UP-VSE: A Unified Process - Based Lifecycle Model for Very Small Entities

Jhon Alvarez(✉) and Julio Hurtado(✉)

Departamento de Sistemas, Universidad del Cauca, Popayán, Colombia
{jjalvarezl,ahurtado}@unicauca.edu.co
http://artemisa.unicauca.edu.co/~jjalvarezl
http://artemisa.unicauca.edu.co/~ahurtado

Abstract. In an attempt to increase the competitiveness of small organizations, the ISO has developed the ISO 29110 standard. However, this represents a new challenge for Very Small Entities (VSE) because this new reference model to be learnt, applied and adopted. This paper presents a VSE lifecycle model based on UP in order to reduce some effort in organizations that previously know and have used the unified process. This model implements the technical practices of ISO/IEC 29110-5-1-1:2012 providing to software companies providing to software organizations a ready to use ISO 29110 implementation. This proposal has been empirically assessed by defining the Software Development Process of the Industrial Automation Engineering Group (PRODIGIA).

Keywords: UP-VSE · Software process improvement
Unified process · ISO 29110 · VSE

List of Acronyms

AUP	Agile Unified Process
AVISPA	Analysis and Visualization for Software Process Assessment
CMMI	Capability Maturity Model Integration
EPF	Eclipse Process Framework
PRODIGIA	Software Development Process of the Industrial Automation Engineering Group
RUP	Rational Unified Process
SPI	Software Process Improvement
UP	Unified Process
UP-VSE	VSE lifecycle model based on Unified Process
UPEDU	Unified Process for EDUcation
VSE	Very Small Entities

J. E. Serrano C. and J. C. Martínez-Santos (Eds.): CCC 2018, CCIS 885, pp. 260–275, 2018.
https://doi.org/10.1007/978-3-319-98998-3_21

1 Introduction

Achieve quality products and productive teams in a competitive software industry normally require a focus on the software process, under the premises that the processes determine the quality of the products [1] and the interdependent relationship between the process and the project productivity [2]. According to Jacobson et al. "A process defines who is doing what, when and how to build a software product or improving an existing one" [3]. Although VSE represent the 75% out of the software development industry, however the software industry has created standards, such as, ISO/IEC 12207 [4] and the Capability Maturity Model Integration (CMMI) [5], designed for large organizations [6,7]. SC7-WG24 group has developed the ISO/IEC 29110 standard, a reference model focused on VSE [6,8]. O'Connor et al. [6] agree that VSE require specialized support in the definition of their processes and software lifecycle standards, under the premise that practices of software standards such as ISO/IEC 12207 and CMMI, are not suitable for these contexts, since applying lifecycle models in a VSE is a hard task and these models are normally excluded when Software Process Improvement (SPI) projects are scoped. However, organizations still requires effort for learning and adopting the ISO 29110 standard [9]. On the other hand, the Unified Process (UP) is described as a "generic framework that can be specialized for a variety of software systems for different application areas, different organizations types, different competence levels and different project sizes" [3]. UP is a reusable and adaptable software life cycle model, well known in the software industry and widely accepted by the academic community [10], and it is defined as a "comprehensible software development process framework emphasizing use-cases, architecture focus and an iterative approach" [11]. The wide use of the UP in the software industry and the academy was the key strategy in building VSE lifecycle model based on Unified Process (UP-VSE), a ready to use implementation of ISO/IEC 29110 standard for VSE based on the UP.

For the reasons mentioned above, this paper proposes an UP based implementation of ISO/IEC 29110 standard for UP-VSE. Rational Unified Process (RUP) is widely known and many organizations have tried to adopt it. The standard implementation related to the UP was carried out by assessing sub process elements in the corresponding UP workflows proposed by Jacobson et al. [3] and using RUP process templates with the Tutelkan project: a reusable lifecycle model to implement SPI on small environments [12].

UP-VSE was been applied as case study [13] in a research group from a regional university, in research projects where software development take place. The remainder of this document has been organized as follows: Sect. 2 introduces the related works, such as UP definition and structure, the ISO/IEC 29110 standard, the relationship between the UP and the ISO/IEC 29110 and applications in industry. Section 3 contains the UP-VSE process disciplines with its process assessment. Section 4 presents a study case: the PRODIGIA process showing its applicability and adoption. Section 5 describes the conclusions, limitations and further work.

2 Background and Related Work

2.1 The Unified Process

The UP life cycle consist of 4 phases (inception, elaboration, construction, and transition) with their related milestones which contain an indefinite number of iterations that follow their fundamental workflows (requirements, analysis, design, implementation and testing) [3]. The Unified Process UP is composed of a set of interconected process elements (work products, tasks, roles and activities) with a comprehensible order that is suitable to diverse project and team sizes in order to obtain a high quality software product [3]. UP is composed by phases, milestones and iterations that offer a view of the project progress. Each milestone is a project indicator and each one represents a limit among phases. Iterations offer an UP view and each one represents a valuable software increment for the stakeholder [3]. UP consists of 3 fundamental features: (i) it is *use case-driven*, "a project directed by use cases means that its progress is measured by completed use cases" [3]; (ii) it is *architecture-centered*, architecture gives a clear perspecive of software construction to the work team [3]; and (iii) it is *iterative and incremental*, since it splits the project into smaller sub-projects called iterations, which results in valuable product increments [3]. Hanssen et al. [14] conclude that UP is a process framework, and therefore, it requieres an adaptation effort to specific context by organizations. For that reason, "no process is universal, they cannot be applied in its integrity" [13]. Trying to address an implementation to cope with software project variations, the UP is designed for flexibility and extensibility. It allows a variety of lifecycle strategies and also the selection of the artifacts to be produced; it also defines activities to be performed, when and who are to perform those activities [13]. UP has been widely accepted by industry [14,15]. In fact, arround 50% of small software organizations have used UP as framework for organizational process elaboration [16,17]. There are magazines, software engineering conferences, books, articles, and other reported experiences that support the considerable UP usage in academy and industry [14].

2.2 The ISO/IEC 29110 Standard

In order to supporting VSE software process improvement, the ISO's SC7-WG24 working group has created the ISO 29110 standard. Laporte et al. [6] illustrate the initial development of ISO/IEC 29110 standard, considering aspects of ISO/IEC 12207 standard adapted to the VSE needs. The standard establishes a common framework based on VSE life cycle profiles oriented.

- **Overview (TR ISO 29110-1)**: contains the ISO 29110 vocabulary, business aspects, features and VSE requirements. This section of the standard clarifies the rationale for VSE specific profiles, documents, standards and guidelines, and introduces the basic process concepts, lifecycle and standardization, and the family of ISO 29110 documents.

- **Profiles (ISP)**: allow grouping references and/or parts of documents in order to meet the requirements and characteristics of the VSE, which include:
  - *Framework and taxonomy (ISP 29110-2)*
  - *Profile specifications: (29110-4-m, Profile group m.)*
- **Guides**: contain guidelines for implementation (specific domain) about performing the processes in order to achieve maturity levels (e.g., recommended activities, measures, techniques, templates, models, methods, etc.). There are 2 kinds of guides:
  - *Assessment guide (TR29110-3)*: describes the assessment process to determine the process capabilities and/or organizational maturity.
  - *Management and engineering guide (TR29110-5-m-n, Profile group m: Profile n.)*: describes mandatory aspects in software management and development to be considered for certification.

  This work will consider the engineering guide as a framework for this assessment. This section of the standard has 6 processes (*Software Implementation Initiation*, *Software Requirements Analysis*, *Software Architectural and Detailed Design*, *Software Construction*, *Software Integration and Test*, *Product Delivery*) used as benchmarks aiming at analyzing the UP.

2.3 ISO/IEC 29110 Impact

Rodríguez-Dapaena et al. [18] Ilustrate how to certify VSE's processes by using the ISO/IEC 29110 standard through an adequate incorporation of "best practices". It introduces a cost/benefit alternative approach for VSE's international recognition as quality software producers. This article is relevant since it involves elements that support the use and compatibility of this standard with traditional software development models, such as UP. Laporte et al. [19] Ilustrate a study of implementation of the standard ISO/IEC 29110 to stablish the new advanced profiles about a canadian industry. The results and decisions were documented for future implementations of these profiles with traditional models, such as CMMI. This work aims at implementing an adequate SPI in VSE in order to achieve competitive goals in parallel with the ISO standard goals accomplishment. Laporte et al. [20] Ilustrate several case studies of the implementation of ISO/IEC 29110. Due to its recent emergence, there are knowledge and implementation gaps of this standard in early studies. This work is significant since it illustrates a more adequate way to adopt the standard, taking advantage of the report of the implemented practices to the current work. O'Connor et al. illustrate the Deployment of ISO/ IEC 29110 standard Packages [9,21,22] to ease adoption. DPs have been tested on VSE pilot projects from different countries, such as Canada, France, Belgium and others. However, DPs do not have a formalized process lifecycle that can be used as a reference point or a specific case study in the VSE context [23]. The works mentioned above show the ISO/IEC 29110 impact on industry, and support the standard usage and relevance for this work.

2.4 Quality Standard Implementation Through Different UP Versions

Falbo et al. [24] implemented a UP role mapping to a small team applied to a case study in an organization assessed at Level 3 CMMI. The essential roles must meet at least one of three conditions mentioned in Monteiro et al.'s work [25]; then, they must be mapped to the RUP roles (39 in total), and in this case they were reduced to 13. Once the RUP roles are mapped, it is defined which ones are integrated and which ones are not, indicating -just in some cases- with an "X" in the mapping which roles should be restricted when assigned to the same person. Bryce et al. [26] attempted to form a complete workflow model for an ICT company in order to achieve the third CMMI maturity level by using different methodologies: The Open Group Architectural Framework (TOGAF) and the RUP. These methodologies have been integrated by using support tools and customization processes. These works illustrate the UP relevance and use in the software industry, focusing mainly on the achievement of high quality standards such as CMMI. Grau et al. [27] also developed a life cycle model (ILCM) oriented to a company (the Credit Suisse IT Switzerland) based on RUP, building the gap between this software methodology process and CMMI.

2.5 Cases Involving Unified Process in Small Organizations

Hanssen et al. [14] Involves the RUP adoption in a SME using the Case Study methodology [28]. This work proved that the use of RUP had some positive effects: interviews with RUP users show that there is a great need for better trainning and practical support in order to get most value out of RUP. Sánchez-Gordón et al. [29] illustrate about of the SPI in Small and Medium Enterprises (SME)'s, comprising this current work (UP-VSE) in the creation of prescriptive software process models from lifecycle model standards, and also techniques, frameworks, among others. This article is important since it involves the construction and importance of UP-VSE as part of a systematic revirew in the industry contexts and reports. Nowadays, there are reports about the RUP implementation on industry [14,15]. Therefore, RUP is a widely used lifecycle model that has a long support validity evidenced in reports' dates that are continuously released, and highlights this lifecycle relevance in software industry [14].

2.6 Process Assessments Involving UP

Some efforts for implementing UP and the international ISO standards have been carried out. Rational Corporation assessed RUP against the assessment model and the process performance indicator guidance ISO/IEC 15504-5 which are used to determine capability for each RUP assessed subprocesses. ISO/IEC 15504-5 specifies a framework for assessing the software process that elaborates the mandatory process and defines associated base practices, work products, and management practices. The results reflect the engineering focus of the RUP.

The RUP needs augmentation mainly in the management and organization areas and the traditional quality function to meet the requirements of 15504 at higher ratings [30]. Reinehr et al. [31] carried out an ISO/IEC 12207 implementation using RUP. Those lifecycle models differ from the focus type. The software process model is mainly focused on the project level, whereas the standard is focused on the organizational level. Results concluded that the standard was partially covered by RUP due to punctual tasks that this software model lifecycle covers indirectly throughout concepts, guidelines and checklists. In conclusion, RUP can be used to implement ISO/IEC 12207, but, in order to get a better coverage of the standard, it is necessary to expand the RUP.

2.7 UP/ISO/IEC 29110 Assessments and Their Approaches to the Software Industry

Galvan et al. [32] carried out an ISO/IEC 29110 Project Management Process evaluation among different software methodologies (XP [33], SCRUM [34] and Unified Process for EDUcation (UPEDU) [35]). After contrasting these methodologies and the ISO standard throughout task, roles, and the work products, the assessment accomplishment level shows that SCRUM and UPEDU present a high level of compliance, whereas XP has a moderate level in a 1 to 3 assessment value range. This work shows a need to find a relationship between the standard ISO/IE 29110 and the UP. The work mentioned above supports this idea.

3 UP-VSE Process

3.1 The UP-VSE Approach

UP-VSE is one of the versions of UP family which differs from others, because it is VSE-oriented and its need for standardization, therefore, is based on the ISO/IEC 29110-5-1-1 standard. UP-VSE is an UP-based process model considering ISO/IEC 29110-5-1-1:2012 [36] requirements. UP-VSE keeps the same milestones and phases of the Jacobson et al's. UP [3].

UP-VSE consists of 9 subprocesses, as follows: (i) Initialize the project, (ii) Planning and Management by Iteration. (iii) Requirements Engineering. (iv) Design. (v) Develop Software Increment. (vi) Integrate developed components to a new version of the solution. (vii) Integration Test. (viii) Document Software Manuals. (ix) Deploy current version.

3.2 Reengineering, Updating and Including Disciplines, Activities and Subprocesses from ISO/IEC 29110-5-1-1

Some specific considerations were kept in mind when constructing UP-VSE disciplines:

- There were components of both, the standard and the UP, that are not fully covered because some of the UP components can achieve equal or higher level of component detail that standard required to be carry out the standard.
- Whereas some elements of the standard process mentioned above could not be properly categorize in the disciplines, some information sources from other UP family members (such as the Agile Unified Process (AUP), RUP, UP [3] and UPEDU process web page), were considered. After this research process about these process elements, it was concluded that:
  - Since this proposal only deals with technical aspects, the UP [3] project management discipline was integrated to this proposal because some of the tasks found in the standard assignment did classify in this discipline.
  - The deployment discipline was included based on the effort distribution of the following aspects:
    - the relationship between the number of UP-VSE artifacts constituting the user manuals and this discipline; and
    - the plan to deploy software, which enables the transition to the physical site defined by the customer performing functional testing and integration of software components.
  - The environment discipline wAssociated measures to PRODIGIAas included, since to ISO/IEC 29110-5-1-1 standard organizes tasks of this discipline in their subprocesses, considered by UP as part of project management.
  - The configuration management discipline was included, because the activities of the version management and control and the documentation are part of this discipline in the UP.
  - The UP disciplines were updated with some of the process elements of the standard, preserving the practices and the philosophy of the UP process model.

3.3 Modifications and Omitted Elements from I. Jacobson et al. Unified Process [3] in UP-VSE Creation

- **Software requirements:** UP-VSE omits the glossary and business model as inputs for software development, but a business expert (client) is necessary. The user interface prototyping in UP-VSE is recomended in a more lighter way; User interface drafts are made in paper and pencil with key stakeholders in the final model of the system, or otherwise prepared from the captured requirements and documented for further verification and validation. The activity “structure the use case model” (Sect. 7.4.5) by Jacobson [3] is considered a step in the execution of the use case model in the activity “develop the use case diagram” (task **1.2.2.3** in “Identify Requirements” UP-VSE subprocess).

- **Analysis and Design:** Analysis and design have been considered a single discipline, because there are activities between these two disciplines that are very similar. UP-VSE has been summarized in a few tasks and formulated in general to consider any type of modeling UML that needs to be done by a particular VSE; a generic process that can create, edit and maintain any model defined for any project, considering the abstraction of the relationship model between use case realizations and its artifacts. It is considered to use design models to start the system components coding in a little more casual way. The order of priority of these components to be developed is considered a step in charge of the corresponding roles.
- **Implementation:** Implementation model has been omitted. The deployment model (taken as the deployment plan) has been delegated to run on the latest activities of UP-VSE process in the transition phase. The construction and testing of components and subsystems has been delegated separately, so the configuration discipline can be executed in parallel in order to maintain the iterative and incremental principles from UP families.
- **Test:** Test model was changed to the test plan in order to concentrate all protocols to follow. Development teams define who execute these tests and how. The tests focus on three main components: component testing, system or structure, and deployment. The results of each set of tests are documented in the corresponding test or deployment plan (in case these fields are not found in the template, they will be attached to the document).
- **Management:** Traceability with requirements and previous artifacts is in charge of the role associated with each artifact in a backtracking way. UP-VSE omitted some details about the monitoring by the project management, except for assigning roles to members of the project work team, because it is an activity defined in ISO/IEC 29110-5-1-1, and other aspects such as the following: UP-VSE scope considers the technical section, so management is not a heavily documented process. Anyway, management sections have been documented in relation to the basic prerequisites for starting and managing software development as the project plan, the iteration plan and the management of all project iterations. Each requirement test artifact and previous artifact traceability is in charge of the role associated to each artifact in a backtracking way.

3.4 Role Equivalences Between UP-VSE and ISO/IEC 29110-5-1-1:2012

Table 1 shows the role equivalence between UP-VSE and ISO/IEC 29110-5-1-1.

Table 1. Role equivalence between UP-VSE and ISO/IEC 29110-5-1-1

ISO/IEC 29110-5-1-1:2012		UP-VSE
Customer	CUS	Client
Project Manager	PM	Project Chief
Work Team	WT	System Analyst
		Analyst of the System
		Test Analyst
		Software Architect
		User Interface Designer
		Test Designer
		Tools Specialist
		Tests Specialist
		Requirements Specialist
		Software Performer
		Intergator
		Change Control Chief
		Technical Inspector

4 PRODIGIA: A Case Sudy

Runeson et al. [28] work was used as the research methodology for software engineering. Case studies are carried out in real world, so they have a high degree of realism. The data obtained in the study should be consistent. Validity of the case study depends largely on the design, execution, and analysis of results. The case study scope was the assessment of the VSE subprocesses in relation to the ISO/IEC 29110-5-1-1:2012 standard in a real context. To determine the subset of discipline elements from UP process (tasks, roles and work products) applyable to a VSE context through an empirical application of this lifecycle model in a small entity. This case study aims at answering two research question:

- What UP process elements (tasks, roles and work products) are applicable to a VSE?
- How much does this set of process elements satisfy the ISO/IEC 29110-5-1 -1:2012 subprocesses?

The VSE entity, which in this context has been assumed to be member of PRODIGIA process, is a research group in software development for surgical simulations. The development group was consisted initially by three M. Sc. students, one undergraduate student and two professors who were mentors of projects. The selected case study was "Design and construction of a virtual environment for surgical simulation using haptic interfaces". In this case, the group, —with our support—, defined its own process from the UP literature specification. Product development was carried out in different platforms, by using Virtual Matlab

VTK and QT through Visual Studio programming environment as a technological support. They also used 3D textured prototyped robotic arms from other development projects carried out previously, in order to make internal incisions to human body organs without making epidermis large cuts.

Table 2. Associated measures to PRODIGIA case study.

Indicator	Measurement	Information sources	Tools
Subset of elements of the process with the highest level of acceptance and quality	**PAL** - Process acceptation level from development work team **QL** - Quality level in work products application	Software end users, surgical system developers and managers	Interviews, surveys, project repository (documentation management tool), AVISPA [37] errors report
UP accomplishment degree in relation to ISO/IEC 29110-5-1-1:2012 standard	$\mathbf{ID_{ISO}}$ - Implementation degree of a ISO/IEC 29110-5-1-1 practice in project	Software end users, Developers and managers of the surgical system, The software product	PRODIGIA process model. Project repository. Assessment protocol
Process adoption	**PAD** - Process accomplishment degree from development work team	Process verification	Minutes of meeting has been processed and project repository (documentation management tool) has been consulted
Applicability	**E** - Effort	Process Group Monitoring Artifacts	Minutes of meeting are processed and project repository is consulted

Table 2, shows the defined indicators and collection tools that had been used in the case study according to the research design.

Case Study Execution

The team scheduled and executed a set of weekly meetings to discuss the process and obtain information about relevant software aspects to the process implementation. The main effort investment was the requirements engineering software process. Once the definition of requirements engineering software process in the

AI group was implemented, formalizing PRODIGIA was an easier task because the work group members understood the dynamics.

PRODIGIA Artifacts: Artifacts were downloaded from the RUP free templates web page, except for some PRODIGIA templates (such as the work plan which matches the project plan in the ISO/IEC 29110-5-1-1 standard). The work plan template illustrates the minimal sections to academic research projects in the University of Cauca, and differs from the project plan in its content, mainly in the definition of the roles which are distributed only among students and faculty.

PRODIGIA Roles: Because case study was executed in an academic environment, PRODIGIA has the following roles:

- **Student:** This role develops all and each of the Unified Process artifact needed in PRODIGIA.
- **Thesis mentor:** This role is the academic project manager. It validates and verifies many of the deliverables achieved, and guides the artifact construction and is a source of requirement information.
- **Stakeholder (person/organization that is interested in other organization progress) of the case study:** This is the main source of organizational requirements.

Table 3 illustrates in detail the role equivalence between PRODIGIA and ISO/IEC 29110-5-1-1.

Table 3. Role equivalence between PRODIGIA and ISO/IEC 29110-5-1-1

ISO/IEC 29110-5-1-1:2012		PRODIGIA
Customer	CUS	*Case Study Stakeholder*
		Thesis mentor
Project Manager	PM	
Work Team	WT	Student

PRODIGIA Task Assessment: The same process assessment was executed. PRODIGIA was assessed in relation to the technical standard section.

The process score was calculated by using Quantitative levels (Q) that together offer a Coverage level (C), as follows:

$$Q_T[Tasks] = 0.\overline{6}7, C_T(Q_T[Tasks]) = P = \text{Partially achieved.}$$

Table 4. PRODIGIA artifact assessment in relation to ISO/IEC 29110-5-1-1:2012 standard

Artifacts			
ISO/IEC 29110-5-1-1	PRODIGIA	C_W	Q_W
Project Plan	Draft Grade	**P**	0.3
Software Requirements Specification Document	Monograph	**F**	1
Software Components Identification	Monograph	**F**	1
Test Cases and Test Procedures	Monograph	**P**	0.3
Test Report	Monograph	**P**	0.3

PRODIGIA Artifact Assessment: Table 4 illustrates assessment executed to each artifact.

As same as the last formula, it was applied to obtain the artifact score in relation to ISO/IEC 29110-5-1-1:

$$Q_W[Work\_Products] = 0.65, C_W(Q_W[Work\_Products]) = P = \text{Partially achieved.}$$

4.1 Results

About Measurement Framework:

- **Process acceptation level from development work team (PAL):** Surveys using a O to 5 scale were answered by 2 students; results were 3.25 and 3.75 for students, while the mentor scored 4.67. In average, score was 3.89 out of 5. Therefore, PRODIGIA process acceptance was 77.8% for process users.
- **Quality level in work product application (QL):** Some work products were not fully applied because PRODIGIA process members had widely unstable requirements and did not had any budget. Therefore, quality level was calculated in 62.25%.
- **UP accomplishment degree in relation to ISO/IEC 29110-5-1-1:2012 standard ($\mathbf{ID_{ISO}}$):** It has been noticed that PRODIGIA lacks specific verification methods, mainly in requirements, but after regular meetings to monitor progress of the project, the ellaborated artifacts are verified indirectly. PRODIGIA is 69.09% $\mathrm{ID_{ISO}}$. This means, that PRODIGIA is 69.09% applicable to VSE industry based on ISO/IEC 29110-3 standard to assessing VSE processes.
- **Process accomplishment degree from development work team (PAD):** This is similar in QL measurement. About process application, design was not applied as well as PRODIGIA suggests. The PRODIGIA project team did not apply testing and coding good practices because they did not have enough knowledge about UP process, its philosophy and practices. Therefore, quality was about 44.03%.

– **Effort (E):** To complete all PRODIGIA subprocesses, with templates definition, Eclipse Process Framework (EPF) documentation and control meetings of the process execution, process socialization with the thesis mentors and their students, took 119 hour/person.

Analysis of Case Results: PRODIGIA subprocesses widely cover the critical aspects of UP software development, but, all PRODIGIA development practices are less robust than UP; Therefore, PRODIGIA has a more limited process scope. It was also observed a better understanding of this lifecycle model against UP in previous experiences, such as previous presentations and conferences about how to use it to develop academic projects by VSE. Due to its academic orientations in some process elements. In a previous research, the work required more effort (**E**) than estimated, due to the PRODIGIA process members' lack of knowledge about the software development oriented process. Defining the rest of process was a faster task, because each participant of PRODIGIA had more knowledge about its process role and, therefore, the implementation and usage of this lifecycle. This work estimated that this software process is feasible to be implemented in specific contexts, mainly in a software development academic contexts.

In this work the **PAD** was the lowest measure that highlighted:

– The process was not followed as expected. Based on this aspect, this mean that
– it is still difficult to create good practices and a keep process awareness throughout the project and
– software development was performed without an appropriate documentation.

However, surveys to the process users (**PAL**) showed a high acceptance. This evidenced that:

– The process made the team to be aware oft the importance of the process to software development and
– people realized that this was the first step to make an adequate construction in order to achieve the developers' commitment to the software documentation and development good practices with an acceptable quality level in work product application (**QL**).

5 Conclusions, Limitations and Future Works

This work presents a UP-based methodological framework to develop software artifacts in a VSE whose methods are well known to industry and widely used by academic environments. The framework uses the Jacobson et al.'s [3] UP and the ISO/IEC 29110-5-1-1:2012 basic profile. UP-VSE assessment through PRODIGIA allowed to visualize some gaps about artifacts and the standard completion. Some improvements were the task number decreasing without decreasing or affecting product quality and the standard scope. PRODIGIA, as a software development process approach can not perform an acceptable level of coverage of the ISO/IEC 29110-5-1-1 profile ($\mathbf{ID_{ISO}}$) due to the specific scope managed in this case study that addressed software development process to accomplish more academic issues than software process ones. It is necessary to instantiate the UP-VSE approach to an industry development environment with the aim of achieving closer ISO/IEC 29110-5-1-1:2012 objectives and getting high rate in $\mathbf{ID_{ISO}}$ measurement. The case study illustrated that PRODIGIA process users are more acquainted to hardware than software process models, that is why they put more effort in coding and in developing applications without centering in using good practices. In order to achieve ISO/IEC 29110-5-1-1 standard, a need for implementing configuration control and environment management in UP-VSE was detected. UP-VSE needs to be improved in all the contexts where VSE's are involved. Likewise, it is necessary to develop more versions of this lifecycle model to reach intermediate and advance profiles that are being developed by ISO. There is also a need to execute more detailed assessments in order to find posible gaps and improve and simplify the process. It is necessary to improve the UP-VSE approach in all contexts where VSE's are involved, and develop more versions of this lifecycle model in order to achieve intermediate and advance profiles that are being developing by ISO. More detailed assessments need to be executed in order to find more gaps and achieve process simplification and improvement.

References

1. Humphrey, W.S.: Managing the Software Process. SEI Series in Software Engineering. Addison-Wesley, Boston (1989)
2. Cockburn, A.: Selecting a project's methodology. IEEE Softw. **17**(4), 64–71 (2000)
3. Jacobson, I., Booch, G., Rumbaugh, J., Rumbaugh, J., Booch, G.: The Unified Software Development Process, vol. 1. Addison-Wesley, Reading (1999)
4. Joint Technical Committee ISO/IEC JTC 1, Information Technology, Subcommittee SC 7. ISO/IEC 12207: systems and software engineering - software life cycle processes. International Organization for Standardization (ISO), Genève, Switzerland, 2nd edn., February 2008
5. Kneuper, R.: CMMI: Improving Software and Systems Development Processes Using Capability Maturity Model Integration. Rocky Nook, San Rafael (2008)
6. Laporte, C.Y., Alexandre, S., O'Connor, R.V.: A software engineering lifecycle standard for very small enterprises. In: O'Connor, R.V., Baddoo, N., Smolander, K., Messnarz, R. (eds.) EuroSPI 2008. CCIS, vol. 16, pp. 129–141. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85936-9_12
7. O'Connor, R.V., Laporte, C.Y.: Using ISO/IEC 29110 to harness process improvement in very small entities. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 225–235. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22206-1_20
8. Laporte, C.Y., April, A., Renault, A.: Applying international software engineering standards in very small enterprises. Crosstalk **20**(2), 29–30 (2007)
9. Ribaud, V., Saliou, P., O'Connor, R.V., Laporte, C.Y.: Software engineering support activities for very small entities. In: Riel, A., O'Connor, R., Tichkiewitch, S., Messnarz, R. (eds.) EuroSPI 2010. CCIS, vol. 99, pp. 165–176. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15666-3_15

10. Halling, M., Zuser, W., Kohle, M., Biffl, S.: Teaching the unified process to undergraduate students. In: 2002 Proceedings of the 15th Conference onSoftware Engineering Education and Training, (CSEE&T 2002), pp. 148–159. IEEE (2002)
11. Hanssen, G.K., Westerheim, H., Bjørnson, F.O.: Using rational unified process in an SME – a case study. In: Richardson, I., Abrahamsson, P., Messnarz, R. (eds.) EuroSPI 2005. LNCS, vol. 3792, pp. 142–150. Springer, Heidelberg (2005). https://doi.org/10.1007/11586012_14
12. Valdés, G., Visconti, M., Astudillo, H.: The tutelkan reference process: a reusable process model for enabling SPI in small settings. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 179–190. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22206-1_16
13. Hunt, J.: An introduction to the UML and the unified process. In: Hunt, J. (ed.) Guide to the Unified Process featuring UML, Java and Design Patterns. SPC, pp. 21–37. Springer, London (2003). https://doi.org/10.1007/1-85233-856-3_3
14. Hanssen, G.K., Westerheim, H., Bjørnson, F.O.: Tailoring RUP to a defined project type: a case study. In: Bomarius, F., Komi-Sirviö, S. (eds.) PROFES 2005. LNCS, vol. 3547, pp. 314–327. Springer, Heidelberg (2005). https://doi.org/10.1007/11497455_26
15. Hanssen, G.K., Bjørnson, F.O., Westerheim, H.: Tailoring and introduction of the rational unified process. In: Abrahamsson, P., Baddoo, N., Margaria, T., Messnarz, R. (eds.) EuroSPI 2007. LNCS, vol. 4764, pp. 7–18. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75381-0_2
16. Hurtado, J.A., Pino, F., Vidal, J.: Sistema integral para el mejoramiento de los procesos de desarrollo de software en colombia (SIMEP-SW). Departamento de Sistemas, Universidad del Cauca. Popayán, pp. 1–31 (2003)
17. Camacho, M.C., Alegría, J.A.H.: Analizing the viability for adopting the software process line approach in small entities. 2012 7th Colombian Computing Congress (CCC), pp. 1–6 (2012)
18. Rodríguez-Dapena, P., Buitrago-Botero, M.F.: How to certify the very small entity software processes using ISO/IEC 29110. In: Rout, T., O'Connor, R.V., Dorling, A. (eds.) SPICE 2015. CCIS, vol. 526, pp. 245–252. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19860-6_20
19. Laporte, C.Y., Tremblay, N., Menaceur, J., Poliquin, D.: Implementing the new ISO/IEC 29110 systems engineering process standard in a small public transportation company. In: Kreiner, C., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) EuroSPI 2016. CCIS, vol. 633, pp. 15–29. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44817-6_2
20. Laporte, C.Y., O'Connor, R.V.: A multi-case study analysis of software process improvement in very small companies using ISO/IEC 29110. In: Kreiner, C., O'Connor, R.V., Poth, A., Messnarz, R. (eds.) EuroSPI 2016. CCIS, vol. 633, pp. 30–44. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44817-6_3
21. O'Connor, R.V., Laporte, C.Y.: Software project management in very small entities with ISO/IEC 29110. In: Winkler, D., O'Connor, R.V., Messnarz, R. (eds.) EuroSPI 2012. CCIS, vol. 301, pp. 330–341. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31199-4_29
22. O'Connor, R.V., Laporte, C.Y.: Deploying lifecycle profiles for very small entities: an early stage industry view. In: O'Connor, R.V., Rout, T., McCaffery, F., Dorling, A. (eds.) SPICE 2011. CCIS, vol. 155, pp. 227–230. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21233-8_27

23. Laporte, C.Y., O'Connor, R.V., Paucar, L.H.G.: The implementation of ISO/IEC 29110 software engineering standards and guides in very small entities. In: Maciaszek, L.A., Filipe, J. (eds.) ENASE 2015. CCIS, vol. 599, pp. 162–179. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30243-0_9
24. Falbo, R.A., Borges, L.S.M., Valente, F.F.R., Using knowledge management to improve software process performance in a CMM level 3 organization. In: 2004 Proceedings of the Fourth International Conference on Quality Software, QSIC 2004, pp. 162–169 (2004)
25. Borges, P., Monteiro, P., Machado, R.J.: Mapping RUP roles to small software development teams. In: Biffl, S., Winkler, D., Bergsmann, J. (eds.) SWQD 2012. LNBIP, vol. 94, pp. 59–70. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27213-4_5
26. Day, B., Ke-Zun, S.C., Lovelock, L., Lutteroth, C., Climbing the ladder: CMMI level 3. In: 2009 IEEE International Enterprise Distributed Object Computing Conference, EDOC 2009, pp. 97–106 (2009)
27. Sägesser, K., Joseph, B., Grau, R.: Introducing an iterative life-cycle model at credit Suisse IT Switzerland. IEEE Softw. **30**(2), 68–73 (2013)
28. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Softw. Engg. **14**(2), 131–164 (2009)
29. Sánchez-Gordón, M.-L., Colomo-Palacios, R., de Amescua Seco, A., O'Connor, R.V.: The route to software process improvement in small- and medium-sized enterprises. In: Kuhrmann, M., Münch, J., Richardson, I., Rausch, A., Zhang, H. (eds.) Managing Software Process Evolution: Traditional, Agile and Beyond – How to Handle Process Change, pp. 109–136. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31545-4_7
30. Rational Corporation Whitepaper: Assessing the rational unified process against ISO/IEC15504-5: information technology - software process assessment part 5: an assessment model and indicator guidance. Rational Software Corporation, United States (2000)
31. Reinehr, S.S., Balduino, R., Machado, C., de Paula Pessôa, M.S.: Implementing ISO/IEC 12207 standard using rational unified process. In: Software Engineering Research and Practice, pp. 667–680 (2003)
32. Cruz-Cunha, M.M., et al.: Conference on ENTERprise information systems/international conference on project management/conference on health and social care information systems and technologies, CENTERIS/ProjMAN/HCist 2015 7–9 October 2015 a compliance analysis of agile methodologies with the ISO/IEC 29110 project management process. Procedia Comput. Sci. **64**, 188–195 (2015)
33. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley Longman Publishing Co. Inc., Boston (2000)
34. Sutherland, J., Schwaber, K.: The scrum guide. The definitive guide to scrum: the rules of the game. Scrum.org, October 2013
35. Robillard, P.N., Kruchten, P., d'Astous, P.: Software Engineering Using the UPEDU. Addison-Wesley Longman Publishing Co., Inc., Boston (2002)
36. Joint Technical Committee ISO/IEC JTC 1, Information Technology, Subcommittee SC 7. ISO/IEC 29110-5-1-1: software engineering - lifecycle profiles for very small entities (VSEs) - Part 5-1-1: management and engineering guide: generic profile group: entry profile. International Organization for Standardization (ISO), Genève, Switzerland, 1st edn., September 2012
37. Hurtado Alegría, J.A., Bastarrica, M.C., Bergel, A.: AVISPA: a tool for analyzing software process models. J. Soft.: Evol. Process **26**(4), 434–450 (2014)