The ISO/IEC 29110-5-1-2 in the Capstone Project Course of Informatic Engineering Program

authors

Line
Line
Line
Line
Line

Abstract. The ISO/IEC 29110 standard has been developed for small organizations and establishes a reasonable minimum set of practices to be performed in small system or software development projects. However, few efforts have been made in the academic context of university adoption. This study assessed compliance with the ISO/IEC 29110 Software Engineering Basic Profile (SEBP) practices in a final year course of the informatic engineering program. The course is developed under the project-oriented learning (POL) model and is an integrative course at the end of the curriculum. After assessing five student projects, it is determined that there is a high level of compliance with SEBP, which allows us to conclude that the course achieves its purpose. In addition, after performing a critical analysis, some improvements and limitations to be applied in the course were identified.

Keywords: ISO/IEC 29110, Software engineering education, academic experience.

1 Introduction

ISO/IEC 29110 is a set of standards developed for small organizations that develop or maintain systems or software [1]. Their publication started in 2011 but still there are some documents under development. [2]. ISO/IEC 29110 covers two disciplines: software engineering and systems engineering, and establishes a set of four profiles (input, basic, intermediate and advanced) [3]. In addition, it has only one profile for IT services management. Nowadays, the Basic profile of the software engineering discipline is the only one that can be certified [4].

For the software industry, which consists mainly of micro, small and medium-sized enterprises [1], ISO/IEC 29110 represents an important support tool. This Standard establishes a relevant minimum set of practices to be taken into account in software development projects [1]. In particular, for developing countries, as noted [5], having process standards will contribute to increasing the competitiveness of software-developing SMEs. However, to achieve these achievements, it is also necessary for software developers to apply the practices set out in this model [5], which in this case is ISO/IEC 29110-5-1.

In this context, the university is primarily responsible for the quality training of professionals working in the software industry [5]. It is expected that the curricula will be suitable for such tasks and have recognition. In this sense, it is important for the accreditation of engineering programs that the curricula include at least one course where students put into practice the knowledge acquired. In particular, as suggested by engineering accreditors [6], [7], engineering study programs should offer an integrative course at the end of the degree, where students have to solve a complex problem with multiple constraints and applying the corresponding standards and models.

In this study, we present the analysis of the assessment of compliance with the Basic Profile of software engineering standard ISO/IEC 29110-5.1-2 of five software development projects carried out at an integrating course of the 9th cycle of a computer engineering program. The article is organized as follows: in Section 2, the Reference Framework is presented; in Section 3, the course is described; in Section 4, the evaluation of the projects is shown; in Section 5, conclusions are presented.

2 Background

This section summarizes the software engineering basic profile (SEBP) of ISO/IEC 29110, the project-oriented learning (POL) method, and the capstone project course of an accredited engineering program.

2.1 ISO/IEC 29110 Software Engineering Basic Profile (SEBP)

ISO/IEC 29110 defines a set of 4 process profiles that have been defined for small organizations (such as companies) that develop software or systems and have fewer than 25 employees [1]. The software engineering Basic profile (SEBP) is one of the certifiable profiles of ISO/IEC 29110-5-1-2 [8], with which several organizations have been certified, in different parts of the world [9], [10], [11], [4].

The SEBP consists of two processes: Software Implementation (SI) and Project Management (PM) [12]. This profile is what characterizes a small organization (e.g. a small company) developing only one non-critical software project at a time [1].

2.2 Project-Oriented Learning

Project Oriented Learning (POL) is defined as a didactic learning method in which students take greater responsibility for their own learning process, by applying in a project their previously acquired skills and knowledge [13], [14], [15], [16]. For a project to be of the POL type [15], [17], it must be fulfilled that the project: (i) is the central component and not peripheral to the curriculum, (ii) should focus on problems that induce students to confront the basic concepts and principles of one or more disciplines, (iii) involves students in a creative research process, and (iv) is directed, to a large extent, by the students themselves. In addition, in [18], it is noted that students solve diverse and complex problems to complete their project by: (i) asking and debugging questions; (ii) designing plans or experiments; (iii) making predictions; (iv) collecting and analyzing data; (v) discussing ideas; (vi) drawing conclusions, (vii) communicating ideas, (viii) asking new questions, and (ix) creating artifacts.

A study on engineering education made by the Massachusetts Institute of Technology [19], highlights, among other trends, the adoption of methods based or similar to POL. The study in [19] shows that leading engineering education institutions have introduced POL into their programs. These courses provide strong support for students to reflect, contextualize, and apply knowledge and skills; but they require careful supervision [19].

2.3 Capstone Project Course

A capstone project course (CPC), according to [20], is a course offered at the end of the academic program (when the student is close to graduation). In this course, a problem is solved through an important project or complex task that allows students to integrate knowledge and skills acquired during their student stage.

In addition, in [7], it is noted that a CPC must offer engineering design experience, and requires that the level of achievement of a set of student outcomes (RE for Results) be measured. Therefore, according to the pedagogical design of the course, the authentic situation (statement of the problem) must be carefully established so that it can be verified what is expected of the student at the end of the course.

3 Course Description

The study was carried out on a set of projects developed in a course called "Software Development Project", with the following characteristics:

- The course is part of the 9th cycle of the curriculum of a computer engineering program which is accredited for more than 20 years. These recognitions were granted by engineering accreditations such as ABET and ICACIT, and they remain to date.
- The course is developed under the didactic methodology of POL, with the project being the central theme of the course throughout the academic period.
- The course is declared as capstone project course (CPC) so it is developed as a major design project in engineering that complies with two accrediting organizations: ABET and ICACIT.
- The course has declared the measurement of a RE called numerical experimentation. This student result has motivated the design of the problems (authentic situation) to have certain special characteristics.

Considering the above, it can be pointed out that the purpose of the course is to offer a learning space through a complex engineering problem with multiple restrictions that allows the integration of knowledge and skills acquired in the academic program, and requires the use of engineering standards. This purpose is largely achieved by using the didactic technique of POL, however, the subject of alignment to the standard is a subject that has not been formally evaluated.

3.1 Description of course projects

Considering what is stablished in the syllabus of the course, especially what is related to the RE of numerical experimentation, the statement of the problem (authentic situation) describes a scenario that corresponds to the Vehicle Routing Problem (VRP) type. The authentic situation includes variations of some VRP so that its complexity increases and, above all, the solution is different from that of other academic periods. These types of problems (VRP) are solved, among others, by meta-heuristic algorithms [21], [22], [23].

In the statement of the authentic situation, the need for planning and visualizing the movement of vehicles as planned is stablished. This situation implies that the solution has two major components: the planner that solves the established problem and the visualizer that is in charge of presenting the solution as a map where the movement of the vehicles can be seen. To solve the scheduler problem, a non-functional requirement is established for them: to propose two meta-heuristic algorithms and to decide the best algorithm based on a numerical experimentation. To implement the visualization component, teams must consider three work scenarios, such as in a city parcel delivery system: (a) a first scenario is the daily operation, where a logistics worker looks at the operations in real-time on a map and interacts with the vehicles; (b) A second scenario is the weekly simulation where 7-day trades are displayed within 30 minutes.; and (c) a third scenario is called simulation-collapse when operations reach logistic collapse in package distribution using data generated with growth rates.

In addition, the requirement to use ISO/IEC 29110-5-1-2 has been indicated and students have been provided with a template to verify compliance with the standard.

From the perspective of work organization, the teams are made up, in principle, of four students; an amount that depends on the number of students enrolled in the course. Every week they have a presentation of project progress according to the schedule established at the beginning of the course. The schedule presents four stages. Stage I Project formulation and solution definition (weeks 1-4). Stage II Selection and design of the solution- (weeks 5-7). Stage III Construction and testing (weeks 8-12). Stage IV Delivery and final presentation (weeks 13-15).

At the beginning of the academic period, the statement of the authentic situation is explained to the students, and they are given three templates: one for recording and updating the project risks, another for registering project hours, and the third for self-assessment of the progress of the project regarding compliance of the SEBP of ISO/IEC 29110-5-1-2. This last template must be completed by students, and presented at 3 different moments of the course.

3.2 Students Project Deliverables

The project establishes a set of technical documents without specifying its internal structure. For this, students have taken previous courses where they have been taught these documents in different methodologies. Then, it is the responsibility of the students to establish the internal structure of the requested documents, based on the software development methodology they have chosen. Students are also told that the minimum document structures are given in Section 9 of ISO/IEC 29110-5-1-2. The required documents for the course are:

- DP report - Product definition report.
- PT Report - Prototype definition report.
- CP Report - Product Comparison Report
- LE Report - Requirements List (similar to a software requirements specification)
- PP Report - Detailed project plan.
- PGC Report - Configuration management plan.
- ISA Report - Algorithm Selection Report.
- Analysis report.
- Design report.
- Evidence case report.

In addition, students had been told that they can use equivalent documents, in case they opt for a particular methodology. As they are delivered, all documents are reviewed and can be resubmitted as a new version, if required.

4 Projects Assessment

For the study, an assessment of ISO/IEC 29110-5-1-2 practices was carried out for each project. Then, a critical analysis of the academic experience was performed.

3.1 SEBP Assessment

The ISO/IEC 33020 [24] process attribute ratings were used as a reference for the assessment. This rating and its corresponding values are [24]: [N] Not achieved (0 to 15% achievement), [P] Partially achieved (>15% to 50% achievement), [L] Largely achieved (>50% to 85% achievement) and [F] Fully achieved (>50% to 85% achievement).

ISO/IEC 33020 [24] uses these values to qualify process attributes and no other process elements. However, we decided to assess the students' practices in the course on the same scale. On that basis, the following rating was used:

- [N] Not achieved, did not perform or did almost nothing.
- [P] Partially achieved, the practice was partially performed, and this is considered insufficient.
- [L] Long achieved, the indispensable has been done, but there is still room for improvement.
- [F] Fully accomplished, it has been done more than necessary, and is considered complete (sufficient).

To carry out the assessment, the project deliverables, the course working meeting log and the agreements made in the progress reviews, stored in the repository course, were used as a basis.

3.2 Task Analyze

The assessment was performed on the five work teams of the course. Students used a spreadsheet to record their progress against ISO/IEC 29110 practices. The assessment results of this study were recorded in the same spreadsheet. To facilitate the analysis, one letter has been used for each project (A, B, C, D, E).

Table 1 and Table 2 present the consolidated results of the five project assessments carried out at the end of the course, based on project management and software implementation processes, respectively. Both tables have the same structure as explained below. The "Id" column, has the task identifier according to 29110-5-1-2 [8]. In the column "Project Rating", the activity rating of each project {A, B, C, D, E} are presented, according to the defined classification values {NA, N, P, L, F}. Finally, the column "Project Count" presents the number of projects that reached each value {~, N, P, L, F}, where ~ represents Not applicable.

Table 1. Tasks assessment - Project Management

Id	Project Rating					Projects count				
	NA	N	P	L	F	~	N	P	L	F
PM.1.1	--	--	--	--	**ABCDE**	--	--	--	--	**5**
PM.1.2	--	--	--	--	**ABCDE**	--	--	--	--	**5**
PM.1.3	--	--	--	--	**ABCDE**	--	--	--	--	**5**
PM.1.4	--	--	--	**ABCD-**	**----E**	--	--	--	**4**	**1**
PM.1.5	--	--	--	**A----**	**-BCDE**	--	--	--	**1**	**4**
PM.1.6	--	--	--	--	**ABCDE**	--	--	--	--	**5**
PM.1.7	--	--	--	**A----**	**-BCDE**	--	--	--	**1**	**4**
PM.1.8	--	**-B-DE**	**A-C--**	--	--	--	**3**	**2**	--	--
PM.1.9	--	--	--	--	**ABCDE**	--	--	--	--	**5**
PM.1.10	--	--	--	**A-CDE**	**-B---**	--	--	--	**4**	**1**
PM.1.11	--	--	--	**-B---**	**A-CDE**	--	--	--	**1**	**4**
PM.1.12	--	--	--	**A----**	**-BCDE**	--	--	--	**1**	**4**
PM.1.13	--	--	--	**A-C--**	**-B-DE**	--	--	--	**2**	**3**
PM.1.14	--	--	--	**----E**	**ABCD-**	--	--	--	**1**	**4**
PM.1.15	--	--	--	**-B--E**	**A-CD-**	--	--	--	**2**	**3**
PM.2.1	--	--	**---D-**	**ABC-E**	--	--	--	1	**4**	--
PM.2.2	--	--	--	--	**ABCDE**	--	--	--	--	**5**
PM.2.3	--	--	--	**AB---**	**--CDE**	--	--	--	**2**	**3**
PM.2.4	--	--	--	**AB---**	**--CDE**	--	--	--	**2**	**3**
PM.2.5	--	--	--	**--CDE**	**AB---**	--	--	--	**3**	**2**
PM.2.6	--	--	--	**--C-E**	**AB-D-**	--	--	--	**2**	**3**
PM.3.1	--	**---D-**	**ABC-E**	--	--	--	1	4	--	--
PM.3.2	--	--	--	--	**ABCDE**	--	--	--	--	**5**
PM.3.3	--	--	--	--	**ABCDE**	--	--	--	--	**5**
PM.4.1.	--	--	--	**A----**	**-BCDE**	--	--	--	**1**	**4**
PM.4.2	--	--	--	--	**ABCDE**	--	--	--	--	**5**

Table 2. Tasks assessment - Software Implementation

Id	Project Rating					Projects count				
	NA	N	P	L	F	~	N	P	L	F
SI.1.1	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.1.2	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.2.1	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.2.2	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.2.3	--	--	--	**---DE**	**ABC--**	--	--	--	**2**	**3**
SI.2.4	--	--	--	**----E**	**ABCD-**	--	--	--	**1**	**4**
SI.2.5	**ABCDE**	--	--	--	--	**5**	--	--	--	--
SI.2.6	**ABCDE**	--	--	--	--	**5**	--	--	--	--
SI.2.7	--	--	--	**--C-E**	**AB-D-**	--	--	--	**2**	**3**
SI.3.1	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.3.2	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.3.3	--	--	--	**A----**	**-BCDE**	--	--	--	**1**	**4**
SI.3.4	--	--	--	**A---E**	**-BCD-**	--	--	--	**2**	**3**
SI.3.5	--	--	--	**-BCDE**	**A----**	--	--	--	**4**	**1**
SI.3.6	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.3.7	--	**ABCDE**	--	--	--	--	**5**	--	--	--
SI.3.8	--	--	--	**-BC-E**	**A--D-**	--	--	--	**3**	**2**
SI.4.1	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.4.2	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.4.3	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.4.4	--	--	--	**AB--E**	**--CD-**	--	--	--	**3**	**2**
SI.4.5	--	--	--	**----E**	**ABCD-**	--	--	--	**1**	**4**
SI.4.6	--	**ABCDE**	--	--	--	--	**5**	--	--	--
SI.4.7	--	--	--	**ABCDE**	--	--	--	--	**5**	--
SI.5.1	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.5.2	--	--	--	**A-C-E**	**-B-D-**	--	--	--	**3**	**2**
SI.5.3	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.5.4	--	--	--	**----E**	**ABCD-**	--	--	--	**1**	**4**
SI.5.5	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.5.6	--	**ABC-E**	--	--	**---D-**	--	**4**	--	--	**1**
SI.5.7	--	**ABCDE**	--	--	--	--	**5**	--	--	--
SI.5.8	--	**ABCDE**	--	--	--	--	**5**	--	--	--
SI.5.9	--	**ABCDE**	--	--	--	--	**5**	--	--	--
SI.5.10	--	**ABCDE**	--	--	--	--	**5**	--	--	--
SI.5.11	--	--	--	**ABCDE**	--	--	--	--	**5**	--
SI.6.1	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.6.2	--	--	--	--	**ABCDE**	--	--	--	--	**5**
SI.6.3	--	**ABCDE**	--	--	--	--	**5**	--	--	--
SI.6.4	--	**ABCDE**	--	--	--	--	**5**	--	--	--
SI.6.5	--	**ABCDE**	--	--	--	--	**5**	--	--	--
SI.6.6	--	--	--	**---D-**	**ABC-E**	--	--	--	**1**	**4**

In the case of the project management (PM) process, Table 1 shows a high degree of adhesion to the standard. It can be observed that:

- the project effort and cost estimate task PM.1.8 is not formally carried out. Students consider an internal timeline of the team they have prepared for their project. The method used for the effort and cost estimation could not be identified in the reports. However, there is a "course calendar" that sets the milestones off a group of project activities. This situation may influence the failure to determine effort and cost estimation,
- the project progress assessment task PM.3.1 is performed using a single project variable that is completed from the planned tasks. No other project variables are evaluated.

In the case of software implementation (SI) process, Table 2 shows clearly that there is a group of practices that is fulfilled and a smaller group that is not. In particular, it may be mentioned that:

- Tasks SI.2.5. and SI.2.6 are declared optional in the SEBP and refer to user documentation and its verification. Due to the complexity and workload of the project, student teams drop it out. With respect to the course project, it is understood that: (a) the purpose of the planning algorithm is to solve a complicated problem; (b) requires the implementation of two metaheuristic algorithms and the selection of an algorithm using numerical experimentation; (c) the diagrams are presented as a prototype of the solution covering the 3 scenarios (everyday operations, weekly simulation and collapse simulation); and (d) the expected solution does not require much interaction with users,
- SI.3.7, SI.4.6 and SI.5.6 are tasks related to traceability registration. This practice was not implemented by students. To understand the context, keep in mind that: (a) projects must be completed in the course (approx. 16 weeks); (b) There is no client or user to use the solution after it is implemented; (c) the proposed problem is not real, it is a realistic problem, thus it is not a concrete solution for any client,
- Tasks SI.5.7 and SI.5.8 refer to the Operations Guide, and tasks SI.5.9 and SI.5.10 refer to the User Guide, which have not been performed on the projects. Tasks SI.6.3, SI.6.4 and SI.6.5 also refer to maintenance documentation. The final solution was delivered in the form of a version deployed on a server and a demonstration of the software and its functioning was performed in accordance with the scenarios established in the course. This has not contributed to any maintenance documentation tasks.

- **3.3 Course Analyze from SEBP**

  At the course “Software Development Project” a real problem is proposed to be solved through a software project. In the project, students work as software engineers and the professor assumes many roles such as: a user, a reviewer or a client. Tables A.1 and A.2 (see Appendix A) show the tasks in which the professor has more involvement, respectively in the project management and software implementation processes. This implies that the professor has to be permanently involved in the work teams' projects.

5 Conclusions

The assessments performed on the five projects showed that most of the SEBP's practices are carried out by working groups. In this sense, the use of a SEBP in student projects is carried out in the Capstone Project course, in accordance with the ABET or ICACIT criteria.

In addition, after a critical analysis, based on the nature of the course, it can be observed that the tasks: (a) SI.2.5 and SI.2.6 were not considered by the students because they were optional and referred to a preliminary User Manual; (b) SI.5.7 and SI.5.8 should be incorporated as a non-functional requirement that implies to develop an instructive (or similar) document for deployment of the solution; (c) SI.5.9 and SI.5.10 could be incorporated as videos to be submitted, explaining the 3 scenarios foreseen in the problem; (d) SI.6.3, SI.6.4 and SI.6.5 should be incorporated in the form of a template containing the minimum elements of a maintenance manual for such solutions; (e) SI.3.7, SI.4.6 and SI.5.6 should be indicated in the working proposal as a non-functional requirement; (f) PM.1.8 should be indicated in the work proposal as a non-functional requirement requiring effort and cost estimation in the project plan; and, (g) PM.3.1 should be indicated in the work proposal as a non-functional requirement as a worksheet for monitoring progress with more than one variable.

Based on the review of professor participation, some annotations should be included in the capstone project course to reinforce compliance with SEBP practices. Also, it is necessary to show the importance of tasks, leaving the context realistic and assuming an actual context.

References

1. ISO/IEC: ISO/IEC TR 29110-1:2016 Systems and Software Engineering -- Lifecycle Profiles for Very Small Entities (VSEs) -- Part 1: Overview. , Geneva (2016).
2. ISO: ISO/IEC 29110 in ISO Web, https://www.iso.org/search.html?q=29110, last accessed 2023/06/13.
3. O'Connor, R., Laporte, C.Y.: The Evolution of the ISO/IEC 29110 Set of Standards and Guides. International Journal of Information Technologies and Systems Approach. 10, 1–21 (2017). https://doi.org/10.4018/IJITSA.2017010101.
4. NYCE: Certificación ISO/IEC 29110, https://www.nyce.org.mx/certificacion-isoiec-29110/, last accessed 2023/06/13.
5. UNCTAD: Information Economy Report 2012. The Software Industry and Developing Countries. (2012).
6. ABET: Criteria for Accrediting Engineering Programs, 2023 – 2024.
7. ICACIT: Criterios de Acreditación de ICACIT para Programas de Pregrado. , Lima (2023).
8. ISO/IEC: ISO/IEC TR 29110-5-1-2:2011 Software Engineering — Lifecycle Profiles for Very Small Entities (VSEs) — Part 5-1-2: Management and Engineering Guide: Generic Profile Group: Basic Profile. , Geneva (2011).
9. Rodríguez-Dapena, P., Buitrago-Botero, M.F.: How to Certify the very small Entity software processes using ISO/IEC 29110. In: Communications in Computer and Information Science. pp. 245–252. Springer Verlag (2015). https://doi.org/10.1007/978-3-319-19860-6_20.

10. Mejía, J., Bonilla, E., Faustino, I., Jhordany, E., Villanueva, E.: Supporting vses to assess internal processes and projects for certification in the standard iso/iec 29110. RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao. 2021, 80–96 (2021). https://doi.org/10.17013/RISTI.41.80-96.
11. Muñoz, M., Mejia, J., Laporte, C.Y.: Implementation of ISO/IEC 29110 in software development centers from Mexican universities: An experience of the Zacatecas State. RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao. 43–54 (2018). https://doi.org/10.17013/risti.29.43-54.
12. Laporte, C.Y., Muñoz, M., Mejia, J., Oconnor, R. V.: Applying Software Engineering Standards in Very Small Entities. From Startups to Grownups, (2018). https://doi.org/10.1109/MS.2017.4541041.
13. Almulla, M.A.: The Effectiveness of the Project-Based Learning (PBL) Approach as a Way to Engage Students in Learning. Sage Open. 10, (2020). https://doi.org/10.1177/2158244020938702.
14. ITESM: El método de proyectos como técnica didáctica. ITESM (2013).
15. Sharma, A., Dutt, H., Naveen Venkat Sai, C., Naik, S.M.: Impact of project based learning methodology in engineering. Procedia Comput Sci. 172, 922–926 (2020). https://doi.org/10.1016/j.procs.2020.05.133.
16. Bippert, K., Espinosa, T.: Project-Oriented Learning & Teaching: Expectations vs. Reality. In: Texas Association for Literacy Education Yearbook. pp. 19–29 (2019).
17. Thomas, J.W.: A Review of Research on Project- Based Learning. (2000).
18. Blumenfeld, P.C., Soloway, E., Marx. Ronald W., Krajcik, J.S., Guzdial, M., Palincsar, A.: Motivating Project-Based Learning: Sustaining the doing, Supporting the Learning. Educ Psychol. 26, 369–398 (1991).
19. Graham, R.: The global state of the art in engineering education. (2018).
20. ABET: ABET - Glossary, https://www.abet.org/program-evaluators/training/glossary, last accessed 2023/06/24.
21. Liong, C.Y., Wan Rosmanira, I., Khairuddin, O., Mourad, Z.: Vehicle Routing Problem: Models and Solutions. Journal of Quality Measurement and Analysis JQMA. 4, 205–218 (2008).
22. Konstantakopoulos, G.D., Gayialis, S.P., Kechagias, E.P.: Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. Operational Research. 22, 2033–2062 (2022). https://doi.org/10.1007/s12351-020-00600-7.
23. Alba, E., Dorronsoro, B.: Logistics: The Vehicle Routing Problem. In: Cellular genetic algorithms. pp. 176–186 (2008). https://doi.org/10.1007/978-0-387-77610-1.
24. ISO/IEC: ISO/IEC 33020:2019 Information technology — Process assessment — Process measurement framework for assessment of process capability. 2da Ed. , Geneva (2019).

Appendix A: Professor involvement in tasks of SEBP. URL

Note to reviewers. The appendix below has only been included for peer-review process. The appendix will be available by URL in a camera ready and published version.

Appendix A: Professor involvement in tasks of SEBP

Table A.1. Professor involvement in task of SEBP-Project Management.

Id	Task (truncate description)	Doc	Est
PM.1.1	Review the statement of work	--	X
PM.1.2	Define the delivery instructions…	X	--
PM.1.3	Identify the specific tasks…	--	X
PM.1.4	Establish the estimated duration to perform …	--	X
PM.1.5	Identify and document the resources….	--	X
PM.1.6	Establish the composition of work team…	--	X
PM.1.7	Assign estimated start and completion dates…	--	X
PM.1.8	Calculate and document the project estimated…	--	X
PM.1.9	Identify and document the risks which may ...	--	X
PM.1.10	Document the version control strategy..	--	X
PM.1.11	Generate the project plan …	--	X
PM.1.12	Include product description, scope, objectives …	--	X
PM.1.13	Verify and obtain approval of the project ..	X	--
PM.1.14	Review and accept the project plan..	--	X
PM.1.15	Establish the project repository using the version …	--	X
PM.2.1	Monitor the project plan execution and record …	--	X
PM.2.2	Analyse and evaluate the change request …	X	--
PM.2.3	Conduct revision meetings with the work team…	--	X
PM.2.4	Conduct revision meetings with the customer…	X	--
PM.2.5	Perform backup according to the version …	--	X
PM.2.6	Perform project repository recovery …	--	X
PM.3.1	Evaluate project progress …	--	X
PM.3.2	Establish actions to correct deviations …	--	X
PM.3.3	Identify changes to requirements…	X	--
PM.4.1.	Formalize the completion …	X	--
PM.4.2	Update project repository.	--	X

Table A.2. Professor involvement in task of SEBP-Software Implementation

Id	Task (truncate description)	Doc	Est
SI.1.1	Revision of the current project plan with …	--	X
SI.1.2	Set or update the implementation environment.	X	--
SI.2.1	Assign tasks to the work team members …	--	X
SI.2.2	Document or update the requirements …	--	X
SI.2.3	Verify and obtain approval of the requirements...	X	--
SI.2.4	Validate and obtain approval of the requirements …	X	--
SI.2.5	Document the preliminary version of the software ...	--	X
SI.2.6	Verify and obtain approval of the software ...	X	--
SI.2.7	Incorporate the requirements specification …	--	X
SI.3.1	Assign tasks to the work team members related ...	--	X
SI.3.2	Understand requirements specifications.	--	X
SI.3.3	Document or update the software design ...	--	X
SI.3.4	Verify and obtain approval of the software design …	X	--
SI.3.5	Establish or update test cases and test …	--	X
SI.3.6	Verify and obtain approval of the test cases ...	X	--
SI.3.7	Update the traceability record incorporating …	--	--
SI.3.8	Incorporate the software design, and traceability ...	--	X
SI.4.1	Assign tasks to the work team members	--	X
SI.4.2	Understand software design.	--	X
SI.4.3	Construct or update software ...	--	X
SI.4.4	Design or update unit test cases …	--	X
SI.4.5	Correct the defects found until successful …	--	X
SI.4.6	Update the traceability record incorporating ...	--	X
SI.4.7	Incorporate software components and traceability ...	--	X
SI.5.1	Assign tasks to the work team members related …	--	X
SI.5.2	Understand test cases and test procedures ...	--	X
SI.5.3	Integrates the software using software components ...	--	X
SI.5.4	Perform software tests using test cases ...	--	X
SI.5.5	Correct the defects found and perform regression…	--	X
SI.5.6	Updates the traceability record if appropriate…	--	X
SI.5.7	Document the product operation guide …	--	X
SI.5.8	Verify and obtain approval of the product operation ...	X	--
SI.5.9	Document the software user documentation …	--	--
SI.5.10	Verify and obtain approval of the software ...	X	--
SI.5.11	Incorporate the test cases and test procedures ...	--	X
SI.6.1	Assign tasks to the work team members …	--	X
SI.6.2	Understand software configuration ...	--	X
SI.6.3	Document the maintenance documentation ...	--	X
SI.6.4	Verify and obtain approval of the maintenance ...	X	--
SI.6.5	Incorporate the maintenance documentation ...	--	X
SI.6.6	Perform delivery according to delivery …	X	--