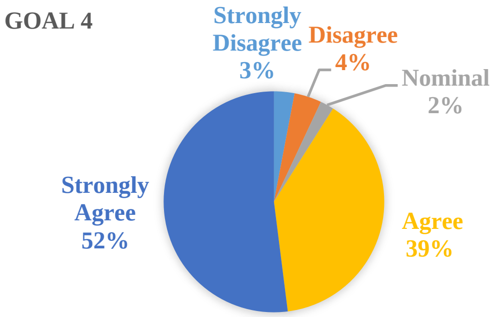

Fig 5: Cumulative analysis of goal 4

5.5 Final Cumulative Analysis of 4 Goals

Figure 6 displays the final cumulative analysis of four goals about the role of critical factors to achieve successful SPI in a software company. According to figure 6, 86% of the respondents agree, 9% disagree and 5% remain neutral. The results of survey clearly support the proposal of this research.

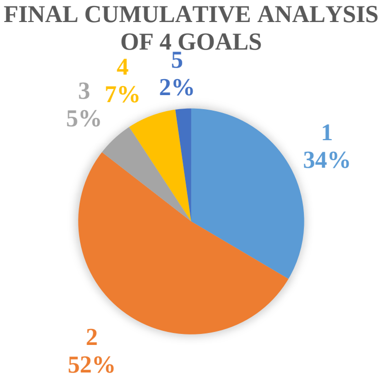

Fig 6: Final cumulative analysis of 4 goals

6. CONCLUSION

SPI has a critical role in software organizations, including meeting specific business goals and enhancing the operational environment. SPI enables firms to deliver software products faster into the market or produce high-quality software products. Thus, the concept represents the models, planning techniques, tools, and tasks applied to improve specific metrics in the software development process. Software firms apply it to reduce software development costs, achieve high product quality, and improve the speed of software development. The attainment of effective SPI requires a set of activities and factors initiated in the software development environment. For instance, some software firms embrace gamification, human factors, relevant skills, experts, and organizational structure that constitute CSFs to articulate SPI. CSFs represent the factors that the project team must complete to achieve established goals through the effective implementation of SPI. Therefore, software firms must identify and articulate CSFs to achieve guaranteed SPI. Software development environment futures numerous factors that challenge success. In this regard, most software firms fail to initiate proper documentation for the entire software development process, control and manage issues related to software configuration management, prepare a clear software development plan and articulate effective communication. Thus, the absence of analysis of root causes why projects fail, continuous improvement models' review and experienced project managers' inputs ensure organizations or software projects do not identify CSFs required for SPI. As a result, customers complain of late projects, making them unreliable and hard to maintain, while the software projects delivered outside the customer requirement exhibit poor performance and low acceptance rate. In this context, the identification and articulation of CSFs in software projects guarantee SPI. CSFs expand the iron triangle by including different project aspects into the software development process, such as resource coordination, communication, and risk management, inducing SPI. CSFs articulate and monitor organizational sustainability, stakeholders' demands, and project life cycle, shifting project focus from time, quality, and budget to economic, environmental, and social long-term priorities. Hence, a survey involving different stakeholders in the software industry validates an SPI framework that emphasizes CSFs as the primary tool for articulating SPI. CSFs constitute the SPI solution in modern software development processes and projects.

7. REFERENCES

[1] S. Bayona S., J.A. Calvo-Manzano, and T. San Feliu, "Critical Success Factors in Software Process Improvement: A Systematic Review", Communications in Computer and Information Science, 2012.

[2] M. I. Zarour, and M. Alenezi, "Software process improvement initiative in medium size IT organization: A case study", Indonesian Journal of Electrical Engineering and Computer Science, 2021.

[3] L. Sun, and S. Nazir, "Multicriteria Decision Making to Continuous Software Improvement Based on Quality Management, Assurance, and Metrics", Scientific Programming, 2021.

[4] M. Sulayman, C. Urquhart, E. Mendes, and S. Seidel, "Software process improvement success factors for small and medium Web companies: A qualitative study", Information and Software Technology, 2012.

[5] Albuquerque, R., Malucelli, A., and Reinehr, S.S. 2018. Software Process Improvement Programs: What happens after official appraisal?. In Proceedings of the 30th Conference on Software Engineering and Knowledge

Engineering

[6] Herranz, E., Colomo-Palacios, R., and Al-Barakati, A. 2017. Deploying a gamification framework for software process improvement: preliminary results. In Proceedings of the European Conference on Software Process Improvement

[7] M.A. Almomani, S. Basri, and A. Kamil, “Factors influencing the implementation of software process improvement in small and medium enterprises: an empirical study”, Advanced Science Letters, 2018.

[8] M. Chugh, N. Chanderwal, A.K. Mishra, and D.K. Punia, “The effect of knowledge management on perceived software process improvement: Mediating effects of critical success factors and moderating effect of the use of information technology”, VINE Journal of Information and Knowledge Management Systems, 2019.

[9] E. Georgiadou, K. Siakas, E. Berki, J. Estdale, H. Rahanu, and M. Ross, “The sociocultural dimension of the Software Process Improvement Manifesto: Pilot validation by experts”, Journal of Software: Evolution and Process, 2020.

[10] Herranz, E., Colomo-Palacios, R., and Al-Barakati, A. 2017. Deploying a gamification framework for software process improvement: preliminary results. In Proceedings of the European Conference on Software Process Improvement

[11] E. Herranz, J.G. Guzmán, A. de Amescua-Seco, and X. Larrucea, “Gamification for software process improvement: a practical approach”, IET Software, 2019.

[12] A.A. Khan, J. Keung, S. Hussain, M. Niazi and M. Tamimy, “Understanding software process improvement in global software development: a theoretical framework of human factors”, ACM SIGAPP Applied Computing Review, 2017.

[13] A.A. Khan, J. Keung, M. Niazi, S. Hussain, and M. Shameem, “GSEPIM: A roadmap for software process assessment and improvement in the domain of global software development”, Journal of software: Evolution and Process, 2019.

[14] Khan, A. A., Keung, J., Niazi, M., Hussain, S., and Zhang, H, “Systematic literature reviews of software process improvement: A tertiary study,” In Proceedings of the European Conference on Software Process Improvement

[15] A.A. Khan, M. Shameem, R.R. Kumar, S. Hussain, and X. Yan, “Fuzzy AHP based prioritization and taxonomy of software process improvement success factors in global software development”, Applied Soft Computing, 2019.

[16] J.C. Lee, Y. C. Shiue, and C. Y. Chen, “An integrated model of the knowledge antecedents for exploring software process improvement success”, Journal of Enterprise Information Management, 2020.

[17] M. Niazi, A. Mishra, and A. Q. Gill, “What do software practitioners really think about software process improvement project success? an exploratory study”, Arabian Journal for Science and Engineering, 2018.

[18] P. Sharma, and A. L. Sangal, “Building a hierarchical structure model of enablers that affect the software process improvement in software SMEs—A mixed method approach”, Computer Standards & Interfaces, 2019.

[19] F.J. Vasconcellos, G.B. Landre, J.A.O., J.L. Oliveira, R.A. Ferreira, and A.M. Vincenzi, “Approaches to strategic alignment of software process improvement”, A systematic literature review. Journal of systems and software, 2017.

[20] P. Sharma, and A.L. Sangal, “Building a hierarchical structure model of enablers that affect the software process improvement in software SMEs—A mixed method approach”, Computer Standards & Interfaces, 2019.

[21] V. Garousi, A. Tarhan, D. Pfahl, A. Coşkunçay, and O. Demirörs, “Correlation of critical success factors with success of software projects: An empirical investigation”, Software Quality Journal, 2019.

[22] G. Arias, D. Vilches, C. Banchoff, I. Harari, V. Harari, and P. Iuliano, “The 7 key factors to get successful results in the IT Development projects”, Procedia Technology, 2012.

[23] M.H. Nasir, and S. Sahibuddin, “Critical success factors for software projects: A comparative study”, Scientific Research and Essays, 2011.

[24] A.F. Otoom, G. Al Kateb, M. Hammad, R.J. Sweis, and H. Hijazi, “Success Factors Importance Based on Software Project Organization Structure”, Information, 2019.

[25] T. Chow, and D.B. Cao, “A survey study of critical success factors in agile software projects”, The Journal of Systems and Software, 2008.

[26] B. Hussein, “The influence of project characteristics on project success factors. Insights from 21 real life project cases from Norway”, Procedia Computer Science, 2019.

[27] T.B. Venczel, L. Berényi, and K. Hriczó, “Project Management Success Factors”, Journal of Physics: Conference Series, 2021.

[28] J. Vrchota, P. Rehoˇr, M. Maˇríková, and M. Pech, “Critical Success Factors of the Project Management in Relation to Industry 4.0 for Sustainability of Projects”, Sustainability, 2021.

[29] J. Sobieraj, and D. Metelski, “Quantifying Critical Success Factors (CSFs) in Management of Investment-Construction Projects: Insights from Bayesian Model Averaging”, Buildings, 2021.