Identifying deficiencies in the traceability record of a software development company under the ISO/IEC 29110 standard: A detailed analysis of a case study.

Victor Terrón-Macias, Jezreel Mejía*

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}@cimat.mx

Abstract. This article presents a comprehensive analysis of the traceability record DNA according to the ISO/IEC 29110 standard, focusing on traceability and its characteristics. Through three analyses conducted in a company located in Zacatecas during its recertification process in 2022, the compliance of the traceability record and its associated characteristics were evaluated.

Firstly, the DNA of the company's traceability record was examined, identifying its components and its relationship with other work products. Next, three specific analyses were conducted. The first one focused on the overall compliance of the traceability record, revealing that the company has deficiencies in 12.5% of the evaluated aspects.

The second analysis focused on identifying the characteristics of the traceability record and their compliance. As a result, it was found that the company does not comply with 20% of these characteristics, indicating significant areas for improvement.

The third analysis focuses on the coverage of improvement proposals implemented by the company to achieve a traceability record. As a result of this analysis, it was identified that the company needs to comply with 77% of the characteristics outlined in the improvement proposals.

This study highlights the importance of a solid and comprehensive traceability record in accordance with the ISO/IEC 29110 standard. The obtained results provide valuable information for the company in its recertification process and emphasize the need for corrective actions to address the identified deficiencies in the compliance of the traceability record and its specific characteristics.

1 Introduction.

Companies dedicated to software development are experiencing strong growth in line with the increasing demand for products in the sector [1]. Therefore, software quality management is gaining significant importance to ensure the proper functioning of computer systems.

In this regard, and from the perspective of Software Engineering, Ian Sommerville mentions that quality management can be achieved through established processes based on standards and quality procedures [2, p. 654]. Hence, such organizations

require certifications in standards or models that provide assurance of the quality of the generated software products and services [3].

In this context, the International Organization for Standardization (ISO) has created a standard specifically aimed at very small entities (***VSEs***) that focuses on the development of non-critical software: ISO/IEC 29110 [4]. This standard aims to assist entities of up to 25 individuals through the incorporation of two processes: Project Management (PM) and Software Implementation (SI). These processes encompass activities, tasks, and work products [5]. However, as indicated by M. Muñoz et al. [6], difficulties have been identified in its adoption due to lack of understanding, version control issues in components, functionalities, and documentation, test development, lack of tracking, among others. These issues pertain to incidents in areas such as traceability record, which have a higher incidence in software product defects [7].

G. Spanoudakis [8] defines software traceability as the 'ability to relate artifacts created during the development of the software system.' Furthermore, he states that traceability relationships help identify overlaps, satisfaction level, dependencies, evolution, generalization or refinement, conflicts, and rationalization. Additionally, he asserts that the traceability matrix is an important factor in supporting other activities. This element aims to improve the quality of software systems, generate consistent documentation, understand the development process, and easily identify process elements.

D. Méndez [9] states that the lack of development of a traceability record is one of the main problems leading to software product defects, accounting for 2.08% of them. For this reason, the standard incorporates a traceability record, which is addressed in this section and is also known as a traceability log.

2 Importance and problems of traceability record.

The traceability matrix is complex as it requires empirical investigation into organizational processes regarding the establishment of traceability relationships in the software development lifecycle.

G. Spanoudakis [8] defines software traceability as the 'ability to relate artifacts created during the development of the software system.' Additionally, he states that traceability relationships help identify couplings, satisfaction levels, dependencies, evolution, generalization or refinement, conflicts, and rationalization. Furthermore, he identified the challenges in constructing a traceability matrix, including high time and resource costs, difficulty in manually identifying relationships, lack of precise semantics, and lack of adoption of techniques, among others.

D. Méndez, et al [9] in their study, concluded that 27% of software failures are attributed to project management defects, of which 2.08% are specifically related to traceability record issues.

Regarding the ISO/IEC 29110 standard, the traceability matrix is generated from SI.3.3. However, it is suggested to start creating it from the project plan version, i.e., from PM.1.13, as it should contain information about the elements to be produced and the elements that should provide information for the traceability record, among others.

The main problems identified in creating a traceability matrix are summarized in Fig. 1.

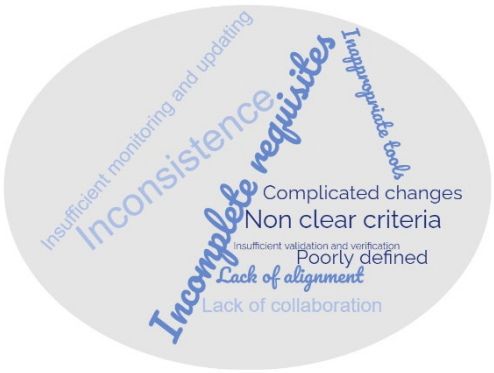

Fig. 1 – Common problems in creating a traceability matrix.

Another problem identified by G. Spanoudakis [8] relates to requirements, mainly due to incompleteness, inconsistency, and poor definition.

3 Traceability record DNA.

Within the standard implementation guide [5], traceability record is mentioned starting from SI.3.3. However, it is a part of software configuration. Although the standard does not explicitly mention it, this record should be continuously updated. According to the implementation guide, the minimum elements constituting a traceability record are requirements specification, software components, test cases, and procedures.

To identify other work products that need to be integrated into the traceability record, a comprehensive mapping of the standard's processes is necessary to identify the traceability of work products, as shown in Annex A.

While Table of the Annex A contains information about product issues, it does not address roles, work product typology, versions, and related activities, which are elements required to create the traceability matrix. Therefore, Table of the Annex B includes this information to focus the analysis solely on the traceability matrix and its constituent elements.

To enhance the understanding of the presented information, a diagram was created that encompasses the entire process mapping of the standard at the task level, including roles and work product states. Fig. 2 displays a portion of this mapping designed to identify and support Table of Annex A. For the complete mapping, refer to Annex C.

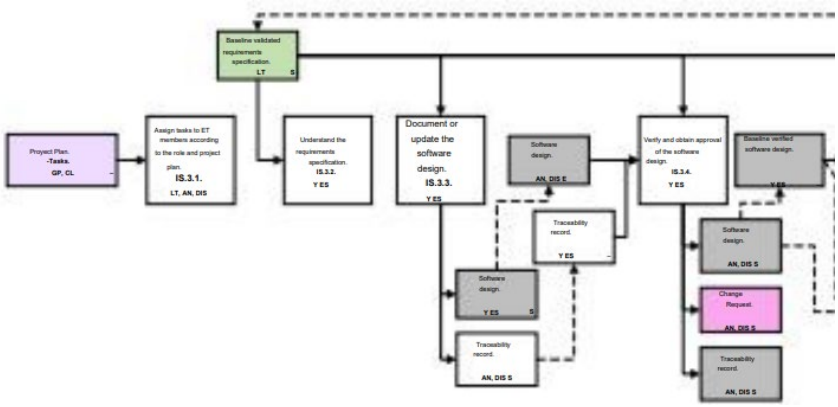

Fig. 2 – Standard process mapping.

In Fig. 2, a portion of the mapping is shown as it exceeds the dimensions of the page.

4 Proposal for improvement to the traceability record

In this section, the proposals for improvement to the traceability record are addressed following the identification of the work product DNA.

4.1 Template for traceability record in VSEs

S. Alexandre and C. Laporte [10] proposes a traceability record format (available in Annex D). According to the proposal, the record should have the following elements:

- 1st Page: Cover page.
  - Company logo.
  - Pagination.
  - Date of creation.
  - Process.
  - Code with nomenclature (standardized).
  - Versioning table for the traceability matrix format.
- 2nd Page: Traceability record.
  - Company logo.
  - Record ID.
  - Requirement ID.
  - Specification or requirement name.
  - Design component IDs.
  - Component name.
  - Code module or unit ID.
  - Unit test IDs.
  - Unit test results.

However, analyzing the proposed format along with the difficulties identified in creating the traceability record, the following issues that could hinder the traceability reading among elements are identified:

- Date and time of when an event occurred regarding a work product.
- Lack of an additional details' column.

- Excessive columns for identifiers, which provide little information about the version of the work product or associated tasks.
- No record of historical events that the work product has undergone.
- No typology of work products for the traceability of each element. By typology, we refer to whether the work product is an input, internal, or output.

Due to the aforementioned issues that could pose challenges in the development of the traceability record, a proposed traceability record format is presented below, aiming to address the identified difficulties.

4.2 Improvement proposal for the traceability record template

After analyzing the format and aiming to improve the traceability record format, it is important to define the relevant information to be recorded and audited. This format can be adapted to the needs and context of the EMP. Taking the above into consideration, the following fields are suggested to be added:

1. Event ID: A unique identifier for each recorded event, facilitating specific reference and search.
2. Date and Time: The exact moment when the event occurred, which is essential for temporal tracking and event correlation.
3. User or Responsible Entity: The name or identification of the person or entity who initiated or performed the event. This helps attribute responsibilities and provides a clear understanding of who was responsible in each case.
4. Event Description: A brief explanation or summary of the event or action taken. This provides key information about the nature of the event and its purpose.
5. Additional Details: Any relevant additional information related to the specific event. This may vary depending on the context but could include data such as location, resources used, configured parameters, among others.
6. Event Result or Status: An indicator of the resulting outcome or status of the event. For example, whether the event was successful, failed, or in progress.
7. Version or Version Number: If the event is related to a specific version of a product, software, or document, it may be useful to record the corresponding version.
8. Change Log: If modifications are made to the event over time, you can include a field to track and record the changes made, including the date and time of each modification.

Similarly, as proposed by M. García et al. [11] and as shown in Fig. 3, they propose a traceability metamodel to facilitate the construction of relationships between work products and the generic phases of software development (analysis, design, construction, and testing).

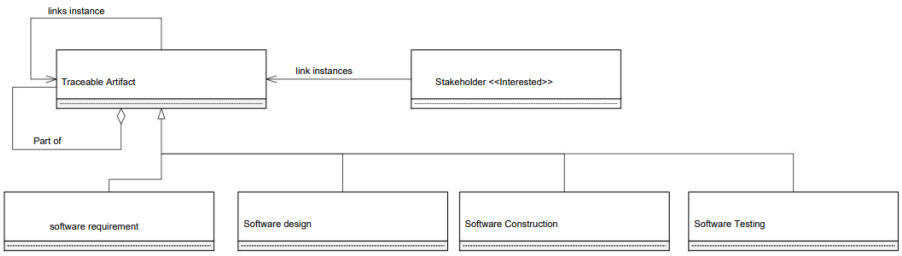

Fig. 3 – Traceability metamodel for requirements from the perspective of software development proposed by M. García, et al [11].

The stakeholder class represents the agents involved in the software project; considering that the analysis is based on the basic profile of the standard, these are the roles in this class.

The Traceable Artifact class refers to the elements that can be traced, such as documents: diagrams, components, and tests, among others. For other classes inherit this class:

- Software Requirements: Where all requirements engineering activities are performed (elicitation, analysis, specification, validation, and management).
- Software Design: Identifies all possible artifacts in the design phase.
- Software Construction: Contains all possible elements defined for construction.
- Software Testing: Identifies the elements to be tested in the testing phase.

Considering the proposed metamodel and with the aim of proposing a traceability record format with the elements mentioned in the standard, the existing relationships between the software development phases and the work products (artifacts) to which traceability must be performed were diagrammed, as shown in Fig. 4 (Reefer to Annex E).

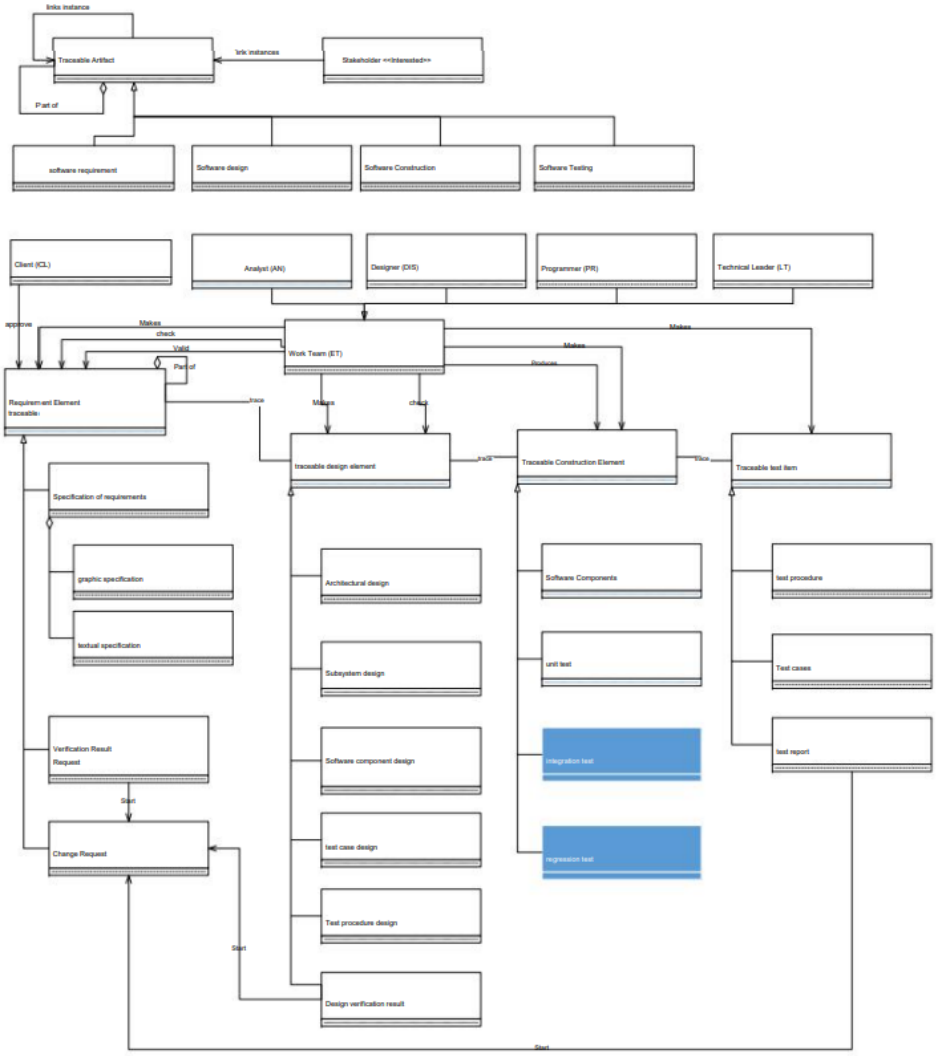

Fig. 4 – Traceability record model and its elements.

The proposed improvement to this format includes requirements, design, construction, and testing elements. Additionally, it is necessary to standardize the identifiers in this format, which should be standardized according to the company's criteria.

The file with proposed improvements for the traceability matrix format is available in Annex F.

5 Analysis of the traceability record.

This section mentions the analysis of the documentation provided by the company regarding the traceability record.

5.1 1st Analysis: Overall compliance of the related work products to the traceability record.

Based on the establishment of relationships and in accordance with the work products related to the traceability record, Table XLIX presents the analysis result, considering the scale defined in the ISO/IEC 29110-3 standard evaluation guide [12], "N" is Not fulfilled, "P" is Partially fulfilled, "L" is Largely fulfilled, and "T" is Fully fulfilled.".

Table 1. Compliance of the work products in the traceability record.

Process/Element	Result
Does the documentation contain change requests?	N
Does the documentation contain requirement specification?	L
Does the documentation contain verification results?	T
Does the documentation contain software design?	L
Does the documentation contain test cases?	L
Does the documentation contain test procedures?	L
Does the documentation contain software components?"	T

The results are shown in Fig. 5.

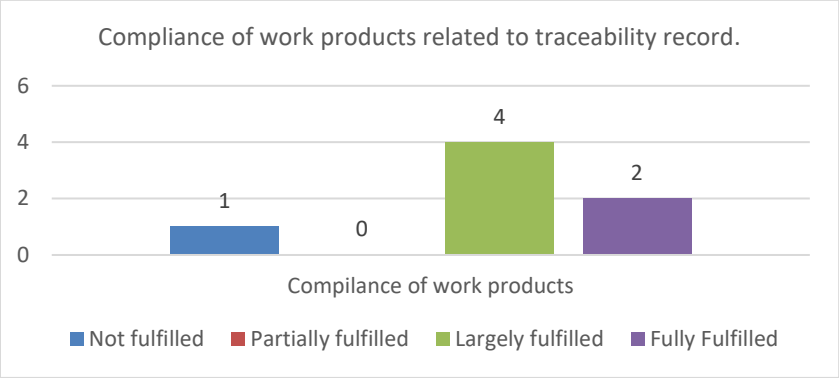

Fig. 5 – Results of compliance of work products related to traceability record.

From the above, it can be concluded that the company does not comply with the change request work product; as mentioned, it is included in their documentation but is empty; therefore, it is not considered a valid format for evaluation, representing 12.5%. On the other hand, the work products considerably achieved by the company are requirement specification, verification results, software design, test cases, and test procedures, representing 62.5%. Lastly, the work products that the company fully achieves are verification results and software components, representing 25%.

5.2 2nd Analysis: Compliance with traceability record characteristics.

Table 2 presents the analysis of the work product comparing the characteristics indicated by the standard implementation guide and those of the document provided by the company.

Table 2. Characteristics of the delivered traceability work product.

Work Product	Requirements specification	Software design elements	Software components	Test cases and test procedures	Status
Traceability record	X	X		X	X

As observed, as a result of the analysis, it is identified that the format provided by the company does not directly include the associated test cases but rather references the test cases and procedures. It is suggested to further break down this section in the format to facilitate the readability and execution of traceability.

Furthermore, when attempting to trace the elements, it is identified that the evolution over time of the elements comprising the traceability record cannot be observed.

In addition to the improvement findings and considering the characteristics of a traceability record, it should be readable from left to right and right to left. However, with the format provided by the company, it is difficult to establish traceability between the elements.

It can be concluded that the traceability record provided by the company only fulfills 80% of the required characteristics.

Table 3. Characteristics of the change request work product.

Work Product	Change proposal	Request status	Contact information of requester	Impacted systems	Impact on existing systems operation	Request Criticality and required date	Applicable states
Change request							

Within the documentation provided and as shown in Table 3, the change request form was located. Upon analysis of the document, it was identified that it does not meet any criteria as, although the document exists, it is empty.

Table 4. Characteristics of the change request work product.

Work Product	Introduction	Functionality	User interface	External Interfaces	ReliabilitEfficiency	Maintenance	Portability	Limitations	Reusability	Regulations	Status
Requirements specification	X	X	X	X	X	X	X	X	X		X

As can be observed in Table 4, this work product demonstrates a higher compliance with the required characteristics of the format. However, it does not include the aspect of reusability.

Table 5. Characteristics of the software design work product.

Work Product	Architectonic design	Required components	Considerations	Detailed design	Status
Software design	X	X	X		X

As can be observed in Table 5 and according to the implementation guide, the characteristics of the work product are considered and compared concerning the characteristics of the work product delivered by the company.

The next work product to analyze is Test Cases and Test Procedures; refer to Table 6 and Table 7.

Table 6. Characteristics of the test cases work product.

Work Product	Test case identifier	Test element	Input Specifications	Environmental requirements	Special procedures	Requirements interface dependencies	Status
Test cases	X	X	X	X	X		X

As can be seen from the analysis results of this work product, it is identified that the interface dependencies section is not met.

The Table 7 shows the analysis of the test procedures, considering the characteristics established by the implementation guide to compare it with the work product delivered by the company.

Table 7. Characteristics of the test procedures work product.

Work Product	Identifier	Descrption	End date	Possible issues	Person responsible for the procedure	Steps of procedure	Status
Test procedures	X	X	X	X	X	X	X

After analyzing the work product, it is identified that the test execution does not contain any possible issues identified in the implementation of its procedures.

Table 8 shows the analysis of the work product's software components, comparing the characteristics of the work product delivered by the company against those indicated in the implementation guide of the standard.

Table 8. Characteristics of the software components work product.

Work Product	Related code units
Software components	

After analyzing this work product, it is identified that the related code units are not found, which means that they are not included in the documentation, but the code segments are shown as image.

Table 9 presents the analysis of verification results work product.

Table 9. Characteristics of the verification results work product.

Work Product	Participants	Date	Location	Duration	Verification checklist	Approved items	Not approved items	Pending items	Identified defects	Status
Verification results	X	X	X	X	X	X	X	X	X	X

As can be seen, the verification results meet the characteristics indicated in the implementation guide. However, it is suggested to provide more detailed verification results by incorporating relevant information from each test.

The Fig. 6 resumes all the information.

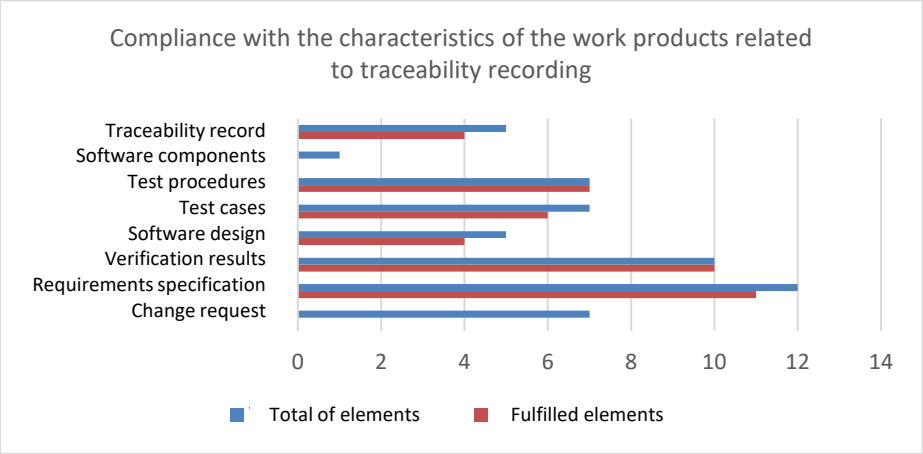

Fig. 6 – Results of compliance of the characteristics of the work products related to traceability record.

From Fig. 6, it is concluded that the products related to traceability recording 2 are completely non-compliant: software components and change requests. Despite the non-compliance of the related work products, the traceability record delivered by the company shows an 80% coverage of characteristics.

5.3 3rd Analysis: Compliance with traceability record

According to the improvement proposals and work products related to traceability recording, this section presents the analysis of compliance with the characteristics of the work products delivered by the company by comparing them against the improvement proposals. The results are described in Table 10.

Table 10. Characteristics of the verification results work product.

	ELEMENT TO COMPLY WITH							
WORK PRODUCT	EVENT IDENTIFIER	DATE AND HOUR	RESPONSIBLE USER	DESCRIPTION	ADDITIONAL RELEVANT DETAILS	EVENT STATUS	VERSION	CHANGE LOG
Verification results		X	X					X

	ELEMENT TO COMPLY WITH						
WORK PRODUCT	WORK PRODUCTS	WORK PRODUCT TYPOLOGIES	WORK PRODUCT STATUS	TEST CASES AND TEST PROCEDURES STATUS	TEST TYPES	EXECUTION RESULTS	EVENT TYPE
Standardization of identifier naming	X		X				

After analyzing the compliance of the improvement proposals regarding testing characteristics, it is identified that the company only fulfills five elements out of a total of characteristics. This represents a compliance rate of 33% of the characteristics, as shown in Fig. 7.

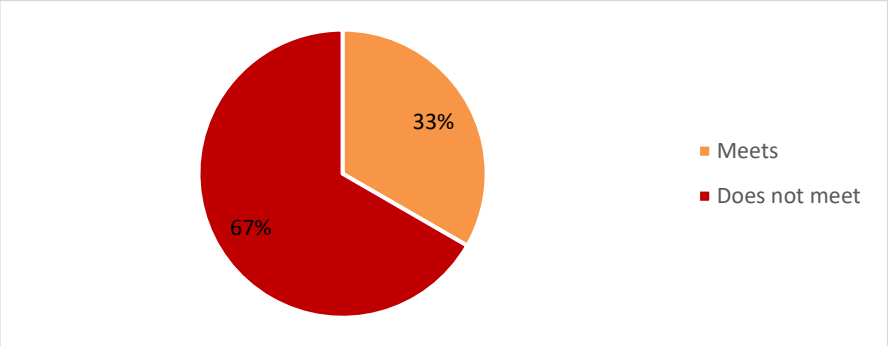

Fig. 7 – Characteristics fulfilled regarding traceability recording and improvement proposals.

From the analysis results, it is concluded that regarding the compliance of the work product characteristics according to the improvement proposals, there is an average compliance rate of 33.6%, reflecting the findings of improvements and issues encountered. It is affirmed that if the company implements the proposed improvements, it enhances the compliance of the work product characteristics and improves the company's performance in adopting and executing the standard. The improvement proposals provide the company with the means to meet the work products' required characteristics while improving certification outcomes.

6 Conclusions

In conclusion, companies in the software development sector are experiencing significant growth due to increasing product demand. As a result, software quality management becomes crucial to ensure the proper functioning of computer systems. Ian Sommerville emphasizes the importance of quality management through established processes and standards. Certifications in standards or models, such as ISO/IEC 29110, assure software quality.

However, difficulties have been identified in adopting ISO/IEC 29110, particularly in the traceability record. Traceability is essential for identifying software development overlaps, dependencies, and conflicts. The lack of a traceability record contributes to software defects.

Improving the traceability record requires a standardized format and inclusion of necessary information such as event ID, date and time, responsible entity, event description, additional details, event result, version, and change log. A proposed format should address the challenges of establishing traceability relationships and readability among elements.

The company's traceability record reveals areas of non-compliance, particularly in change requests, software components, and related code units. However, the traceability record shows partial compliance in requirement specification, software design, test cases, test procedures, and verification results.

Overall, there is a need for improvement in complying with the characteristics of the traceability record. Implementing proposed improvements will enhance compliance, improve software quality, and facilitate certification processes.

Annex A

Table 11. Incidence of work products.

Work product.	Type of product work.	Process, activity & origin task.	Destiny.	Notes.
Statement of work.	Input.		PM.1.1.	The source has no status and the destination changes to reviewed.
		PM.1.1.	PM.1.2.	It is received in a revised state, it does not change to its Output.
			PM.1.3.	It is received in a revised state, it does not change to its Output.
			PM.1.5.	It is received in a revised state, it does not change to its Output.
			PM.1.12.	It is received in a revised state, it does not change to its Output.
Software configuration.	Input.	SI.6.6.	PM.4.1.	From the origin it comes with a released status and when entering the destination, it changes to delivered.
Change request.	Input.	PM.3.3.	PM.2.2.	It is generated at the source with the status of started and changes to the evaluated status.
		PM.2.2.	PM.2.4.	When its status is evaluated, it changes to the accepted status.
Change request.	Internal.	PM.3.3.	PM.2.2.	It is generated at the source with the status of started and changes to the evaluated status.
		PM.2.2.	PM.2.4.	When its status is evaluated, it changes to the accepted status.
Corrective actions.	Internal.	PM.2.	PM.2.3.	It lacks state to the origin and destination.
			PM.3.2.	It lacks state to the origin and destina-

Work product.	Type of product work.	Process, activity & origin task.	Destiny.	Notes.
				tion.
Meeting minutes.	Internal.	PM.1.	PM.1.14	It lacks state to the origin and destination.
		PM.2.	PM.2.3.	At the source it has no status, at the destination it has a status of Updated.
Verificattion results.	Internal.	PM.1.	PM.1.13.	It lacks state to the origin and destination.
Advance report.	Internal.	PM.2.1.	PM.2.3.	It lacks state to the origin and destination.
			PM.2.4.	
			PM.3.1.	At the origin it lacks state, at the destination it changes to the evaluated state.
Backup of the project repository.	Internal.	PM.2.	PM.2.5.	It lacks state to the origin and destination.
			PM.2.6.	
Project plan.	Output.	PM.1.12.	PM.1.13.	At the origin it lacks status, at the destination it comes out with a Verified status.
		PM.1.13.	PM.1.14.	At the origin it is received with a Verified status, at the destination it leaves with an accepted status.
		PM.1.14.	PM.2.1.	At the origin it is received with a status of accepted, its status does not change.
			PM.3.1.	At the origin it is received with a status of accepted, its status does not change.
			PM.4.1.	At the origin it is received with a status of accepted, its status does not change.
Project plan.	Output.		PM.2.2.	At the origin it is received with a status of accepted, its status does not change.
		PM.1.14.	PM.2.2.	At the source it is received with the status of accepted, at the destination it changes to the

Work product.	Type of product work.	Process, activity & origin task.	Destiny.	Notes.
				Updated status.
		PM.2.2	PM.2.4.	At the source it is received with the Updated status, at the destination the status is maintained.
Acceptance minutes.	Output.	PM.4.	PM.4.1.	It lacks state to the origin and destination.
Project repository.	Output.	PM.1.	PM.1.15.	It lacks state to the origin and destination.
		PM.1.15.	PM.2.5.	At the source it is received without state, at the destination it changes to the recovered state.
		PM.1.15 / PM.2.5.	PM.4.2.	At the source it is stateless, at the destination it changes to the recovered state.
Meeting minutes.	Output.	PM.1. PM.2.	PM.2.3.	At the origin it lacks status, at the destination it comes out with a status of Updated.
Software configuration.	Output.	SI.6.6.	PM.4.1.	Status change from delivered to accepted.
		PM.4.1.	PM.4.2.	At the origin it is received with a delivered status, at the destination it leaves with an accepted status.
Project plan.	Input.	PM.1.14.	SI.1.1.	At the source it has the status of Verified at the destination it changes to the accepted status.
		SI.1.1	SI.1.2.	At the origin it has a status of accepted, at the destination it changes to reviewed.
			SI.2.1.	The revised state is maintained.
			SI.2.2.	The revised state is maintained.
			SI.3.1.	The revised state is maintained.
			SI.4.1.	The revised state is maintained.
			SI.5.1.	The revised state is maintained.
			SI.6.1.	The revised state is maintained.

Work product.	Type of product work.	Process, activity & origin task.	Destiny.	Notes.
Project repository.	Input.	PM.1.15 & PM.4.2.	SI.3.8.	At the source it has no status, at the destination it changes to Updated status.
Validation result.	Internal.	SI.2.	SI.2.4.	As Output we get the Validation results.
Verification result.	Internal.	SI.5.	SI.5.8.	It lacks state to the origin and destination.
			SI.5.10.	It lacks state to the origin and destination.
Software configuration.	Output.	SI.2.7.	SI.3.8., SI.4.7., SI.5.11	The source must have an updated status, the destinations have no status, however, it can be considered that the Updated status is maintained as new sections are added to it.
		SI.5.11.	SI.6.6.	At the source the status is Updated, at the destination it is released.
Change request.	Output.	SI.2.	SI.2.3., SI.2.6.	It is generated in activity 2 with status Updated.

Annex B

Table 12. Relationships between work products based on their types, states, and roles.

Process	Work product type	Activity and task related to the work product	Work product	Work producto states	Roles involved in the work product
PM.	Input	1.1	Job statement	Reviewed	PM, TL, CL
		4.1	Software configuration	Delivered	PM, CL
		2.2, 3.3	Change request	Initiated	CL, PM, TL
	Internal	2.2, 2.4	Change request	Evaluated, accepted	PM, TL, WT, CL

Process	Work product type	Activity and task related to the work product	Work product	Work producto states	Roles involved in the work product
		2.3, 3.2	Corrective actions	N/A	PM, TL, ET
		1.14, 2.3	Meeting minute	Updated	CL, WT, PM, TL
		1.13	Verification results	N/A	PM, TL
		2.1, 2.3, 2.4, 3.1, 3.2, 3.3	Progress report	Evaluated	PM, TL, WT, CL
		2.5, 2.6	Backup of project repository	Updated, recovered	PM, TL, ET
	Output	1.2 a 1.15, 2.2, 2.4	Project plan	Verified, accepted	PM, TL, CL, WT
		4.1	Acceptance minute	N/A	PM, CL
		1.15, 4.2, 2.6	Project repository	Backed up, Verified, Updated, recovered	PM, LT
		2.4	Meeting minute	Updated	PM, TL, WT, CL
		4.2	Software configuration	Accepted	GP
IS	Input	1.1	Project plan	Reviewed, accepted	PM, TL, ET
		3.8	Project repository	N/A	TL
	Internal	2.4	Validation results	Validated	CL, AN
		2.3, 2.6, 3.4, 5.8, 5.10	Verification results	Verified	AN, TL, CL, PR, DES
	Output	2.7, 3.8, 4.7, 5.11, 6.5, 6.6	Software configuration	Preliminary, Verified, released	TL, ET
		2.3, 2.6, 3.4	Change request	Proposed	AN, LT

Annex C

Standard process mapping link:
https://shorturl.at/quC04

Annex D

Template for traceability record in VSE's (S. Alexandre and C. Laporte proposal):
https://shorturl.at/duK69

Annex E

Model of traceability record and its elements:
https://shorturl.at/pQRX1

Annex F

Template for traceability record in VSE's (Our proposal):
https://shorturl.at/gwCMW

References

[1] Moisés Rodríguez, Mario Piattini, "Revisión sistemática sobre la certificación del producto de software," *Computer Science and Engineering,* no. 16-24, p. 9, Julio 2012.

[2] Ian Sommerville, Ingeniería de software, Novena ed., CDMX, CDMX: Pearson, 2011, pp. 1-792.

[3] Mirna Muñoz, Jezreel Mejía, Adriana Peña, Claude Laporte, Gloria Piedad Gasca-Hurtado, Maria Clara Gómez-Álvarez, "An Exploratory Analysis of the perception of the utility of proven practices of the software basic profile of ISO/IEC 29110 by a set of VSEs in Mexico," *Springer Nature Switzerland AG,* pp. 1-18, 2021.

[4] Apinporn Methawachananont, Marut Buranarach, Pakaimart AMSURIYA, Sompol CHAIMONGKHON, Kamthorn KRAIRAKSA ,Supnithi Thepchai, "Software Process Capability Self-Assessment Support System Based on Task and Work Product Characteristics: A Case Study of ISO/IEC 29110 Standard," *IEICE Transactions on Information and Systems,* vol. E103.D, pp. 339-347, Febrero 2020.

[5] ISO/IEC WG24, ISO/IEC TR 29110-5-1-2: Software engineering — Lifecycle profiles for Very Small Entities (VSEs) Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile., 1 ed., Switzerland, Geneva: IEEE, 2011, pp. 1-83.

[6] Mirna Muñoz, Patricia Montoya, "Identification of issues in the implementation of the ISO/IEC29119: comparison between the state of the art and the state of practice," in *2021 10th International Conference On Software Process Improvement (CIMPS)*, Torreón, Coahuila, México, 2021.

[7] Rupa Mahanti, Jiju Antony, "Confluence of six sigma, simulation and software development," *Managerial Auditing Journal,* vol. 20, no. 7, pp. 739-762, Septiembre 2005.

[8] GEORGE SPANOUDAKIS, ANDREA ZISMAN, SOFTWARE TRACEABILITY: A ROADMAP, Northampton square, 2005, pp. 1-34.

[9] Daniel Méndez Fernández, Stefan Wagner, Marcos Kalinowski, André Schekelmann, Ahmet Tuzcu, Tayana Conte, Rodrigo Spinola, Rafarl Prikladnicki, "Naming the Pain in Requirements Engineering: Comparing Practices in Brazil and Germany," *IEEE Software,* pp. 16-23, Septiembre 2015.

[10] Claude Y. Laporte, "Deployment Packages," 21 marzo 2023. [Online]. Available: https://profs.etsmtl.ca/claporte/english/vse/indexs.html. [Accessed 10 mayo 2023].

[11] M. D. P. H. Marco Antonio García, "Traceability Information Model for Very Small Entities with ISO/IEC 29110," *ICGDA,* pp. 1-6, 20 Abril 2018.

[12] ISO/IEC, Asesment guide ISO/IEC 29110-3, IEEE, 2018.