Analysis of the use of software process improvement models in agile development

Análisis del uso de modelos de mejora de procesos de software en el desarrollo ágil

Héctor Cornide-Reyes[1]* https://orcid.org/0000-0002-9130-184X
Gianina Madrigal[1] https://orcid.org/0009-0009-3399-4738
Guisselle Muñoz[1] https://orcid.org/0009-0000-5597-5380
Cristian Duran[1] https://orcid.org/0009-0007-9190-6287
José Jorquera[1] https://orcid.org/0009-0008-1707-6163
Jenny Morales[2] https://orcid.org/0000-0001-9940-0490

Recibido 5 de abril de 2024, aceptado 28 de mayo de 2024
Received: April 05, 2024 Accepted: May 28, 2024

ABSTRACT

Nowadays, with global software development, software process improvement models have again focused the scientific community's attention. The need to increase the probability of success in these projects appears as the main reason to use a software process improvement model. In this article, a quick literature review was conducted to find evidence of using traditional process improvement models in agile software development. A total of fifteen primary studies were meticulously selected, from which significant insights were extracted for this research. This work's main contribution is identifying the challenges of using a software process improvement model. In conclusion, despite the evidence on the use of models, there is still a need to facilitate their integration into agile models such as Scrum.

Keywords: Software process improvement, software engineering, global software development, SPI, agile.

RESUMEN

En la actualidad, con el desarrollo global de software, los modelos de mejora de procesos de software han vuelto a centrar la atención de la comunidad científica. La necesidad de aumentar la probabilidad de éxito en estos proyectos aparece como la principal razón para utilizar un modelo de mejora de procesos de software. En este artículo, hemos realizado una revisión de la literatura para encontrar evidencias respecto al uso de los modelos tradicionales de mejora de procesos en el desarrollo ágil de software. Seleccionamos 15 estudios primarios de los que pudimos extraer información valiosa para nuestra investigación. La principal contribución de este trabajo es la identificación de los retos que existen cuando se utiliza un modelo de mejora de procesos de software. En conclusión, podemos afirmar que, a pesar de la evidencia sobre el uso de modelos, todavía existe la necesidad de facilitar su integración en modelos ágiles como Scrum.

Palabras clave: Mejora de procesos de software, ingeniería de software, desarrollo global de software, SPI, ágil.

[1] Universidad de Atacama, Departamento de Ingeniería Informática y Ciencias de la Computación. Copiapó, Chile. Email: hector.cornide@uda.cl; gianina.madrigal@uda.cl; guisselle.munoz@uda.cl; cristian.duran@uda.cl; jose.jorquera@uda.cl
[2] Universidad Católica del Maule, Departamento de Economía y Administración. Talca, Chile. Email: jmoralesb@ucm.cl
* Autor de correspondencia: hector.cornide@uda.cl

INTRODUCTION

Software Process Improvement (SPI) is a constant challenge in the software development industry [1]. Organizations strive to find effective ways to optimize their processes for greater efficiency, quality, and customer satisfaction. However, this task has difficulties and obstacles that must be overcome [2], [3].

For several years, the SPI kept the scientific community's attention as an object of study to improve the quality of the software product and, consequently, increase the project's probability of success. In this sense, several organizations were tempted to adopt one of the models or standards that pointed to SPI. Within this group, the most widely studied and applied models/standards are MoProSoft [4], EvalProSoft [5], CompetiSoft [6]; Project SPICE [7]; Metrica V3 [8], and CMMI [9].

The implementation of the Process Model for the Software Industry (MoProSoft) in software process improvement (SPI) has proven to be an effective solution for software development companies [10]. By using MoProSoft in conjunction with EvalProSoft, companies can increase their project management, reduce costs, plan more effectively, and obtain a detailed and accurate evaluation of their processes. EvalProSoft, specifically designed to evaluate MoProSoft implementation from the evaluator's point of view, offers a structured and systematic approach to evaluate and confirm the execution of business processes. This technique proposes metrics and evaluation criteria that allow measuring the capacity and maturity level of the implemented processes. By collecting data, reviewing documentation, and conducting interviews with process managers and work teams, EvalProSoft generates documentary and oral evidence that supports attribute ratings and rating profiles. MoProSoft also allows for better process control, reduced stress and effort for the staff, and improved software quality. Using the CompetiSoft methodological framework can have several advantages for small software companies, such as improving software quality, reducing costs and development times, and increasing customer satisfaction. The methodological framework is based on action research and case studies, allowing companies to tailor the framework to their specific needs and get practical feedback on its application [10].

CMMI provides a common language and framework for process improvement that can be used across different industries and disciplines. Compared to models created independently, it is designed to serve better users of separate Capability Maturity Models (CMMs) and other models. CMMI combines the essential elements of different existing models into a unified approach. This procedure allows industries not explicitly related to software and systems engineering to use process improvement techniques that have been successful in the software engineering community. These essential elements comprise the "core" of CMMI, which focuses on project and process management. By building on this core, industries outside of software and systems engineering can adapt and build on it to suit their disciplines. They can take advantage of techniques and approaches proven in the software engineering community and apply them in their contexts and processes [11].

Metrica V3 is a widely used method to improve software development processes. Its systematic, data-based approach provides guidelines and recommendations for measuring and improving the quality and productivity of these processes. This method identifies relevant metrics that evaluate software processes' performance and establish improvement objectives. Systematic planning, measurement, analysis, and processes improvement are carried out through a set of practices and activities, using concrete data as a basis for decision-making [8]. Metrica V3 is adaptable to the specific needs of each organization. It can be applied at different stages of the software life cycle, ranging from planning and estimation to delivery and maintenance. Its main objective is to provide a solid basis for evaluating and improving the quality of software products and services and optimizing resources and process performance.

SPICE (ISO/IEC 15504) is an internationally recognized standard that provides a structured and systematic framework for evaluating and improving software processes. It allows for identifying areas for improvement, compare performance with others and establish a solid foundation for continuous improvement. Thus, it enables the identification of strengths and weaknesses, set improvement goals, and make informed decisions to optimize the quality and efficiency of their software processes [12].

With the advent of agile methods, agile software process improvement (agile SPI) has emerged as a promising approach to address process improvement issues in traditional software development [4], [5]. Agile SPI combines agile principles and practices with the goals of continuous improvement, enabling organizations to adapt quickly to change and constantly improve their processes.

The importance of agile SPI lies in its ability to address the limitations of traditional process improvement approaches and reap the benefits of agile approaches. A quick literature review [13] can identify the most relevant and significant aspects of the agile SPI, providing a solid base to understand its importance and practical implications.

This article aims to analyze the scientific evidence that informs about the challenges associated with using traditional software process improvement models in agile software development. Following this objective, a quick literature review was conducted to understand these issues better. A search string was developed to carry out this review, while the databases considered were WoS, Scopus, ACM Digital Library, and IEEE Xplore. Fifteen primary studies were selected and analyzed, obtaining quite encouraging results.

This article continues as follows. The section "Traditional Models of Software Process Improvement" presents the main traditional models of the software improvement process. section "Methodology" describes the methodology used to conduct the literature review. The section "Discussion of The Results," analyzes and discusses the findings. Finally, it presents the main conclusions of this work.

TRADITIONAL MODELS OF SOFTWARE PROCESS IMPROVEMENT

As mentioned in the section "Introduction," software process improvement has always held the attention of the scientific community due to the demands of the current times regarding improving product quality or increasing productivity. Next, the fundamentals of the so-called traditional SPI models will be provided.

Moprosoft, Competisoft, and Evalprosoft

The Mexican software process improvement models MoProSoft, CompetiSoft, and EvalProSoft are government initiatives designed to promote continuous improvement of software development processes in Mexican organizations. Each of them is briefly described below:

- MoProSoft [4]: Process Model for the Software Industry. MoProSoft is a software process improvement model designed to help companies improve their software development processes. It is based on international and national standards, and its main objective is to increase quality and productivity in software development. MoProSoft has three maturity levels, focusing on project management, software development, maintenance, and testing.
- CompetiSoft [6]: Competitiveness Model in Software Development. CompetiSoft is a software process improvement model that focuses on improving the competitiveness of Mexican software companies. It is based on the international model Capability Maturity Model Integration (CMMI) and the ISO/IEC 12207 standard. CompetiSoft is designed for companies of different sizes and has four maturity levels.
- EvalProSoft [5]: Software Process Evaluation Model. EvalProSoft is a software process evaluation model designed to assess the maturity of software development processes in Mexican companies. It is based on the international CMMI model and the ISO/IEC 15504 standard. EvalProSoft uses a level-based evaluation approach and has four maturity levels.

Based on recognized international and national standards, these models provide a structured and gradual approach for organizations to evaluate, improve, and optimize their processes, allowing them to achieve higher quality, productivity, and competitiveness in the software market.

METRICA V3

Metrica V3 (Software Process Improvement Metric, Version 3) is a model used in software process improvement [2]. Metrica V3 is a process improvement methodology that provides guidelines and recommendations to measure and improve the quality and productivity of software development processes.

Metrica V3 focuses on identifying relevant metrics to evaluate the performance of software processes and establish improvement objectives. It provides a set of practices and activities to plan, measure, analyze, and improve processes based on a systematic and data-driven approach [8]. This process improvement methodology aims to provide organizations with a solid foundation for evaluating and improving the quality of software products and services and optimizing resources and process performance.

Metrica V3 is used in various contexts and organizations in the private and public sectors. It can be tailored to each organization's specific needs and applied to different stages of the software life cycle, from planning and estimating to delivery and maintenance [8].

Like SPI, Metrica V3 has several advantages and disadvantages. Regarding the benefits, its ability to provide a systematic and quantitative evaluation and measurement of software quality stands out. It enables development teams to identify and fix problem areas, significantly improving software efficiency and reliability. Additionally, it facilitates communication between team members and stakeholders, thus promoting an alignment of expectations and informed decision-making [8]. However, some disadvantages associated with Metrica V3 must also be considered. One of them is its potential complexity regarding its implementation and correct use since it requires careful planning, accurate data collection, and exhaustive analysis, which can be expensive and time-consuming. It is critical to understand both the advantages and limitations of this metric and use it as an integral part of a broader approach to developing high-quality software [14].

CMMI

CMMI (Capability Maturity Model Integration) is widely used in software process improvement. CMMI is an internationally recognized process improvement model that provides detailed guidance for evaluating and improving organizational processes regarding maturity and capability. This model offers a structured and systematic approach to identifying areas for improvement and establishing clear goals to raise the maturity and capability of an organization's processes. It provides a set of recommended practices and process areas that cover critical aspects of software development, project management, configuration management, and risk management, among others [8], [9]. Organizations looking to improve their software processes and reach higher levels of maturity and capability often look to CMMI as a reference and framework to guide their improvement efforts. CMMI adoption involves evaluating current processes, identifying areas for improvement, setting improvement goals, and implementing practices and actions to achieve those goals.

CMMI has become a de facto standard in the software development industry and is used by many organizations, both in the private and public sectors. However, it is essential to highlight that adopting CMMI requires a significant investment of time, resources, and effort since it implies a cultural change and the implementation of existing processes [15].

Many companies and organizations use CMMI as a vehicle for SPI. However, some have raised concerns about the upfront costs and positive corporate bottom-line effects of implementing a CMMI-based software process improvement program [16]. It is an internationally recognized model that provides a detailed guide for evaluating and improving organizational processes regarding maturity and capacity.

Proyecto Spice (ISO/IEC 15504)

Project SPICE (Software Process Improvement and Capability Determination) is an internationally recognized software process improvement model. SPICE provides a structured and systematic framework for evaluating and improving software processes in organizations [17].

Smaller companies face significant challenges when considering the implementation of SPICE [18]. First, they face a very high cost associated with implementing and meeting the standard's requirements. Additionally to the price, implementing the standard involves considerable documentation and bureaucracy. Small companies, often with small teams and limited resources, may find the documentation process required by the standard too cumbersome and distract from their core software development activities.

Another challenge is that SPICE does not clearly state the software processes that must be followed. It provides a general framework for process evaluation and improvement but does not offer specific guidance on implementing software processes effectively. This can make it even more challenging for small businesses seeking clear, actionable guidance on improving their software development practices. For these reasons, SPICE is not widely used in the software development industry. Although the SPICE model can be useful in certain contexts and specific organizations, its adoption is not as widespread as other process improvement models, such as CMMI or Metrica V3.

Using a traditional process improvement model depends on several factors, such as the organization's objectives, the project's context, and industry preferences [19]. It is essential to conduct a thorough analysis to determine which process improvement model is the most suitable for an organization and adapt it to its specific needs. Table 1 summarizes each improvement process model according to its characteristics.

METHODOLOGY

The main objective of this work is to analyze the scientific evidence that informs about the challenges associated with using traditional software process improvement models in agile software development. A literature review process was carried out based on the guidelines proposed by [20] to achieve this objective, and it is consistent with other recently published rapid literature reviews [21], [22]. The process was conducted through the following steps: 1) statement of the research questions, 2) search process, 3) selection of studies, and 4) analysis of results. All these steps are described below.

Research questions

The research questions that drive this literature review are:

Q1: What are the challenges when incorporating an SPI model into agile software development?

Q2: What traditional process improvement models have been used in agile software development?

Search process

The databases used were WoS, Scopus, ACM Digital Library, and IEEE Xplore. This selection is mainly due to the reputation of these databases in the discipline and the fact that we have full access to the published material.

The search string used the following concepts and denominations:

- SPI, "software process improvement".
- Agile scrum, asd, "agile software development," Kanban, "lean software development," lsd, xp, "extreme programming," Lean, XP, "xtreme programming."
- moprosoft, competisof, evalprosoft.
- "Metrica V3", "metricaV3."
- "ISO/IEC 15504," "Spice project."
- CMMI, "Capability Maturity Model Integration."

The results of applying the search string in each database engine were Web of Science: 23; Scopus: 255; and Google Scholar: 349. The inclusion/exclusion criteria were as follows: Articles with the highest number of citations will be selected; the last ten years will be explored. The articles must be from the Computer Science area in the English language. After applying these inclusion/exclusion criteria, 11 articles from Web of Science, 52 from Scopus, and 63 from Google Scholar passed the selection stage.

Selection of primary studies

With the set of articles obtained, a first filter was applied by reading the abstracts of each one to discard those that had a different orientation. The second filter consisted of reading of the articles thoroughly. Subsequently, 13 primary studies were

Table 1. Summary of characteristics identified in the SPI models.

Characteristic criteria	MoProSoft, CompetiSoft and EvalProSoft	Metrica V3	CMMI	Project SPICE
Oriented to companies	Small and Medium	Small and Medium	Big	Big
Industry presence	Low	Low	High	Medium
Level of the complexity of use	Low	Medium	High	High
Experience and knowledge required	Medium	Medium	High	High
Resources to implement	Low	Low	High	High

selected for analysis and discussion (3 in Web of Science; 6 in Scopus; and 4 in Google Scholar).

Besides the articles found, 2 doctoral theses were added through a manual search. The authors considered their incorporation relevant to answer the research questions posed. Finally, a total of 15 scientific articles were analyzed. Table 2 shows the titles of the selected primary studies, type of article, year of publication and DOI.

General description of the results

Once the primary studies have been selected, we complete a form to conduct a first analysis of the results. From the list, we observe that 46% of the articles are published in some journal of the discipline, 40% in conference proceedings, and 14% in doctoral theses. Regarding the traditional SPI models, CMMI had the most significant presence in the articles (73%), followed by SPICE with 40% of the articles and MoProSoft with 20%. Metrica V3 was the least mentioned traditional SPI model in primary studies. Regarding the research method, 40% of the articles used a mapping or systematic literature review to carry out the research, while 33% used a case study to propose new SPI models [23], [25], [29], [31], [34].

Limitations

For this research, we have focused on the main scientific article databases, WoS and Scopus, and

Table 2. Primary studies selected.

ID	Name	Type	Year	DOI/ISBN
[23]	Application of a software agility assessment model – AgilityMod in the field	Journal	2019	10.1016/j.csi.2018.07.002
[24]	How has SPI changed in times of agile development? Results from a multi-method study	Journal	2019	10.1002/smr.2182
[25]	Scrum+: A scaled Scrum for the agile global software development project management with multiple models	Journal	2019	10.17533//udea.redin.20190519
[26]	Agile Practices Adoption in CMMI Organizations: A Systematic Literature Review	Conference	2016	10.1007/978-3-319-48523-2_6
[27]	Software process improvement: a systematic mapping study on the state of the art	Journal	2016	10.7717/peerj-cs.62
[28]	A systematic literature review of agile and maturity model research	Journal	2017	10.28945/3666
[29]	Scrum-DR: An Extension of the Scrum Framework Adherent to the Capability Maturity Model Using Design Rationale Techniques	Conference	2017	10.1109/CHILECON.2017.8229530
[30]	Investigating gaps on Agile Improvement Solutions and their successful adoption in industry projects - A systematic literature review	Conference	2018	10.18293/SEKE2018-185
[31]	A scrum-based software quality model to raise maturity of software in sme(s)	Journal	2020	19928645
[32]	Challenges in Combining Agile Development and CMMI: A Systematic Literature Review	Conference	2021	10.1145/3457784.3457803
[33]	A Framework for considering Quality of Data through Software Development	Conference	2022	10.1109/CONISOFT55708.2022.00012
[34]	A model for improving training of software developers in small companies	Journal	2015	10.1109/TLA.2015.7112002
[14]	Refining a Software System Deployment Process Model through Empirical Studies	Journal	2023	10.24215/16666038.23. e06
[35]	Acceptance of Software Process Improvement Models in Small and Medium Sized Enterprises: Empirical Findings of IT Sector in Turkey	Thesis	2023	https://hdl.handle.net/11511/102144
[36]	Supporting requirements engineering processes in small software enterprises	Thesis	2015	https://repositorio.uchile.cl/handle/2250/134724

excluded others such as SciELO or Latindex. Conference proceedings have been included with the IEEE and ACM publishers, which have significant presences in Computer Engineering and Computer Science publications. Proceedings books from other publishers not indexed in WoS or Scopus are excluded from our search. Therefore, it is very likely that the works left out of this study are a minority and have a lower impact on scientific dissemination (in terms of impact factor and number of citations) than the articles considered.

DISCUSSION OF THE RESULTS

By analyzing the selected primary studies, the evidence has helped us understand the existing challenges better when using traditional SPI models in agile software development processes. The following is a description of the most relevant findings for our research.

Regarding the first research question, which seeks to identify the challenges in incorporating an SPI model into agile development, we find the following relevant findings. No specific information on the challenges of incorporating SPI models into agile software development is provided in [34]. However, it is mentioned that teaching the agile philosophy from the academic areas is a challenge since it implies a change in the paradigm of the construction of applications, difficult to visualize and value by designers who have not lived the actual experience of development in companies. It is also highlighted that the CAAM training model seeks to solve the gap in the constant improvement of training in Mexican SMEs that use agile models.

Several challenges related to evaluating and improving agility in software development projects were discovered in [23] by applying the AgilityMod model to eleven organizations. These challenges include the need to establish effective communication channels when the customer is not physically present, ensure customer engagement, achieve an optimal level of granularity in user stories, maintain a steady pace of product backlog growth to ensure smooth development flow, conduct retrospective and review meetings effectively, manage technical debt, and identify dependencies between design elements for efficient change management. On the one hand, [25] mentions that the Scrum+ guide has been designed to reduce some of the challenges present in Global Software Development (GSD). It adapts the proposed solutions and harmonizes Scrum with other certifiable reference models, such as CMMI-DEV, ISO 9001, and ISO /IEC 15504, providing support for multi-model environments that a global software development project may face. On the other hand, in [24]. There are several challenges and limitations to integrating SPI with agile methods. These challenges include a lack of widely accepted and used agile maturity models, a lack of understanding of how to integrate agile methods into existing maturity models, and a paucity of specific tools and techniques to measure and evaluate the effectiveness of agile methods in improving software processes. Furthermore, existing maturity models need to be adapted to become more agile and flexible, which can be challenging for organizations that are used to following standardized processes.

In [14], incorporating Metrica v3 in developing DepProMod poses several challenges that require careful consideration and adaptation. DepProMod is based on a step-by-step approach, aligned with the CMMI-DEV standard capability levels, to address the software deployment process in SMEs. However, ensuring that the Metrics v3 measurement framework aligns appropriately with the agile principles and values that DepProMod intends to support is necessary. The selection and adaptation of the predefined metrics of Métrica v3 is another important challenge since they must be meaningful, actionable, and aligned with the goals and objectives of DepProMod. Furthermore, data collection and analysis must be balanced with the iterative and incremental approaches of agile development in DepProMod, avoiding an overhead that can affect the agility of the process. Integrating Metrica v3 in DepProMod also implies incorporating measurement activities into the existing practices of the utilization process. This integration must be done in a way that does not interrupt the flow and efficiency of the process in DepProMod. Agile teams must understand the usage context and use metrics to make informed decisions and drive continuous improvement.

In [35], the awareness of Turkish SMEs about the importance of SPI activities is highlighted, and the demand to adapt them to their own needs. A variation in attitudes towards SPI is identified according to the size of the companies, with micro-enterprises showing less motivation. Key factors such as management support, perception of productivity and quality, competencies, and resources affect the acceptance of productivity improvement activities. The shortage of skilled labor is highlighted as a critical constraint, highlighting the importance of staff training.

In [36], the classification of software practices proposed in the literature according to their perceived usefulness by the SSE is highlighted, which provides a clear guide for SPI professionals when selecting the most appropriate practices. The study also emphasizes the importance of the type of project as a critical factor in the choice of practices, allowing a more precise and practical approach to implementing SPI in Chilean SMEs.

In [30], two related investigations reached similar conclusions about joint CMMI implementation and agile practices. One of these studies pointed out that to reach higher levels of CMMI maturity, it is not enough to use only the agile approach; but it is necessary to complement it with other non-agile practices. This addition implies challenges when trying to reach higher levels of CMMI maturity using exclusively agile practices. On the other hand, in the other study, the authors carried out a systematic review and found challenges related to the adoption of CMMI together with agile practices, such as the necessary cultural change, the integration of both approaches and the lack of guidance in this regard. However, they also highlighted that these challenges can be overcome through proper planning and execution.

In [31], CMMI is mentioned as lagging in structured processes, process management, engineering practices, and quality assurance. In addition, aspects such as process quality assurance, building critical software, and maintaining an inventory of reusable artifacts are not adequately considered. There are also challenges related to integration, risk management, project budget control, and a lack of consideration for outsourced labor providers. However, the record also suggests that combining CMMI with agile methods can help address some of these challenges and improve software development practices.

In [32], the results were classified into different categories to address the joint implementation of CMMI and Agile. The authors managed to identify various challenges and classified them into seven different categories. In the first category, it was found that Agile alone cannot achieve the complete application of CMMI; it must be complemented with other methods. The second category highlights documentation, contracts, and planning issues in the agile context. Agile emphasizes customer collaboration over contract negotiation, but fine-tuning contractual frameworks can make agile development easier.

Additionally, agile planning differs from conventional approaches because it is more fluid and open to change. The third category addresses the balance between control and agility, as CMMI promotes process control and accountability, which can be inconsistent with agile principles. The fourth category highlights that adopting these approaches entails the need for additional resources, effort, and time on the part of the organization. The fifth category focuses on knowledge and organizational culture. The sixth category points to organizational or team resistance to introducing new work methods. Finally, the seventh category refers to organizational challenges, such as the incompatibility of the team or organization structure with agile approaches and the need to transition from legacy processes to agile and CMMI compliance processes.

Regarding the second research question, it is possible to point out that Scrum is the most referenced agile method [24]-[26], [29], [31], [32] when evaluating an SPI model or proposing a new one. In those articles where an agile method is not directly mentioned, it is possible to identify emphasizing the importance of the agile values and principles of the manifesto or directly pointing out agile practices that can be the subject of SPI. According to the literature review and the identifiers in the primary studies, CMMI and SPICE are the traditional SPI models that have been considered for agile software development initiatives. For the other two traditional models (MoProSoft/CompetiSoft/EvalProSoft and Metrica V3), it was not possible to identify direct applications with any agile method. Both models arise from government initiatives (Mexico and Spain, respectively) that sought to strengthen the internal software development industry to improve its competitive conditions at an international level.

CONCLUSIONS

This article presents a literature review to identify the challenges in incorporating an SPI model into agile software development. Additionally, an attempt was made to gather evidence on using traditional SPI models in agility. In general, we can infer that incorporating SPI models into agile software development can present challenges related to adapting processes and organizational culture to agile principles, and ensuring adequate and continuous training for team members. Based on the review, both SPICE and CMMI have been used and considered for SPI in agile development. Unfortunately, there is insufficient evidence to evaluate the other two traditional SPI models (MoProSoft/CompetiSoft/EvalProSoft and Metrica V3). In future work, we intend to analyze the new SPI proposals identified in this research to generate decision criteria that allow companies to choose the SPI model that best suits their needs.

REFERENCES

[1] A.A. Khan and J. Keung, “Systematic review of success factors and barriers for software process improvement in global software development,” *IET Software*, vol. 10, no. 5, pp. 125-135, 2016.

[2] H.C. Reyes, M. Rodriguez, P.L. Torres, and R. Villarroel, “Identification of technical and technological factors that influence software development projects,” *41st International Conference of the Chilean Computer Science Society (SCCC)*, Nov. 2022, doi:10.1109/sccc57464.2022.10000330.

[3] A.O. Balogun *et al.*, “Towards the sustainability of small and medium software enterprises through the implementation of software process improvement: Empirical investigation,” *Journal of Software: Evolution and Process*, vol. 34, no. 8, 2022, doi: 10.1002/smr.2466.

[4] Secretaría de Economía, *Modelo de Procesos para la Industria de Software: MoProSoft*, v1.3, México, 2005.

[5] S. Álvarez Arriaga, “Moprosoft un modelo básico para el éxito en el desarrollo y mantenimiento del software”, Tesis de Grado, Facultad de Ingeniería. Universidad Nacional Autónoma de México, 2011. [En línea]. Disponible en: http://www.ptolomeo.unam.mx:8080/xmlui/handle/132.248.52.100/180

[6] H. Oktaba, F. García, M. Piattini, F. Ruiz, F. J. Pino, and C. Alquicira, “Software Process Improvement: The Competisoft Project,” *Computer*, vol. 40, no. 10, pp. 21-28, oct. 2007, doi: 10.1109/MC.2007.361.

[7] Information Technology, Software Process Assessment, ISO/IEC TR 15504, International Organization for Standardization, 1998.

[8] Portal de Administración Electrónica, “Métrica v.3,” administracionelectronica.gob.es. España, 2001. https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Metrica_v3.html

[9] CMMI-SE/SW/IPPD, v.1.1 (2001), Carnegie Mellon University, Software Engineering Institute’s Digital Library, doi: 10.1184/R1/6572468.v1.

[10] F.J. Pino, F. García, M. Piattini, and H. Oktaba, “A research framework for building SPI proposals in small organizations: the COMPETISOFT experience,” *Software Quality Journal*, vol. 24, no. 3, pp. 489-518, 2016, doi: 10.1007/s11219-015-9278-2.

[11] J. Ferguson, “The CMMI Project,” *INCOSE International Symposium*, vol. 9, Jun. 01, 1999, doi: 10.1002/j.2334-5837.1999.tb00317.x.

[12] K.E. Emam and H.W. Jung, “An empirical evaluation of the ISO/IEC 15504 assessment model,” *Journal of System and Software*, vol. 59, no. 1, pp. 23-41, oct. 2001, doi: 10.1016/S0164-1212(01)00046-2.

[13] C. Santana, F. Queiroz, A. Vasconcelos, and C. Gusmão, “Software Process Improvement in Agile Software Development a Systematic Literature Review,” in *41st Euromicro Conference on Software Engineering and Advanced Applications*, Aug. 26, 2015, pp. 325-332, doi: 10.1109/SEAA.2015.82.

[14] M.D. Panizzi, M. Genero, and R. Bertone, “Refining a software system deployment process model through empirical studies,” *Journal of Computer Science and Technology*, vol. 23, no 1, p. e06, 2023, doi: 10.24215/16666038.23.e06.

[15] R. Constantinescu and I. Mihnea, “Capability Maturity Model Integration,” *Journal of Applied Quantitative Methods*, vol. 2,

no. 1, 2007. [Online]. Available: https://www.jaqm.ro/issues/volume-2,issue-1/pdfs/constantinescu_iacob.pdf

[16] M. Sivashankar, A. Kalpana, and A. Jeyakumar, “A framework approach using CMMI for SPI to Indian SME’S,” *International Conference on Innovative Computing Technologies (ICICT)*, Mar. 29, 2010, pp. 1-5, doi: 10.1109/ICINNOVCT.2010.5440086.

[17] T. Rout, K. Emam, M. Fusani, D.R. Goldenson, and H.-W. Jung, “SPICE in retrospect: Developing a standard for process assessment,” *Journal of Systems and Software*, vol. 80, Sep. 01, 2007, pp. 1483-1493, doi: 10.1016/j.jss.2007.01.045.

[18] X. Larrucea, R.V. O’Connor, R.C. Palacios, and C.Y. Laporte, “Software Process Improvement in Very Small Organizations,” *IEEE Software*, vol. 33, pp. 85-89, 2016, doi: 10.1109/MS.2016.42.

[19] M. Suominen and T. Mäkinen, “On the applicability of capability models for small software organizations: does the use of standard processes lead to a better achievement of business goals?,” *Software Quality Journal*, vol. 22, pp. 579-591, 2013, doi: 10.1007/s11219-013-9201-7.

[20] H. Snyder, “Literature review as a research methodology: An overview and guidelines,” *Journal of Business Research*, vol. 104, pp. 333-339, 2019, doi: 10.1016/J.JBUSRES.2019.07.039.

[21] H. Cornide-Reyes, J. Morales, F. Silva-Aravena, A. Ocqueteau, N. Melendez, and R. Villarroel, “Implementing Digital Transformation Processes in Industry 4.0*,” Social Computing and Social Media. HCII 2023. Lecture Notes in Computer Science,* vol. 14025, A. Coman and S. Vasilache, Eds. 2023, pp. 323-335, doi: 10.1007/978-3-031-35915-6_24.

[22] J. Morales, H. Cornide-Reyes, P. Rossel, P. Sáez, and F. Silva-Aravena, “Virtual Reality, Augmented Reality and Metaverse: Customer Experience Approach and User Experience Evaluation Methods. Literature Review,” *Social Computing and Social Media. HCII 2023. Lecture Notes in Computer Science. Lecture Notes in Computer Science*, A. Coman and S. Vasilache, Eds. 2023, pp. 554-566, doi: 10.1007/978-3-031-35915-6_40.

[23] Ö. Özcan-Top and O. Demirörs, “Application of a software agility assessment model - AgilityMod in the field,” *Computer Standards & Interfaces*, vol. 62, pp. 1-16, 2019, doi: 10.1016/j.csi.2018.07.002.

[24] S. Küpper, D. Pfahl, K. Jürisoo, P. Diebold, J. Münch, and M. Kuhrmann, “How has SPI changed in times of agile development? Results from a multi-method study,” *Journal of Software: Evolution and Process*, vol. 31, no. 11, 2019, doi: 10.1002/smr.2182.

[25] C.J. Pardo Calvache, P.R. Chilito Gomez, D.E. Viveros Meneses, and F.J. Pino Correa, “Scrum+: A scaled Scrum for the agile global software development project management with multiple models”, *Revista Facultad de Ingenieria Universidad de Antioquia*, no. 93, pp.105-116, 2019, doi: 10.17533//udea.redin.20190519.

[26] M. Palomino, A. Dávila, K. Melendez, and M. Pessoa, “Agile practices adoption in CMMI organizations: a systematic literature review,” *Trends and Applications in Software Engineering. CIMPS 2016. Advances in Intelligent Systems and Computing*, vol. 537, J. Mejia, M. Muñoz, Á. Rocha, T. San Feliu,and A. Peña, Eds. pp. 57-67, 2017, doi: 10.1007/978-3-319-48523-2_6.

[27] M. Kuhrmann, P. Diebold, and J. Münch, “Software process improvement: a systematic mapping study on the state of the art,” *PeerJ Computer Science*, vol. 2, p. e62, 2016, doi: 10.7717/peerj-cs.62.

[28] V. Henriques and M. Tanner, “A Systematic Literature Review of Agile Maturity Model Research,” *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 12, pp. 53-73, 2017, doi: 10.28945/3666.

[29] S. Kawamoto and J. R. de Almeida, “Scrum-DR: An extension of the scrum framework adherent to the capability maturity model using design rationale techniques,” *2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON),* Pucon, Chile, 2017, pp. 1-7, doi: 10.1109/CHILECON.2017.8229530.

[30] Arthur Silva Freire *et al.*, “Investigating gaps on Agile Improvement Solutions and their successful adoption in industry projects - A systematic literature review,” in *Proceedings*

of the 30th International Conference on Software Engineering and Knowledge Engineering (SEKE), 2018, doi: 10.18293/seke2018-185.

[31] S. Amer, N. Badr, and A. Hamad, "A scrum-based software quality model to raise maturity of software in sme(s)," *Journal of Theoretical and Applied Information Technology*, vol. 98, no. 7, 2020.

[32] A. Ferdinansyah and B. Purwandari, "Challenges in Combining Agile Development and CMMI: A Systematic Literature Review," *10th International Conference on Software and Computer Applications*, Feb. 2021, doi: 10.1145/3457784.3457803.

[33] C. Guerra-García, H.G. Pérez-González, F. Martínez-Pérez, S.E. Nava-Muñoz, and R. Juárez-Ramírez, "A Framework for considering Quality of data through software development," *2022 10th International Conference in Software Engineering Research and Innovation (CONISOFT)*, Ciudad Modelo, San José Chiapa, Mexico, 2022, pp. 1-10, doi: 10.1109/CONISOFT55708.2022.00012.

[34] C. Enriquez and P. Gomez Gil, "A model for improving training of software developers in small companies," *IEEE Latin America Transactions*, vol. 13, no. 5, pp. 1453-1461, 2015, doi: 10.1109/tla.2015.7112002.

[35] S. Durmuş, "Acceptance of Software Process Improvement Models in Small and Medium Sized Enterprises: Empirical Findings of IT Sector in Turkey," Ph.D. thesis, Department of Information Systems. Middle East Technical University, Ankara, Turkey, 2023. [Online]. Available: https://open.metu.edu.tr/handle/11511/102144

[36] A. Quispe Sanca, "Supporting requirements engineering processes in small software enterprises", Tesis Doctoral. Departamento de Ciencias de la Computación. U*niversidad de Chile*, 2015. [En línea]. Disponible en: https://repositorio.uchile.cl/handle/2250/134724