Framework proposal to improve models and standards.

Victor Terrón-Macias, Jezreel Mejía*, Mirna Muñoz

Centro de Investigación en Matemáticas, Parque Quantum, Ciudad del Conocimiento, Av. Lassec, Andador Galileo Galilei, Manzana, 3, Lote, 7, 98160, Zacatecas, ZAC, México
{victor.terron, jmejia, mirna.munoz}@cimat.mx

Abstract. This article delves into combining models and standards in software development. Through a comprehensive exploration, the article highlights the suggested elements to consider when adopting different models or standards and the importance of mathematical formalization to ensure compatibility and minimize the risk of failure in multi-model environments. Mathematical formalization emphasizes the need for a systematic and rigorous approach to verify and validate the coherence and interaction between selected models and standards. By implementing models and standards in a compatible manner, organizations can increase the likelihood of success, drive continuous improvement, and adapt to changing needs. Furthermore, the article underscores the significance of research and analysis in addressing the need for mathematical formalization. By bridging this gap, the integration of models and standards can be enhanced, leading to more efficient and effective software development practices. Overall, this is a valuable resource for practitioners, researchers, and organizations seeking to optimize their software development processes. Organizations can establish a solid foundation for successful integration and evolution in the dynamic software landscape by considering the suggested elements and adopting a systematic approach.

Keywords: Software improvement, model integration, multi-model environments, mathematical formalization.

1 Introduction.

A growing number of organizations have been attempting to improve their software processes using standards or models, such as ISO/IEC 29110, CMMI, SIX SIGMA, ISO/IEC 12207, among others. This is because the quality of a product depends on the quality of the processes, as indicated by N. Ehsan et al. [1].

In the software development industry, it is not uncommon for companies to hold multiple certifications and implement different models and standards simultaneously. Implementing these methodologies is often approached empirically, needing a systematic and well-defined approach. Consequently, the adoption of these models becomes a challenging task. Despite their benefits, limitations, difficulties, and disadvantages must be taken into consideration, particularly in multi-model environments where the risk of failure is heightened [2].

The difficulties encountered in adopting of any model, standard, or methodology are manifold. One of the primary challenges lies in selecting the most suitable model for a particular organization's needs. Making an informed choice requires a deep understanding of the organization's requirements, resources, and expertise. Unfortunately, lack of resources or expertise [3]. Additionally, the absence of formalized elements or guidelines for suggesting the implementation approach further compounds the adoption challenges. Consequently, organizations may need help to navigate the complexities of these models and standards.

D. Stelzer and W. Mellis propose an incremental adoption approach [3] to avoid the complexities of adopting models and/or standards. This approach advocates for a step-by-step implementation of the chosen model or standard, allowing organizations to gradually adapt their processes and build the necessary expertise and resources. By adopting this incremental approach, companies can minimize the risks associated with the adoption process and increase the likelihood of successful implementation. Moreover, this approach provides an opportunity to continuously evaluate the effectiveness of the adopted model and make necessary adjustments along the way.

2 Related works.

To identify related works, a systematic review of the literature was carried out, for which the following research questions are posed.

What are the most recent scientific articles discussing the challenges and benefits of combining different models or standards in various fields?

This question aims to uncover the contemporary discourse surrounding the integration of diverse models or standards and the insights gained from these endeavors. By analyzing recent literature, we seek to identify the primary motivations behind such integration and the potential advantages it brings to different domains.

How have researchers addressed the compatibility issues between different models or standards when attempting to combine them? Are there any emerging best practices?

This question delves into the practical aspect of combining models or standards. By investigating how researchers have navigated the complexities of compatibility issues, we aim to unearth strategies, methodologies, and emerging best practices that contribute to successful integration across fields.

What are the key factors to consider when selecting models or standards for combination to ensure accurate and reliable results?

The selection of models or standards for combination is a critical step that directly influences the validity and reliability of outcomes. By exploring the literature, we intend to identify the factors that researchers should weigh when making these choices. This question seeks to provide insights into the decision-making process and criteria that lead to robust and trustworthy results.

The inclusion criteria for the articles are:

- ***Publication Date:*** Articles published within the last five years (from 05/2017 to 07/2023) will be considered to ensure relevance to current discussions and advancements.
- ***Topic Relevance:*** Articles should explicitly discuss or address the challenges and benefits of combining different models or standards in various fields. Relevance to the central theme of integrated approaches and their implications should be evident in the title, abstract, or keywords.
- ***Cross-Disciplinary Focus:*** Articles that span multiple disciplines or domains are preferred, showcasing the application of combined models or standards across various sectors.
- ***Original Research:*** Primary research articles that present empirical studies, case studies, experiments, or real-world applications are prioritized to provide insights based on concrete data and outcomes.
- ***Compatibility Solutions:*** Articles should explore or propose methodologies, techniques, or strategies to address compatibility issues when combining different models or standards. Identification of emerging best practices or innovative approaches is a key criterion.
- ***Selection Criteria and Factors:*** Articles should discuss or provide insights into the key factors researchers should consider when selecting models or standards for combination to ensure accurate and reliable results. This may include criteria for model suitability, domain relevance, data integration, and validation methods.
- ***Methodological Detail:*** Articles should provide a clear explanation of the methodologies, frameworks, or processes employed to combine models or standards, demonstrating how the integration was achieved and its impact.
- ***Outcome Evaluation:*** Articles should assess or discuss the outcomes of combining models or standards, including benefits, limitations, and potential challenges encountered during the integration process.
- ***Peer-Reviewed:*** Articles should be published in reputable, peer-reviewed journals, conferences, or other scholarly outlets to ensure the credibility and reliability of the research.
- ***Language:*** Articles published in English will be included to facilitate comprehensive analysis and understanding.

The selected studies have been chosen because their objectives are closely aligned with the goals established in this work. They are briefly described below:

(S1) M. Baldassarre et al. [4] compared the ISO/IEC 12207 and ISO/IEC 15504-7 standards with the CMMI-DEV model at a low level to identify the degree of coverage of the standards with the maturity levels of the standard and presents the potential of one over the other to support the improvement process. In this article, the following steps are taken, studying, and analyzing the models to identify their requirements, identifying the structure of the models and standards, comparing the models and standards, and establishing relationships between activities or processes.

(S2) J. Crisostomo et al. [5] analyze the literature of the ISO/IEC 12207 standard and the CMMI-DEV model, aiming to introduce or improve their practices with each other considering these models. This article conducts a systematic literature review

with the objective of verifying coverage between standards and models. The five steps it follows are:

- Analyzing the models to be compared.
- Designing comparison criteria.
- Executing the comparison.
- Presenting the results.
- Analyzing the results.

(S3) S. Hyung and H. Gyun [6] analyze the relationship between the ISO/IEC 15504 standard, CMMI (Capability Maturity Model Integration), and the k-model. The study compares the practices of the k-model with CMMI and ISO/IEC 15504, integrating them to improve the development process. As a first step, this article analyzes the ISO/IEC 15504 standard, the CMMI model, and the K-model. It then identifies their structure and architecture, followed by identifying the levels of coverage and characteristics. The article also highlights similarities in the adoption of each standard and mode.

(S4) N. Ehsan et al. [1] identify the relationship between SPICE (Software Process Improvement and Capability Determination) and CMMI (Capability Maturity Model Integration), which are models created by ISO/IEC and SEI (Software Engineering Institute), respectively. This relationship aims to achieve process improvement and adopt the model and the standard. This article follows a series of steps to integrate the CMMI model with the ISO/IEC 15504 standard. As a first step, it analyzes the model and the standard, then identifies the coverage of one concerning the other. Next, it identifies the differences and finally compares them. The article also mentions that adopting process areas depends on the maturity levels of the company.

(S5) M. Niazi et al. [2] conducted a study that incorporates practices from the CMMI (Capability Maturity Model Integration) and ISO's SPICE standard (Software Process Improvement and Capability Determination) to achieve software process improvement through interviews. This article first studies and analyzes the structure of the model and standard. It then defines critical success factors and barriers between available literature and empirical study. This study considers practitioners' experiences, opinions, and views to identify factors that have both positive and negative impacts on software process improvement.

(S6) A. Ferreiro et al. [7] apply the ISO/IEC 9001:2000 standard, MPS.BR (Melhoria de Processo do Software Brasileiro) and CMMI (Capability Maturity Model Integration) to achieve software process maturity and improve the development process of a project. This article proposes the following steps for adoption: as a first step, defining objectives and needs; then, planning assessments; defining and managing an action plan; developing the improvement plan; training team members; executing the processes; identifying improvement opportunities; and analyzing the improvement opportunities to obtain the respective certification ultimately.

(S7) N. Khamalakshi and H. Naganna [8] explained the relationships between CMMI and Six Sigma and the implementation within companies. The steps that are established are:

- Selecting a business objective.
- Collecting information.
- Analyzing the information to identify cause-effect relationships.

- Prioritizing issues.
- Identifying possible causes.
- Developing causal analysis.
- Prioritizing causes.
- Identifying possible solutions.
- Developing an action plan.
- Implementing improvements.

(S8) J. Mejía et al. [9] developed proposals to integrate security practices from the ISO/IEC 27001 standard into the ISO/IEC 29110 standard for mobile application development. This article proposes a series of steps to integrate security practices from different standards:

- It analyzes and studies the standards at the task level to identify their structures.
- It establishes the explicit elements at the framework level where traceability will be developed. It also identifies and analyzes the frameworks around interest. The article examines the terminology and concepts that match and/or are a priority for traceability.
- It generates a proposal for security practices towards the ISO/IEC 29110 standard.

(S9) P. Maciel et al. [10] developed a tool called "29110-TS" that presents improvement proposals regarding the security of ISO/IEC 27001 and ISO/IEC 27034 standards to the ISO/IEC 29110 standard. This article is based in the proposal of J. Mejía, et al. [9].

(S10) K. Suteeca and S. Ramingwong [11] developed a framework to apply ISO/IEC 29110 on SCRUM, focusing on the explanation of the relationships between Agile practices and an implementation of software processes defined by the ISO/IEC 29110 standard. This article first analyzes the ISO/IEC 29110 standard and the SCRUM methodology. It then creates graphical representations of the system processes using IDEF0 diagrams. The paper aims to identify areas and activities of convergence where they can be implemented together.

(S11) A. Salazar and C. Ovalle [12] designed a framework that integrates RUP (Rational Unified Process), SCRUM, and ISO/IEC 9126 to improve software quality in the construction phase by implementing methodologies and the standard. This article outlines a series of steps. The first step is to adopt the four phases of RUP (Rational Unified Process), incorporating the activities outlined by the ISO/IEC 9126 standard. Additionally, it integrates activities and requirements from the SCRUM methodology.

After the analysis of the 11 articles of the analysis of related works, sections 3 and 4 address the reasons and steps respectively for the combination of standards or models.

3 Identified reasons in related works for properly Combining Models and Standards.

As a result of the analysis of related works, reasons why organizations find it necessary to combine models and standards were identified:

1. ***Ensure coherence and alignment with objectives and project requisites:*** Software development standards and models often have different approaches and principles. Therefore, it is necessary to guarantee this key criterion. This ensures that standards and models complement each other and work harmoniously instead of conflicting, as suggested by S. Bordeau et al. [19]. On the other hand, this ensures that the models and standards to be combined respect and align with the objectives of the organization and the project.
2. ***Contextual adaptation to the market:*** A. Brogui et al. [20] suggests that each project and organization have its unique characteristics and requirements. By considering key criteria, you can adapt the implementation of software development standards and market's needs. This allows you to leverage the benefits of each standard or model while tailoring them to fit the needs and constraints of your project.
3. ***Continuous improvement:*** Key criteria help evaluate and measure the effectiveness of the process and the implementation of software development standards and models. This enables the identification of areas for improvement and adjustments to optimize the process. Continuous improvement is crucial to ensure that the implementation remains effective over time and can adapt to project environment changes, as suggested by M. Muñoz et al. [21].
4. ***Regulatory compliance with market needs:*** By considering key criteria, you can ensure that the implementation of software development standards and models complies with applicable regulatory and normative requirements in your industry, as indicated by E. Kempe and Aaron Massey [22]. This is particularly important in highly regulated sectors such as healthcare or finance, where non-compliance can have legal or security consequences.
5. ***Improve team capabilities and competencies:*** J. Carvalho, et al. [23] suggest that the effective implementation of software development standards and models also relies on the skills and competencies of the team. By considering key criteria, you can evaluate whether the team has the necessary capabilities to implement the selected standards and models. This allows for identifying knowledge gaps and providing additional training or appropriate role assignments and responsibilities to improve capabilities and competencies.

By properly combining models and standards based on these reasons, organizations can achieve a more effective and efficient software development process that aligns with their specific goals, context, and compliance requirements.

The following section describes how authors combine different models and standards.

4 Identified steps in related works for combining standards and models.

As a result of the related works section, the following steps are identified because when combining software development standards and models, it is crucial to consider critical criteria to ensure effective implementation, as mentioned by S. Kaur et al. [13]:

1. ***Establish the reference standard or model:*** This reference standard or model mark on which others (models or standards) will be implemented as suggest J. Mejía, et al. and P. Maciel.
2. ***Select standards or models that are coherent and aligned:*** Select the set of standards or models.
3. ***Project objectives and scope:*** The École de technologie supérieure [14] indicates that the first step in every project is to define the project objectives and scope to determine which software development standards and models are relevant to meet those objectives. For example, if the project has critical security requirements, it may be necessary to consider standards like ISO/IEC 27001 or development models like DevSecOps.
4. ***Compatibility and consistency:*** Evaluate how different software development standards and models can be combined coherently and without conflicts. The analysis should identify areas of overlap and consistency between the selected standards and models, ensuring they are compatible. For example, you want to combine agile development with the ISO/IEC 12207 standard. In that case, you need to identify how agile principles can be integrated coherently with the processes defined in the standard.
5. ***Confirm compatibility with mathematical logic and tautologies:*** V. Terrón and J. Mejía [15] advise the use of tautologies and mathematical logic to ensure that the standard and model are consistent and compatible, ensuring that one does not obstruct the other, the way that it can be applied is:
    a. *Formalization of models and standards:* Use mathematical logic to formalize and accurately represent software models and standards. This involves defining key elements, relationships, and rules using formal languages such as predicate logic or first-order logic.
    b. *Formal verification:* Apply techniques based on mathematical logic to analyze the consistency and compatibility between models and standards. This may include verifying specific properties such as logical consistency, invariance, and absence of conflicts among different system parts.
    c. *Deductive reasoning:* Use deductive reasoning based on logical principles to infer conclusions and demonstrate compatibility between software models and standards. This involves applying rules of inference and logical axioms to reach valid conclusions from established premises.
    d. *Formal modeling:* Employ techniques based on mathematical logic, such as predicate calculus or set theory, to represent the elements of

software models and standards accurately and unambiguously. This facilitates the identification of potential inconsistencies or conflicts among different elements.

e. *Proof theory:* Utilize proof theory to formally demonstrate the compatibility and consistency between software models and standards. This involves constructing solid logical arguments demonstrating the absence of contradictions or incompatibilities in combining the selected approaches.

It is important to note that using mathematical logic to ensure compatibility and consistency in the mixing of software models and standards requires expertise in formal logic and mathematics.

6. ***Check legal and regulatory requirements:*** The combination of standards and development models should comply with applicable legal and regulatory requirements to promote the transparent and efficient allocation of resources, as indicated by OECD [16]. Some projects may be subject to specific regulations, such as Data Protection Laws or industry regulations, and it is important to consider how the selected standards and models align with such regulations.
7. ***Evaluate team culture and capabilities:*** Evaluate the culture and capabilities of the development team to determine which approaches are most appropriate and feasible, since as the PMI mentions, culture influences the performance of work teams [17]. When combining standards and development models, it is important to consider whether the team is familiar with the methods and practices associated with those standards and models or if additional training is required.
8. ***Recognize flexibility and adaptability:*** Recognize that software development standards and models are not rigid and can be adapted to project needs, as indicated by ISO/IEC in every implementation guide [18]. Consider how you can customize and adjust the standards and models better to fit your project's specific characteristics and requirements.
9. ***Research industry best practices:*** Research and consider industry best practices related to the software development standards and models you are using. Tap into the experience and knowledge the software development community accumulated to gain insights and recommendations on effectively combining the standards and models.

The combination of software development standards and models may vary depending on each project's context and specific requirements. Therefore, adapting and adjusting these elements according to the unique needs and characteristics of the organization and the project is suggested.

Fig. 1 shows the suggested elements when adopting different models or standards.

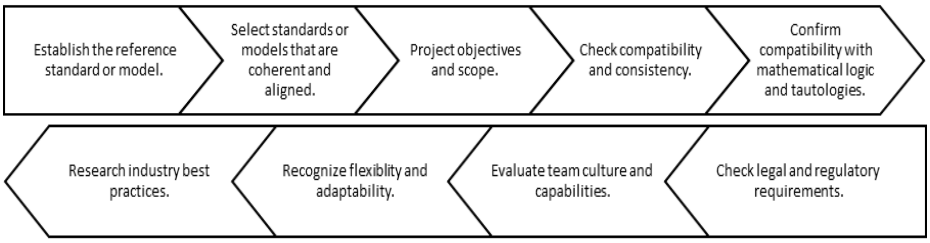

Fig. 1 – ***Overall process of a software development project cycle identified in related works.***

The suggested steps by the authors (ST) that must be covered in the implementation of models and that are addressed in related works are:

ST1. Establish the reference standard or model on which others (models or standards) will be combined.

ST2. Select those that are coherent and aligned.

ST3. Study and analyze the models or standards to identify its explicit structure.

ST4. Establish explicit elements level of the Model or Standard where the traceability will be developed.

ST5. Identify and analyze related Models Standards or Methodologies of the area in interest.

ST6. Analyze the terminology and concepts that coincide and/or priority for the traceability.

ST7. Generate a proposal of practices between Models, Standards, or Methodologies.

ST1 and ST2 are not mentioned explicitly by any study but are identified in all authors' analyses as a previous step to adapt different models and standards to a base. Table 1 shows the steps followed by the studies for adopting the models, standards, or methodologies (MSMs).

Table 1. Steps followed by the studies for the adoption of the MSMs.

Step	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
ST1	X	X	X	X	X	X	X	X	X	X	X
ST2	X	X	X	X	X	X	X	X	X	X	X
ST3	X	X	X	X	X	X	X	X	X	X	X
ST4	X			X	X	X	X	X	X	X	
ST5	X	X						X	X	X	X
ST6	X		X			X	X	X	X	X	X
ST7			X		X	X	X	X	X		X

As can be seen in Table 1, in the context of implementing standards and models or methodologies, and after analyzing related works, the most applied steps are ST1, ST2, and ST3; on the other hand, an evident issue is identified: ***the lack of a mathematical form and a rigorous systematic process to ensure compatibility among them***. This deficiency can increase the possibility of failure in the implementation, as mentioned earlier in multi-model environments. Given this situation, it is crucial to address the need to establish elements to effectively combine standards and models, ensuring their compatibility and optimizing results. Due to this identified issue, the following section addresses a proposal of steps for combining software development models or standards.

5 Proposal of steps for combining software development models or standards.

After analyzing the available literature and considering both the elements and the reasons for adecuately combining models and standards, and recognizing the lack of mathematical formalization, a series of steps is proposed, which are listed below:

1. Establish the reference standard or model on which others (models or standards) will be implemented.
2. Select those standards or models that are coherent and aligned.
3. Study and analyze the standard or model to identify its nature and explicit structure.
    a. Name.
    b. Purpose.
    c. Objectives.
    d. Activity.
    e. Activity description.
    f. Task.
    g. Input Product.
    h. Internal Products.
    i. Output Products.
    j. Roles involved.
    k. Incorporation into the project repository.
4. Use logical mathematics to verify compatibility.
5. Establish the explicit element level of the base framework where traceability will be performed.
6. Identify and analyze related frameworks of the area of interest.
7. Analyze the terminology and concepts that coincide and or prioritize traceability.
8. Generate a proposal of practices between models or standards.

Fig. 2 resumes the proposal of steps to mix software development models or standards.

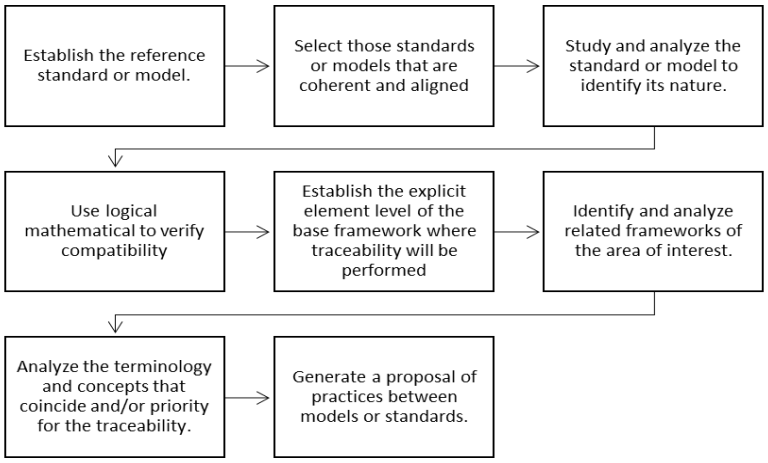

Fig. 2 – ***Proposal of steps for combining software development models or standards.***

These steps provide a solid foundation for ensuring the compatibility of models and standards by mathematically formalizing their adoption and implementation. By establishing a systematic and rigorous approach, we can reduce the uncertainty and risks associated with combining these elements. By mathematically formalizing the processes, we can verify and validate the coherence and interaction between the selected models and standards, ensuring they work together efficiently and effectively. This increases the chances of success in implementation and lays the groundwork for continuous evolution and adaptation to future needs and changes in the environment.

6 Conclusion

We have explored the suggested elements to consider when adopting different models or standards, highlighting the need for mathematical formalization to ensure compatibility and minimize the risk of failure in multi-model environments.

By understanding these elements and following the proposed steps, organizations can establish a solid foundation for integrating models and standards effectively. The systematic and rigorous approach, supported by mathematical formalization, verifies and validates the coherence and interaction between the selected models and standards.

Implementing models and standards in a compatible manner increases the likelihood of success and enables continuous improvement and adaptation to changing needs. It provides organizations a framework to navigate complex software development processes, reducing uncertainty and mitigating risks.

Furthermore, this article emphasizes the importance of research and analysis in identifying gaps and addressing the need for mathematical formalization in the field. By bridging this gap, we can enhance the implementation of models and standards, ultimately leading to more efficient and effective software development practices.

Overall, the knowledge shared in this chat serves as a valuable resource for practitioners, researchers, and organizations seeking to optimize their software development processes through the appropriate combination of models and standards. By considering the suggested elements and adopting a systematic approach, organizations can pave the way for successful integration and evolution in an ever-changing software landscape.

References

[1] Ehsan, N. and Perwaiz, A. and Arif, J. and Mirza, E. and Ishaque, A., "2010 IEEE International Conference on Management of Innovation & Technology," *CMMI / SPICE based process improvement,* vol. 0, no. 0, pp. 859-862, 2010.

[2] Niazi Nahmood, David Wilson, Didar Zowghi, "A maturity model for the implementation of software process improvement: an empirical study," *Systems and software,* vol. 74, no. 2, pp. 155-172, 17 Junio 2007.

[3] Dirk Stelzer, Werner Mellis, "Success factors of organizational change in software process improvement," *Software Process Improvement and Practice,* vol. 4, no. 4, pp. 227-250, 14 Mayo 2000.

[4] Baldassarre, Maria Teresa and Piattini, Mario and Pino, Francisco J. and Visaggio, Giuseppe, "Comparing ISO/IEC 12207 and CMMI-DEV: Towards a mapping of ISO/IEC 15504-7," *2009 ICSE Workshop on Software Quality,* vol. 0, no. 0, pp. 59-64, 12 Junio 2009.

[5] Javier Crisóstomo, Luis Flores, Abraham Melendez, Dávila Karin, "Convergence analysis of ISO/IEC 12207 and CMMI-DEV: A systematic literature review," *2016 XLII Latin American Computing Conference (CLEI),* vol. 0, no. 0, pp. 1-8, 26 enero 2016.

[6] Hwang, Sun-myung and Yeom, Hee-gyun, "Analysis of Relationship among ISO/IEC 15504, CMMI and K-model," *2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing,* vol. 0, no. 0, pp. 306-309, 13 octubre 2009.

[7] Analia Irigoyen Ferreiro Ferreira, Gleison Santos, Roberta Cerqueira, Mariano Montoni, Ahilton Barreto, Andrea O. Soares Barreto, Ana Regina Rocha, "Applying ISO 9001:2000, MPS.BR and CMMI to Achieve Software Process Maturity: BL Informatica's Pathway," *29th International Conference on Software Engineering (ICSE'07),* vol. 0, no. 0, pp. 642-651, 04 junio 2007.

[8] H. Kamalakshi and N. Naganna , "CMMI and Six Sigma-Relationship & Integration," *2009 International Conference on Signal Acquisition and Processing,* vol. 0, no. 0, pp. 174-176, 14 julio 2009.

[9] Jezreel Mejía, Mirna Muñoz, Perla Maciel-Gallegos, Yadira Quiñones, "Proposal to Integrate Security Practices into the ISO/IEC 29110 Standard to Develop Mobile Apps.," in *New Perspectives in Software Engineering. CIMPS 2021. Advances in Intelligent Systems and Computing*, vol. 1416, Monterrey, Nuevo León: Springer,

2021, pp. 29-40.

[10] Perla Maciel-Gallegos, Jezreel Mejía, Yadira Quiñones, "29110-ST: Integrated Security ractices. Case Study," in *New Perspectives in Software Engineering. CIMPS 2022.*, vol. 576, Acapulco, Guerrero: SpringerLink, 2022, pp. 113-125.

[11] Kittitouch Suteeca, Ramingwong Sakgasit, "A framework to apply ISO/IEC29110 on SCRUM," *2016 International Computer Science and Engineering Conference (ICSEC),* vol. 0, pp. 1-5, 14 diciembre 2016.

[12] André Salazar Jo, Christian Ovalle , "Framework integrating RUP, Scrum and ISO 9126 to improve Quality during Software Construction in SMEs," *2022 IEEE Engineering International Research Conference (EIRCON),* vol. 0, no. 0, pp. 1-4, 04 noviembre 2022.

[13] Simarpreet Kaur, Sumeet Kaur Sehra, Sukhijt Singh Sehra, "A Framework for Software Quality Model Selection using TOPSIS," *IEEE International Conference On Recent Trends In Electronics Information Communication Technology,* pp. 1-4, 20-21 Mayo 2016.

[14] École de technologie supérieure, "QSPM Chapter 7: Defining the goaland scope of the software project," in *Quality Software Project Management*, Prentice Hall, 2002, pp. 1-15.

[15] Victor Terrón, Jezreel Mejía, "Propuesta de mejoras para la implementación del estándar ISO/IEC 29110," *Revista Ibérica de sistemas y tecnologías de la información,* no. 49, pp. 1-17, 31 Mayo 2023.

[16] OECD, "OECD/Legal/0413," *OECD Legal Instruments,* p. 1, 07 junio 2023.

[17] V. Suda Lawrence, "The meaning and importance of culture for project success," *PMI® Global Congress,* p. 1, 2007.

[18] ISO/IEC, Asessment guide ISO/IEC 29110-3, IEEE, 2018.

[19] Simon Bourdeau, Pierre Hadaya, Jean-Etienne Lussier, "Assessing the Strategic Alignment of Information Systems Projects: A Design Science Approach," *Projectics / Proyéctica / Projectique,* vol. 20, no. 2, pp. 115-154, 2018.

[20] Antonio Brogui, Javier Cámara, Carlso Canal, Javier Cubo, Ernesto Pimentel, "Dynamic Contextual Adaptation," *Theoretical Computer Science,* vol. 175, no. 2, pp. 81-95, 2007.

[21] Mirna Muñoz, Jezreel Mejía, Alor Ginero, Jose A. Calvo-Manzano, Tom's San Feliu, "Advantages of Using a Multi-model Environment in Software Process Improvement," in *2011 IEEE Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, México, IEEE Xplore, 2011, pp. 397-402.

[22] Evelyn Kempe, Aaron Massey, *2021 IEEE 29th International Requirements Engineering Conference (RE),* pp. 46-57, 20-24 septiembre 2021.

[23] Jorge Oliveira Carvalho, Joao Alvaro and Sousa, Rui Dinis and Sá, "Information Systems Development Course: Integrating Business, IT and IS Competencies," in *2010 IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*, Dublin, Ireland, Dublin: IEEE, 2010, pp. 1-20.

[24] Ruiz, Julio Cesar and Osorio, Zurisadai Benjamin and Mejia, Jezreel and Munoz, Mirna and Chávez, Ana Maria and Olivares, Beatriz Alejandra, "Definition of a Hybrid Measurement Process for the Models ISO/IEC 15504-ISO/IEC 12207:2008 and CMMI Dev 1.3 in SMEs," *2011 IEEE Electronics, Robotics and Automotive Mechanics Conference,* vol. 0, no. 0, pp. 421-426, 9 noviembre 2011.