ISO/IEC 29110: Current overview of the standard

Euclides J. Moreno Campos, Mary-Luz Sanchez-Gordón, and Ricardo Colomo-Palacios

Departamento de Informática

Universidad Carlos III de Madrid

Madrid - España

100289251@alumnos.uc3m.es, mary_sanchezg@hotmail.com, rcolomo@inf.uc3m.es

***Abstract**: The software industry recognizes the value of VSEs in contributing valuable products and services. Unfortunately current ISO/IEC standards do not completely address the needs of VSEs. Due to this, the ISO/IEC 29110 standard has been developed. While reading this paper, one can find previous attempts of approaching ISO and ISO/IEC standards such as ISO/IEC 12207, ISO/IEC 15289, ISO/IEC 15504 and ISO 9001 to VSEs, as well as models inspired on ISO and ISO/IEC standards and Maturity models like CMMI and oriented to VSEs such as MoProSoft and projects like COMPETISOFT. A summary of part 5 of the standard, making special focus on the Entry Profile, belonging to the generic profile group as well as an initial implementation of the standard in VSEs both through the creation of a customized approach and the utilization of the available deployment packages are going to be depicted. Finally a short comment about the importance of Deployment Packages and the Network Support Centers in adopting and raising awareness about the standard, as well as future work to be done on the standard, for example, the completion and creation of new profiles is going to be found.*

***Keywords**: VSEs, Very Small Entities, ISO/IEC 29110, Lifecycle profiles, Software Engineering Standards, Software Processes.*

1. Introduction

According to the OECD (Organization for Economic Co-operation and Development) SME (Small and Medium Enterprises) and Entrepreneurship Outlook report (2005) “SMEs constitute the dominant form of business organization in all countries world-wide, accounting for over 95 % and up to 99 % of the business population depending on country” [1]. Hence, a set of studies cited in [2] show the same behavior in technology-based companies. As depicted in Table 1, in Europe, 85% of the Information Technology (IT) sector's companies have between 1 and 10 employees. In Canada, the Montreal area was surveyed, and it was found that close to 80% of IT companies have fewer than 25 employees. Another study conducted by the Technology Assessment Group (CITA) of Wallonia has published similar data, which reveal that about 60% of IT companies there have fewer than 5 employees. In Brazil, small IT companies represent about 70% of the total number of companies. In Northern Ireland, a survey reports that 66% of IT organizations within companies employ fewer than 20 employees [2]. Finally, a study performed in 2004 by Industria Mexicana de Software demonstrated that 92% of the IT companies in Mexico are small and medium-sized (with less than 100 people) [3]. Thus, certain Very Small Entities (VSEs[1]) provide software components that are being assembled in larger software companies in order to generate critical and intensive software configurations [4,5], so one must conclude that software industry recognizes the value of VSEs in contributing valuable products and services [1,2].

[1] The terms "very small entity” and “very small entities" (VSE/VSEs) have been defined by the ISO/IEC JTC1/SC7 Working Group 24 (WG24) as being "an entity (enterprise, organization, department or project) having up to 25 people" and have subsequently been adopted for the use in the ISO/IEC 29110 standard. *V. Ribaud, P. Saliou, R. V. O’Connor, and C. Y. Laporte, “Software engineering support activities for very small entities,” in Systems, Software and Services Process Improvement, Grenoble (France), 2010, pp. 165–176.*

Size (employees)	IT Companies (%)	Country/Region
<= 10	85	Europe
< 25	80	Montreal (Canada)
< 5	60	Wallonia (Belgium)
(Small IT companies)	70	Brazil
< 20	66	Northern Ireland
< 100	92	Mexico

Table 1. Presence of VSEs in general level IT companies by country/region

According to [1], from studies and surveys conducted, it is clear that the majority of International Standards do not address the needs of VSEs. Conformance with these standards is difficult, if not impossible. Subsequently VSEs have no, or very limited, ways to be recognized as entities that produce quality software in their domain. Therefore, VSEs are often cut off from some economic activities. It has been found that VSEs find it difficult to relate International Standards to their business needs and to justify the application of the standards to their business practices. Most VSEs can neither afford the resources, in terms of number of employees, budget and time, nor do they see a net benefit in establishing software life cycle processes. To rectify some of these difficulties, a set of guides has been developed according to a set of VSE characteristics. The guides are based on subsets of appropriate standards elements, referred to as VSE Profiles (ISO/IEC 12207[2], ISO/IEC 15289[3], ISO/IEC 15504[4], ISO 9001[5]). The so-called guides are gathered into the standard *ISO/IEC 29110 Software engineering — Lifecycle profiles for Very Small Entities*, which describes processes for project management and software implementation [6] and pretends to facilitate access to, and utilization, ISO software engineering standards in VSEs [2].

The section two details a critical analysis of previous attempts of approaching ISO/IEC standards and other methodologies and models such as MoProSoft[6] to Very Small Entities. Section three gives an overview of the standard, making special focus on part five, highlighting the Entry Profile, belonging to the generic profile group. Section four describes an initial implementation of the standard both through the creation of a customized approach and the utilization of the available deployment packages in small technology-based firms. Conclusions are discussed in section five.

2. Previous and related work

2.1. Previous and related standards

The International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) have been working together in order to design standards related to software engineering issues. Products of these efforts are:

[2] International Standard ISO/IEC 12207 Software Life Cycle Processes.
[3] International Standard ISO/IEC 15289 Systems and software engineering — Content of life-cycle information products (documentation).
[4] International Standard ISO/IEC 15504 Software Process Improvement Capability Determination.
[5] International Standard ISO 9001 Quality management systems — Requirements.
[6] Modelo de Procesos para la Industria de Software (Processes Model for the Software Industry).

ISO/IEC 12207 Software Life Cycle Processes

According to [7], this standard establishes a top-level architecture of the life cycle of software. The life cycle begins with an idea or a need that can be satisfied wholly or partly by software and ends with the retirement of the software. The architecture is built with a set of processes and interrelationships among these processes. The derivation of the processes is based upon two basic principles: *modularity and responsibility*.

- Modularity: The processes are modular; that is, they are maximally cohesive and minimally coupled to the practical extent feasible. An individual process is dedicated to a unique function.
- Responsibility: A process is considered to be the responsibility of a party in the software life cycle. In other words, each party has certain responsibilities. Responsibility is one of the key principles of total quality management.

ISO/IEC 15289 Systems and software engineering — Content of life-cycle information products (documentation)

The purpose of this International Standard is to provide requirements for identifying and planning the specific information items (information products) to be developed and revised during systems and software life cycles and service processes. The standard specifies the purpose and content of all identified systems and software life-cycle information items, as well as information items for information technology service management. The information item contents are defined according to generic document types and the specific purpose of the document. Information items may be combined or subdivided as needed for project or organizational purposes [8].

ISO/IEC 15504 Software Process Improvement Capability Determination

This standard defines a reference model for software process assessment, and a set of requirements on assessment models and methods [9]. It is intended to harmonize the many different approaches to software process assessment. It has nine parts. The reference model in Part 2 documents the set of universal software engineering processes that are fundamental to good software engineering and that cover best practice activities. It describes processes that an organization may perform to acquire, supply, develop, operate, evolve and support software and the process attributes that characterize the capability of those processes. The purpose of the reference model is to provide a common basis for different models and methods for software process assessment, ensuring that results of assessments can be reported in a common context [10].

ISO 9001 Quality Management Systems - Requirements

This international standard promotes the adoption of a process-based approach during the development, implementation and efficiency enhancements of a Quality Management System, in order to increase the customer satisfaction through the fulfillment of his or her requirements. ISO 9001 specifies the requirements for a Quality Management System, requirements that can be used for its application by organizations, with certification or contractual purposes. It focuses in Quality Management System efficiency to satisfy customer's requirements [11].

Unfortunately, none of these standards completely address the needs, goals and objectives of small and medium software enterprises. These standards were intended to be performed in larger organizations, in which the software development processes differ from the ones performed in VSEs. Nevertheless, there

have been some attempts to approach some of these standards to the small and medium software firms. Some of these attempts have taken place in Latin America; one of these efforts is called MoProSoft.

2.2. The MoProSoft model

According to [3], the high cost of SW-CMM[7] and CMMI[8] adoption in small enterprises and the need for a national standard were the basic reasons to develop a new software process model for the Mexican software industry. MoProSoft is a process model for small enterprises, built on the well-known practices of SW-CMM, ISO 9000:2000, PMBoK (Project Management Body of Knowledge) and others, and offers a new process structure, some new process documentation elements, a more precise process relationship, and an explicit process improvement mechanism. The model is complemented with the process assessment method EvalProSoft, which is based on the recommendations of ISO/IEC 15504 Part 2. The process model and the assessment method were applied to four small enterprises that had a typical Mexican software industry company profile. After experimentation (the application of MoProSoft and EvalProSoft in the four small software companies), authors evidenced the fulfillment of the following criteria [3]:

- C1. Proper for small and medium-sized enterprises (SME) with low maturity levels.
- C5. Defined as a set of processes based on internationally recognized practices.

Due to the results of this experiment the Mexican Secretary of Economy decided to formally make MoProSoft and EvalProSoft a Mexican standard [3]. Moreover, the Peruvian National Institute for the Defense of Competition and Protection of Intellectual Property (INDECOPI) developed a technical standard based on MoProSoft called *"Software Engineering: Software Development and Maintenance Process and Evaluation Models NTP 291.100:2009"* which was published in 2009. It is also important to highlight that MoProSoft was essential in the development of the ISO/IEC 29110 standard [14].

2.3. The COMPETISOFT project

Another Latin-American initiative is COMPETISOFT. The COMPETISOFT project, according to [15], is based on ISO/IEC 12207, ISO/IEC 15504, CMMI, MANTEMA[9], Métrica v3[10], Agile SPI[11], and mainly MoProSoft and EvalProSoft. This processes model is aimed to companies or internal areas of companies, dedicated to software maintenance and development. This model was developed and enhanced by persons that own wide knowledge in international model contents, as well as expertise on implanting such models in VSEs. This model consists of three categories, which cluster processes according to the typical organization structure:

[7] The Capability Maturity Model for Software (SW-CMM) describes the principles and practices underlying software process maturity and is intended to help software organizations improve the maturity of their software processes in terms of an evolutionary path from chaotic processes to mature, disciplined software processes [12].

[8] CMMI® (Capability Maturity Model® Integration) is a process improvement maturity model for the development of products and services. It consists of best practices that address development and maintenance activities that cover the product lifecycle from conception through delivery and maintenance [13].

[9] MANTEMA is a methodology for managing the Software Maintenance Process. This methodology defines precisely and rigorously all the activities and tasks, which must be executed during the Maintenance process, and explicitly considers the integration of the necessary activities for establishing and completing outsourcing relationships between customer and supplier organizations.

[10] Métrica is a methodological environment developed by the Spanish Ministry of Public Administration. In the latest version of Métrica v3, the object-oriented paradigm is included as a development option, and proposes the use of UML to model different aspects in the software lifecycle. Métrica v3 is the reference frame for software development in Spanish public entities [16].

[11] Agile SPI is a framework intended to support process improvement for the software industry. Its main goal is to motivate small and medium size companies towards improving and certifying their development processes [17].

- Corporate level management: Establishes the rationale of the organization, what the organization is willing to achieve and the respective strategies to make it possible.
- Tactical level management: Establishes action plans to implement strategies in relation to projects, processes and resources. It monitors Operation category and gives feedback to corporative management category.
- Operational level management: Performs maintenance and software development projects that cover customer's needs in terms of time and expected costs and report results to the tactical management category.

According to [14], the COMPETISOFT project is a Process improvement to enhance the competitiveness of small and medium organizations in Latin America. COMPETISOFT defined three objectives:

- Create a common methodological framework in Latin America.
- Spread the process culture into the researchers, academics and students communities.
- Influence in the standardization and certification entities, in order to establish a common and mutually recognized mechanism.

2.4. The 15504MPE project

Another effort in approaching ISO/IEC software engineering standards are depicted in [18], which describes some experiences gained from applying ISO/IEC 15504 for software process assessments focusing on process improvement in four small software companies in Brazil. The assessments has been performed in the context of a project called 15504MPE, which aims at the development of a customized assessment method based on the standard ISO/IEC 15504 adapted to small Brazilian software companies. After experimentation, the authors evidenced important benefits:

- Better understanding of the assessed processes based on the assessment results.
- Strengths and weaknesses of the assessed processes were identified in relation with the process assessment model.
- Suggestions for improvement with relevant impact on the software process were formulated and started to be implemented.
- Increased motivation for improvement due to a better understanding of the actual process and the identified weaknesses.
- Increased commitment to improve process quality.

To shorten the gap between *small and medium enterprises* and their compliance with ISO/IEC software engineering standards, a set of guides has been developed according to a set of VSE characteristics. The guides are based on subsets of appropriate standards elements, referred to as VSE Profiles. This set of guides are gathered in the so-called standard *ISO/IEC 29110 Software engineering — Lifecycle profiles for Very Small Entities* which is going to be overviewed during the next section.

3. ISO/IEC 29110: Overview of the standard

The *ISO/IEC 29110 Software engineering — Lifecycle profiles for Very Small Entities* standard is aimed to approach Software Engineering and Project Management good practices to VSEs. According to [1,19], the ISO/IEC 29110 standard is divided in five parts as follow:

ISO/IEC TR 29110-1 defines the business terms common to the VSE Profile Set of Documents. It introduces processes, lifecycle and standardization concepts, and the ISO/IEC 29110 series. It also introduces the characteristics and requirements of a VSE, and clarifies the rationale for VSE-specific profiles, documents, standards and guides.

ISO/IEC 29110-2 introduces the concepts for software engineering standardized profiles for VSEs, and defines the terms common to the VSE Profile Set of Documents. It establishes the logic behind the definition and application of standardized profiles. It specifies the elements common to all standardized profiles (structure, conformance, assessment) and introduces the taxonomy (catalogue) of ISO/IEC 29110 profiles.

ISO/IEC TR 29110-3 defines the process assessment guidelines and compliance requirements needed to meet the purpose of the defined VSEs Profiles. ISO/IEC TR 29110-3 also contains information that can be useful to developers of assessment methods and assessment tools. ISO/IEC TR 29110-3 is addressed to people who have direct relation with the assessment process, e.g. the assessor and the sponsor of the assessment, who need guidance on ensuring that the requirements for performing an assessment have been met.

ISO/IEC 29110-4-1 provides the specification for all the profiles of the Generic Profile Group. The Generic Profile Group is applicable to VSEs that do not develop critical software products. The profiles are based on subsets of appropriate standards elements. VSEs' Profiles apply and are targeted to authors/providers of guides and authors/providers of tools and other support material.

ISO/IEC 29110-5-1 provides an implementation management and engineering guide for both the Entry and Basic Profile of the Generic Profile Group described in ISO/IEC 29110-4-1. The Entry Profile describes software development of a single application by a single project team with no special risk or situational factors for start-up VSEs (i.e. VSEs who started their operation less than 3 years) and/or for VSEs working on small project (e.g. project size of less than 6 person-months). The Basic Profile describes software development of a single application by a single project team with no special risk or situational factors.

In the following subsections a summary of standard's part 5-1, with a special highlight in the *Project Management and Software Implementation* processes for the Entry Profile belongings to the Generic Profile Group, will be presented.

If one carefully sees, the lifecycle profiles follow the typical structure of an organization. The *advanced profile*[12] addresses the corporate level management, *Intermediate profile*[13] covers the tactical level management and *Entry and Basic profiles* address the operational level management; level that involves Project Management and Software Implementation processes. According to[14], the reason to include the Project Management process is that VSEs' core business is software development (Software Implementation process) and their financial success depends on project profits.

3.1. Project Management (PM) process – Purpose

The purpose of the Project Management process is to establish and carry out in a systematic way the tasks of the software implementation project, which allows complying with the project's objectives in the

[12] This profile targets VSEs, which want to sustain and grow as an independent competitive software development business.
[13] This profile targets VSEs developing multiple projects within the organizational context.

expected quality, time and cost. PM process uses the customer's statement of work to elaborate the project plan. The PM project assessment and control tasks compare the project progress against the project plan. The PM project closure activity delivers the software configuration, produced by the Software Implementation Process, and gets the customer's acceptance to formalize the end of the project. A project repository is established to save the work products during the project [1].

3.1.1. PM objectives

According to [1], the objectives are specific goals that ensure the accomplishment of the process purpose. The objectives are identified by the abbreviation of the process name, followed by the letter "O" and a consecutive number, for example PM.O1, SI.O2, etc. Each objective is followed by the square box, which includes a list of the chosen processes for the entry profile from ISO/IEC 12207:2008 and its outcomes related to the objective. In this paper, the chosen processes are only going to be numbered.

PM.O1. The *Project Plan* for the execution of the project is developed according to the *Statement of Work* and reviewed and accepted by the Customer. The tasks and resources necessary to complete the work are sized and estimated (ISO/IEC 12207:2008, 6.3.1, 6.3.7).

PM.O2. Progress of the project is monitored against the *Project Plan* and recorded in the *Progress Status Record*. Closure of the project is performed to get the Customer acceptance documented in the *Acceptance Record* (ISO/IEC 12207:2008, 6.3.2, 6.3.7, 6.4.8).

PM.O3. The *Changes Requests* are addressed, evaluated and tracked (ISO/IEC 12207:2008, 7.1.2).

PM.O4. Review meetings with the Work Team and the Customer are held. *Agreements* are registered and tracked (ISO/IEC 12207:2008, 7.2.6).

PM.O5. *Risks* are identified as they develop and during the conduct of the project (ISO/IEC 12207:2008, 6.3.4, 7.2.6).

PM.O6. Items of Software Configuration are identified and controlled (ISO/IEC 12207:2008, 7.2.2).

PM.O7. Software Quality Assurance is performed to provide assurance that work products and processes comply with the Project Plan and Requirements Specification (ISO/IEC 12207:2008, 7.2.3).

The authors in [1] also note that the implementation of the Software Quality Assurance process depicted in objective PM.07. is going to be achieved through the performance of the verifications, validations and review tasks performed in Project Management and Software Implementation processes.

3.1.2. Products

According to [1], artifacts of this process are classified in three groups:

Input Products – products required to perform the process and its corresponding source, which can be another process or an external entity to the project, such as the Customer. Identified by the abbreviation of the process name and showed as two column table of product names and sources (see Table 2).

Name	Source
Statement of Work	Customer
Software Configuration	Software Implementation
Change Request	Customer

Table 2. PM Input products [1]

Output Products – products generated by the process and its corresponding destination, which can be another process or an external entity to the project, such as Customer or Organizational Management. Identified by the abbreviation of the process name and showed as two column table of product names and destinations (see Table 3).

Name	Source
Project Plan	Software Implementation
Acceptance Record	Customer
Project Repository	Software Implementation
Meeting Record	Customer
Software Configuration	Customer

Table 3. PM Output products [1]

Internal Products – products generated and consumed by the process. Identified by the abbreviation of the process name and showed as one column table of the product names (see Table 4).

Name
Change Request
Meeting Record (only work team)
Progress status record

Table 4. PM Internal products [1]

3.1.3. PM roles involved

The roles are names and abbreviation of the functions to be performed by project team members. Several roles may be played by a single person and one role may be assumed by several persons. Roles are assigned to project participants based on the characteristics of the project (see Table 5) [1].

Name	Abbreviation	Competency
Customer	CUS	Knowledge of the Customer processes and ability to explain the Customer requirements. The Customer (representative) must have the authority to approve the requirements and their changes. The Customer includes user representatives in order to ensure that the operational environment is addressed. Knowledge and experience in the application domain.
Project Manager	PM	Leadership capability with experience making decisions, planning, personnel management, delegation and supervision, finances and software development.
Work Team	WT	Knowledge and experience according to their roles on the project.

Table 5. PM Roles involved [1]

3.1.4. PM activities

According to [1], an activity is a set of cohesive tasks and a task is a requirement, recommendation, or permissible action, intended to contribute to the achievement of one or more objectives of a process. A process activity is the first level of process workflow decomposition and the second one is a task. Activities are identified by process name abbreviation followed by consecutive number and the activity name.

PM.1 Project planning, (PM.O1, PM.O5, PM.O6, PM.O7)

The Project Planning activity documents the planning details needed to manage the project. The activity provides:

- Reviewed Statement of Work and the tasks needed to provide the contract deliverables and to satisfy customer requirements.
- Project quality assurance approach through verification and validation of work products/deliverables, customer reviews.
- Work team and customer roles and responsibilities.
- Project resources needs.
- Estimates of effort, cost and schedule.
- Identified project risks.
- Project repository to store, handle and deliver controlled product and document versions and baselines.

PM.2 Project plan execution (PM.O2, PM.O3, PM.O4, PM.O5, PM.O7)

The Project Plan Execution activity implements the documented plan on the project. The activity provides:

- Monitoring the project against the Project plan.
- Status of the Project Plan Execution.
- Change Request accepted by the Customer.
- Reviews and agreements with the Customer.

PM.3 Project assessment and control (PM.O2)

The Project Assessment and Control activity evaluates the performance of the plan. The activity provides:

- Evaluation of actual plan performance and progress against targets.
- Change requests tracking.
- Documented problem, corrective action defined, and tacked to closure.

PM.4 Project closure (PM.O2)

The Project Closure activity provides the project's documentation and products in accordance with contract requirements. The activity provides:

- Support of Customer product acceptance.
- Completion of the project and sign of the Acceptance Record.
- Summary and updated project repository for project closure.

In order to facilitate a better understanding of the standard's Project Management process structure, a table showing the interrelation between activities, objectives, roles and products is provided below (see Table 6).

ID	Activities	Associated objectives	Roles Involved	Products Involved
PM.1	Project planning	PM.O1. The Project Plan for the execution of the project is developed according to the Statement of Work and reviewed and accepted by the Customer. The tasks and resources necessary to complete the work are sized and estimated. PM.O5. Risks are identified as they develop and during the conduct of the project. PM.O6. Items of Software Configuration are identified and controlled. PM.O7. Software Quality Assurance is performed to provide assurance that work products and processes comply with the Project Plan and Requirements Specification.	PM, WT, CUS	Statement of Work, Project Plan
PM.2	Project plan execution	PM.O2. Progress of the project is monitored against the Project Plan and recorded in the Progress Status Record. Closure of the project is performed to get the Customer acceptance documented in the Acceptance Record. PM.O3. The Changes Requests are addressed, evaluated and tracked. PM.O4. Review meetings with the Work Team and the Customer are held. Agreements are registered and tracked. PM.O5. Risks are identified as they develop and during the conduct of the project. PM.O7. Software Quality Assurance is performed to provide assurance that work products and processes comply with the Project Plan and Requirements Specification.	PM, WT, CUS	Project Plan, Progress Status Record, Meeting Record, Change Request
PM.3	Project assessment and control	PM.O2. Progress of the project is monitored against the Project Plan and recorded in the Progress Status Record. Closure of the project is performed to get the Customer acceptance documented in the Acceptance Record.	PM, WT	Project Plan, Progress Status Record, Change Request
PM.4	Project closure	PM.O2. Progress of the project is monitored against the Project Plan and recorded in the Progress Status Record. Closure of the project is performed to get the Customer acceptance documented in the Acceptance Record.	PM, CUS	Project Repository

Table 6. Interrelation between PM activities, objectives, roles and products

3.2. Software Implementation (SI) process – Purpose

According to [1], the purpose of the Software Implementation process is the systematic performance of the analysis, software component identification, construction, integration and tests, and product delivery activities for new or modified software products according to the specified requirements. The execution of the SI process is driven by the project plan. SI process starts with an initiation activity of the project plan revision. Project plan will guide the execution of the software requirements analysis, software component identification, software construction, software integration and test, and product delivery activities.

3.2.1. SI objectives

SI.O1. *Tasks* of the activities are performed through the accomplishment of the current *Project Plan*.

SI.O2. *Software requirements* are defined, analyzed for correctness and testability, approved by the Customer, and communicated (ISO/IEC 12207:2008, 6.4.1, 7.1.2).

SI.O3. *Software components* and their *interfaces* are identified (ISO/IEC 12207:2008, 7.1.3).

SI.O4. Software components are produced. *Unit test* are performed to verify the consistency with software requirements (ISO/IEC 12207:2008, 7.1.5).

SI.O5. Software is produced. Software components are integrated and verified using *Test Cases and Test Procedures*. Results are recorded at the *Test Report*. Defects are corrected (ISO/IEC 12207:2008, 7.1.6, 7.1.7).

SI.O6. *Software configuration* is prepared for delivery (ISO/IEC 12207:2008, 6.1.2, 7.2.1).

SI.O7. *Verification and Validation* tasks of all required work products are performed to achieve consistency among output and input products in each activity. Defects are identified and corrected (ISO/IEC 12207:2008, 7.2.4, 7.2.5).

3.2.2. SI products

Artifacts of this process are classified in three groups and presented in the following tables:

Name	Source
Project Plan	Project Management
Project Repository	Project Management

Table 7. SI Input products [1]

Name	Source
Software Configuration: • *Requirements Specification* • *Software*	Project Management

Table 8. SI Output products [1]

Name
Software component identification
Test cases and test procedures
Software component
Test report

Table 9. SI Internal products [1]

3.2.3. SI roles involved

The roles involved in the Software Implementation process are the same which are involved in the Project Management process (Customer, Project Manager, Work Team).

3.2.4. SI activities

According to [1], Software Implementation activities are detailed as follow:

SI.1 Software Implementation initiation (SI.O1)

The Software Implementation Initiation activity ensures that the Project Plan, established in Project Planning activity, is committed to by the Work Team. The activity provides:

- Review of the Project Plan by the Work Team to determine task assignment.
- An implementation environment established.

SI.2 Software requirements analysis (SI.O2, SI.O6, SI.O7)

The Software Requirements Analysis activity analyzes the agreed customer requirements and establishes the validated project software requirements. The activity provides:

- Work Team review of the Project Plan to determine task assignment.
- Elicitation, analysis and specification of customer's requirements.
- Agreement on the customer requirements.
- Verification and validation of requirements.

SI.3 Software component identification (SI.O3, SI.O6, SI.O7)

The Software Component Identification activity transforms the software requirements to the architecture of system software components. The activity provides:

- Work Team review of the Project Plan to determine task assignment.
- Identify software components and associated interfaces.

SI.4 Software construction (SI.O4, SI.O6, SI.O7)

The Software Construction activity develops the software code and data from the Software Component Identification in the SI.3. The activity provides:

- Work Team review of the Project Plan to determine task assignment.
- Understand the identified Software Components.
- Test Cases and Test Procedures for unit and integration testing.
- Coded Software Components and applied unit tests.

SI.5 Software integration and tests (SI.O5, SI.O6, SI.O7)

The Software Integration and Tests activity ensures that the integrated software components satisfy the software requirements. The activity provides:

- Work Team review of the Project Plan to determine task assignment.
- Understanding of Test Cases and Procedures and the integration environment.
- Integrated Software Components, corrected defects and documented results.

SI.6 Product delivery (SI.O6, SI.O7)

The Product Delivery activity provides the integrated software product to the Project Manager and support for delivery. The activity provides:

- Work Team review of the Project Plan to determine task assignment.
- Delivery of the software product and applicable documentation in accordance with the Project Plan.

In order to facilitate a better understanding of the standard's Software Implementation process structure, a table showing the interrelation between activities, objectives, roles and products is provided below (see Table 10).

ID	Activities	Associated objectives	Roles Involved	Products Involved
SI.1	Software Implementation Initiation	SI.O1. Tasks of the activities are performed through the accomplishment of the current Project Plan.	PM, WT	Project Plan
SI.2	Software Requirements Analysis	SI.O2. Software requirements are defined, analyzed for correctness and testability, approved by the Customer, and communicated. SI.O6. Software configuration is prepared for delivery. SI.O7. Verification and Validation tasks of all required work products are performed to achieve consistency among output and input products in each activity. Defects are identified and corrected.	PM, WT	Project Plan, Requirements Specification
SI.3	Software Component Identification	SI.O3. Software components and their interfaces are identified. SI.O6. Software configuration is prepared for delivery. SI.O7. Verification and Validation tasks of all required work products are performed to achieve consistency among output and input products in each activity. Defects are identified and corrected.	PM, WT	Project Plan, Progress Status Record, Change Request, Software Component Identification
SI.4	Software Construction	SI.O4. Software components are produced. Unit test are performed to verify the consistency with software requirements. SI.O6. Software configuration is prepared for delivery. SI.O7. Verification and Validation tasks of all required work products are performed to achieve consistency among output and input products in each activity. Defects are identified and corrected.	PM, WT	Project Plan, Software Component Identification, Requirements Specification, Test Cases and Test Procedures, Software Components
SI.5	Software Integration and Tests	SI.O5. Software is produced. Software components are integrated and verified using Test Cases and Test Procedures. Results are recorded at the Test Report. Defects are corrected. SI.O6. Software configuration is prepared for delivery. SI.O7. Verification and Validation tasks of all required work products are performed to achieve consistency among output and input products in each activity. Defects are identified and corrected.	PM, WT	Project Plan, Test Cases and Test Procedures, Software Components, Software, Test Report, Requirements Specification, Software Configuration
SI.6	Product Delivery	SI.O6. Software configuration is prepared for delivery. SI.O7. Verification and Validation tasks of all required work products are performed to achieve consistency among output	PM, WT	Project Plan, Software Configuration

		and input products in each activity. Defects are identified and corrected.		

Table 10. Interrelation between SI activities, objectives, roles and products

4. ISO/IEC 29110: Implementation in VSEs

In this section, documented and published research work regarding to adoption and implementation of the standard in small organizations are going to be shown. In [20], the authors proposed a three-step approach of the standard in order to implement it in a small Thai government academic institute's IT department. The so-called approach consists of:

A Feasibility Study: This step emphasized on finding the possibility of adapting ISO/IEC 29110 standard into the existing software processes used by the development unit.

Risk Management: All risks were identified and evaluated based on the results of the feasibility study in order to manage the risk that may occur during the implementation. The risk management plan was created and the risk mitigation was defined.

The Execution: After considering the feasibility study result and the risk management report, the implementation of the ISO/IEC 29110 processes (PM and SI) was executed. All constraints defined in the previous steps were reviewed and monitored throughout the implementation plan.

With the three-step implementation approach (the authors say), the case study unit gained a better understanding in the ISO/IEC 29110 standard, subsequently was able to effectively handle and prepare documents under the standard. As a result, the unit has a clear, well-defined, step-by-step approach in the software development that leads to a better reputation of the organization. The authors' findings pointed out the significance of such standard with respect to apply their processes in in-house software units under government agencies.

In [21], the authors talk about Deployment Packages (DPs). Their main objective of DPs is to facilitate the implementation, by VSEs, of a Profile. A deployment package is a set of artifacts developed to facilitate the implementation of a set of practices, of the selected framework, in a VSE. DPs are available, at no cost, on the Internet [1]. This paper outline a pilot project initiative currently underway to evaluate these Deployment Packages and assist very small companies in understanding and exploring the potential usage of an international software process development standard like ISO/IEC 29110.

About pilot projects, they mention there is a series of projects that have been taking place in Canada, utilizing some of the deployment packages developed. For example in Canada a pilot study has been conducted with an IT department with a staff of 4: 1 analyst and 3 developers, who were involved in the translation and implemented 3 DPs: Software Requirements, Version Control and Project Management. In Belgium a VSE of 25 people started with a process assessment phase aiming to identify strengths and weaknesses in development related processes. This company is now working on improvement actions mainly based on the following Deployment Packages: Requirement Analysis, Version Control, and Project Management. Finally in Ireland a VSE of 8 people are working on improving project management and tracking and control practices using the Project Management deployment package. In [22], the authors also comment about a pilot project conducted with a 14-person VSE based in France, which successfully implemented ISO/IEC 29110 processes practices utilizing the available Deployment Packages.

The authors continue explaining that the Brazilian Standard Organization ABNT (Associação Brasileira de Normas Técnicas) has developed an ISO/IEC 29110 certification scheme. A first series of Brazilian VSEs should have obtained an ISO/IEC 29110 certificate of conformity during this year. The auditing scheme, developed by Brazil, will probably be used by other countries, such as Canada, to audit their VSEs. It is also known that the Peruvian National Institute for the Defense of Competition and Protection of Intellectual Property (INDECOPI) has adopted in 2012 the ISO/IEC 29110 international standard, as the NTP-RT-ISO/IEC TR 29110 Peruvian Technical Standard (Norma Técnica Peruana), which is making easier de implementation of the standard in small Peruvian software firms [23].

In [24], authors present results of implementing ISO/IEC 29110 standard (Basic profile part 5-1-2), using DPs, through the performance of pilot projects in very small Irish companies. VSEs were invited to participate in a training program in order to implement the standard. Seven VSEs joined the program. The adoption of the standard was performed in a 4-step-method:

- VSEs were sent a DP and other supporting material.
- VSEs implement the process and report on activities, successes and problems to the researchers.
- The researchers review the reports and return any useful comments to the companies.
- The researchers make any amendment to the process to ensure greater success with the next process module.

After a 3-month period, four of the firms stopped the implementation of the standard, another just quitted, the sixth one did not even started the program after an initial expression of interest and the last company stopped and then restarted work on the standard implementation and submitted some documentation after a while. The authors conclude saying that VSEs have too much work to do, with too little time and people to do it. This was supported by one company, who commented that they do not even know if they are going to be *"in business"* next month, so implementing a standard would be too much workload for them. One can evidence that in some cases, a standard is still viewed as an add-on task, not a way to do business. Nevertheless, despite the lack of apparent success in terms of bringing all companies successfully through this program, the researchers are optimistic about the future for this standard. The authors have detected the need of enhancing the mentoring and assessment labor with VSEs in order to adequately implement this type of programs.

Another initiatives for the dissemination and adoption of the standard are the Network Support Centers (NSC) commented by authors in [22]. The main purpose of NSC, born from an informal meeting conducted by WG24[14] delegates in order to create a network of collaborators, is to facilitate and develop collaborative activities between institutions in the field of software engineering, information technology and others to improve VSE capabilities especially in Software Engineering and Information Technology. The principal goals to achieve by the implementation of NSC are clear: Speed up both, the deployment of Standard and Guides for VSEs and the development and application of Guides and DPs (e.g. through pilot projects). Some participants of the Network Support Centers are:

- Center of Excellence in Information and Communication Technologies (CETIC) - **Belgium.**
- RIOSOFT agent for Brazilian software excellence in Rio de Janeiro - **Brazil.**

[14] The ISO/IEC JCT1/SC7 Working Group 24 was established in 2005 with a mandate to investigate the need for and propose software life cycle profiles and guidelines for use in VSEs [22].

- Superior School of Technology (ETS) - **Canada.**
- Parquesoft Foundation - **Colombia.**
- Tampere University of Technology, Pori - **Finland.**
- University of Western Brittany (UBO) - **France.**
- Quisqueya-America University Institute (INUQUA) - **Haiti.**
- Polytechnic University - **Hong Kong (China).**
- Lero, The Irish Software Engineering Research Center - **Ireland.**
- Public Research Center Henri Tudor - **Luxembourg.**
- University of Lima - **Peru.**
- Institute of Software Promotion for Industries - **Thailand.**

Further countries, such as Ecuador, Mexico, Spain and Japan are considering joining the NSC.

5. Conclusions

As far as one can see, it has come a long way up to the creation of a resilient standard to carry out processes of the software lifecycle in VSEs, from adapting previous ISO and ISO/IEC standards, to the construction of models such as MoProSoft, essence of the ISO/IEC 29110 standard. The adoption has been sometimes difficult as depicted in [24], and sometimes easier but still incipient as described in [20]. One might think that the main efforts in order to raise awareness among VSEs relative to the standard should be done on the Deployment Packages to bring a more simple way to implement it, as well as on the Network Support Centers, as a mean to accelerate its deployment in small and medium software firms.

One must agree with [21] that ISO/IEC 29110, as an emerging standard, has yet work to be done. The main remaining work item is to finalize the development of the two remaining profiles: *Intermediate - Management of more than one project* and *Advanced - Business management and portfolio management practices*. In addition, the development of further Profile Groups for other domains such as: Critical software, game industry, scientific software development, etc.

Finally, after carefully reviewing the existent literature, only the time will tell, according to the worldwide degree of standard's adoption, if it is fully focused on VESs. The history of Software Engineering shows that is necessary to harmonize actual practices with new proposals, therefore the recent creation of the Entry Profile and Deployment Packages are perhaps not enough to facilitate its implementation, but rather the creation of a initial framework in order to guide the implementation process is needed, framework that prepares VSEs to successfully adopt the standard, particularly VSEs located in developing countries.

References

[1] ISO/IEC, “ISO/IEC TR 29110-5-1-1:2012 Software engineering -- Lifecycle profiles for Very Small Entities (VSEs) -- Part 5-1-1: Management and engineering guide: Generic profile group: Entry profile.” ISO Central Secretary, 15-Sep-2012.

[2] C. Y. Laporte, A. April, and A. Renault, “Applying ISO/IEC software engineering standards in small settings: historical perspectives and initial achievements,” in *Proceedings of SPICE Conference, Luxembourg*, 2006.

[3] H. Oktaba, “3.2 MoProSoft®: A Software Process Model for Small Enterprises,” in *International Research Workshop for Process Improvement in Small Settings*, 2006, p. 93.

[4] C. Laporte, R. O'Connor, and G. Fanmuy, "International systems and software engineering standards for very small entities," *CrossTalk J. Def. Softw. Eng.*, vol. 26, no. 3, pp. 28–33, 2013.
[5] C. Y. Laporte, S. Alexandre, and R. V. O'Connor, "A software engineering lifecycle standard for very small enterprises," *Softw. Process Improv.*, pp. 129–141, 2008.
[6] T. Varkoi and T. Mäkinen, "A Process Assessment Model for Very Small Software Entities," in *In: Rout, T., Lami, G. & Fabbrini, F.(eds.). Process Improvement and Capability Determination in Software, Systems Engineering and Service Management, Proceedings of: 10th International SPICE Conference 2010, Pisa, Italy 18-20 May, 2010*, Pisa, Italy, 2010.
[7] R. Singh, "International Standard ISO/IEC 12207 software life cycle processes," *Softw. Process Improv. Pract.*, vol. 2, no. 1, pp. 35–50, 1996.
[8] ISO/IEC, "ISO/IEC 15289 Systems and software engineering — Content of life-cycle information products (documentation)." ISO Central Secretary, 11-Jan-2011.
[9] K. El-Emam and I. Garro, "ISO/IEC 15504," *Int. Organ. Stand.*, Nov. 1999.
[10] M. C. Paulk, "Analyzing the conceptual relationship between ISO/IEC 15504 (software process assessment) and the capability maturity model for software," in *1999 International Conference on Software Quality*, 1999.
[11] ISO, "Norma Internacional ISO 9001 Sistemas de gestión de la calidad — Requisitos (Traducción oficial)." ISO Central Secretary, 15-Nov-2008.
[12] R. Prikladnicki, J. Audy, and R. Evaristo, "Requirements management in global software development: Preliminary findings from a case study in SW-CMM context," in *The International Workshop on Global Software Development*, 2003, pp. 53–58.
[13] C. P. Team, "CMMI for Development, version 1.2," Aug. 2006.
[14] M. M. Trujillo, H. Oktaba, T. Ventura, and R. Torres, "From MoProSoft Level 2 to ISO/IEC 29110 Basic Profile: Bridging the Gap.," in *CIbSE*, 2012, pp. 28–41.
[15] M. Llaneza, G. N. Dapozo, C. L. Greiner, and M. G. Estayno, "Análisis comparativo de modelos de calidad orientado al desarrollo de software en pymes," in *XV Workshop de Investigadores en Ciencias de la Computación*, Paraná, Argentina, 2013.
[16] M. J. Escalona, J. J. Gutiérrez, F. Morero, C. L. Parra, J. Nieto, F. Pérez, F. Martín, and A. Llergo, "A Practical Environment to Apply Model-Driven Web Engineering," *Inf. Syst. Dev.*, pp. 249–258, 2010.
[17] J. A. H. Alegrıa and M. C. Bastarrica, "Implementing CMMI using a combination of agile methods," *CLEI Electron. J.*, vol. 9, no. 1, 2006.
[18] A. Anacleto, C. G. von Wangenheim, C. F. Salviano, and R. Savi, "Experiences gained from applying ISO/IEC 15504 to small software companies in Brazil," in *4th International SPICE Conference on Process Assessment and Improvement, Lisbon, Portugal*, 2004, pp. 33–37.
[19] 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." ISO Central Secretary, 15-May-2011.
[20] V. Siddoo, N. Wongsai, and R. Wetprasit, "An Implementation Approach of ISO/IEC 29110 for Government Organizations."
[21] R. V. O'Connor and C. Y. Laporte, "Towards the provision of assistance for very small entities in deploying software lifecycle standards," in *Proceedings of the 11th International Conference on Product Focused Software*, 2010, pp. 4–7.
[22] R. V. O'Connor and C. Y. Laporte, "An Innovative Approach to the Development of an International Software Process Lifecycle Standard for Very Small Entities.," *Int. J. Inf. Technol. Syst. Approach*, 2013.
[23] INDECOPI, "NTP-RT-ISO/IEC TR 29110-5-1-2. Ingeniería de Software. Perfiles del ciclo de vida para las pequeñas organizaciones (PO). Parte 5-1-2: Guía de gestión e ingeniería: Grupo de perfil genérico. Perfil básico." INDECOPI, 16-May-2012.
[24] R. V. O'Connor and M. Sanders, "Lessons from a Pilot Implementation of ISO/IEC 29110 in a Group of Very Small Irish Companies," in *Software Process Improvement and Capability Determination*, 2013, vol. 349, pp. 243–246.