Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 68 (2021) 2988–2998

ADVANCES IN SPACE RESEARCH
(a COSPAR publication)

www.elsevier.com/locate/asr

Developments in Aerospace Software Engineering practices for VSEs: An overview of the process requirements and practices of integrated Maturity models and Standards

Ricardo Eito-Brun, Antonio Amescua-Seco

Department of Information and Documentation, Universidad Carlos III, Madrid, Spain
Department of Computer Engineering, Universidad Carlos III, Madrid, Spain

Received 21 October 2020; received in revised form 21 May 2021; accepted 25 May 2021
Available online 3 June 2021

Abstract

As part of the evolution of the Space market in the last years – globally referred to as Space 2.0 - small companies are playing an increasingly relevant role in different aerospace projects. Business incubators established by European Space Agency (ESA) and similar entities are evidence of the need of moving initiatives to small companies characterized by greater flexibility to develop specific activities. Software is a key component in most aerospace projects, and the success of the initiatives and projects usually depends on the capability of developing reliable software following well-defined standards. But small entities face some difficulties when adopting software development standards that have been conceived thinking on larger organizations and big programs. The need of defining software development standards tailored to small companies and groups is a permanent subject of discussion not only in the aerospace field, and has led in recent years to the publication of the ISO/IEC 29110 series of systems and software engineering standards and guides, aimed to solve the issues that Very Small Entities (VSEs) () – settings having up to twenty-five people -, found with other standards like CMMI® or SPICE.

This paper discusses the tailoring defined by different aerospace organizations for VSEs in the aerospace industry, and presents a conceptual arrangement of the standard based on meta-modeling languages that allow the extension and full customization with the incorporation of specific software engineering requirements and practices from ECSS (European Cooperation for Space Standardization).

Keywords: Software development; Software engineering standards; SMEs; SPICE; Software; Maturity models

1. Introduction

Very Small Entities (VSEs), settings having up to twenty-five people, are a subset of Small and Medium Enterprises (SMEs). Today, there is a clear agreement on the positive impact that both SMEs and VSEs have on the global economy. OCDE Development, Small and Medium Enterprise Outlook (2019) indicates that "*SMEs account for 99% of all businesses, between 50% and 60% of added value with similar patterns across OECD countries*".

In the Aerospace sector, ESA System for Tendering And Registration (ESA STAR) registration system lists more than 2500 entities within the SME category, with a subcategory for "micro" entities with at most ten people. Universities' research departments involved in aerospace projects should also be added to this list of small entities developing partial solutions that are later integrated within larger, more complex systems.

VSEs are characterized by their capability to innovate and develop new concepts and ideas (Cerezo-Narváez et al., 2019; Castillón-Barraza et al., 2018). But, on the

E-mail address: reito@bib.uc3m.es (R. Eito-Brun)

https://doi.org/10.1016/j.asr.2021.05.026
0273-1177/© 2021 COSPAR. Published by Elsevier B.V.

other side, the need of developing their capabilities as suppliers of larger programs has been recognized as a strategic requirement to achieve a successful, long-term partnership (Reed and Walsh, 2002; Hussein and Cheng, 2016). In the case of VSEs providing software-based solutions, the development of those capabilities depends on the availability of an agreed, widely accepted standard to assess the VSEs' process and provide guidance for their internal improvement programs.

This paper provides an overview of one of the latest achievements in this line of work: the development of the Maturity Model for VSEs in the Space Domain, and presents the result of a research aimed to complement that model with specific requirements coming from European Cooperation for Space Standardization (ECSS) standards for software development. The result of this work complements the maturity model (ESA et al., 2018) with an extended process representation based in the SPEM modeling language that allows customization and tailoring, helping VSEs achieve a better understanding of the standards' requirements.

The definition of SMEs characteristics has always been the subject of discussion and different schemas proposed to decide when an entity is part of this group and its subcategories. Art. 2 of the European Commission (EC) Recommendation of 6 May 2003, which is the schema adopted by ESA, distinguishes three types of SMEs:

- Small to medium, with less than 250 employees and annual turnover less than 50 M euros, and annual balance sheet not exceeding 43 million euros.
- Small, with less than 50 employees and annual turnover below 10 million euros and
- Micro, with less than 10 employees and annual turnover below 2 million euros.

The Maturity Model for VSEs in the space domain (ESA et al., 2018), which is briefly described in section 4, makes an explicit reference to enterprises with no more than 25 people and extends its applicability to departments and project teams up to twenty-five people that may belong to larger organizations.

In the aerospace software development sector, these entities should develop software and systematically apply standard processes to ensure that their outputs meet the demanding requirements of projects. But the difficulties that VSEs face to adopt standard processes have not only been a challenge for entities in the aerospace sector; the software industry, in general, has been sensitive to this need, and different initiatives were developed in the past until the publication of the ISO/IEC 29110 family of standards. ISO/IEC 29110 defines technical and managerial processes, activities, tasks, and work products suited to the characteristics of VSEs and provides a common vocabulary to ensure fluent communication between these entities and upper-level contractors. Before its publication, the use of classical process models like CMMI® and SPICE by SMEs was largely discussed in professional and academic literature. Basri and O'Connor (2010) provided an exhaustive list of factors that made hard the adoption of those models by SMEs: simple development cycles with missed phases and activities, variability of maturity levels at different processes, informal quality control procedures, limited resources for training, short term strategies, etc. The adoption of CMMI® and SPICE was identified with additional costs, bureaucracy, and delays. VSEs' lack of interest in the adoption of a standard software development process was also discussed by O'Connor and Laporte (2011) using the SEI CMMI® data. Similar conclusions were also reported by Felderer and Ramler (2016). In general, it was accepted that traditional software improvement models imposed significant overheads on VSEs, as they did not have at their disposal the time, human and financial resources needed to answer the requirements defined by these complex standards (Ribaud and Saliou, 2010; Raninen et al., 2013; Takeuchi et al., 2014).

This translated into difficulties for VSEs to demonstrate their capability to develop quality software, and upper-level contractors had to assess them as subcontractors using complex models not suited to their objective capabilities (Sánchez-Gordon and O'Connor, 2016). Today, with ISO/IEC 29110, different players have a common, simplified set of processes, activities, tasks, and work products to conduct both improvement programs and assess suppliers' capabilities (Rodriguez´-Dapena and Buitrago-Botero, 2015). The ISO 29110 systems and software engineering series provide a four-stage roadmap from startup to grownups for VSEs (Laporte et al., 2018; Laporte and Mejia, 2020)

The rest of this paper is organized as follows: section 2 introduces the principal achievement in process standardization for VSEs: the ISO/IEC 29110; section 3 presents the maturity model for VSEs in the aerospace industry; sections 4 and 5 describes the research conducted to extend the model with software development requirements using standard modeling techniques and reports the conclusions.

2. ISO/IEC 29110 series of standards and guides

The origin of ISO/IEC 29110 dates to 2004, when ISO/IEC JTC1/SC7 reached a consensus on the need of defining standards and guides suited to the size and characteristics of VSEs, including profiles and guidelines. VSE is defined as "an enterprise, organization (e.g., government agency, non-profit organization), department or project having up to 25 workers". An ISO New Work Item (NWI) was approved in 2005 with the objectives of providing VSEs with access to software engineering standards, documentation that required a minimum adaptation effort for processes, work products, and defining capability and maturity levels similar to those in SPICE and CMMI®. The same year, the plenary meeting of ISO SC7 held in Finland launched a working group (WG24) that prepared a questionnaire addressed to VSEs to validate these hypotheses:

- VSEs' business context requires lighter life cycles.
- VSEs have constraints on time and resources, which made difficult a clear understanding of the benefits that standards provide.
- The benefits that VSEs could obtain from this standard included the recognition of their activities through assessments and audits by accredited entities.

The working group received 392 answers, most of them from companies with less than twenty-five employees (70% of them worked on regulated sectors and some of them developed critical software). The conclusions identified that 28% of the VSE did not use standards due to lack of resources or because standards imposed bureaucratic workload. But 74% of the companies recognized the need of being assessed with respect to standards, and 62% highlighted the need of having guidelines, examples, and templates (Laporte et al., 2008).

The first parts of ISO/IEC 29110 were finally published in 2011. It focused on the definition of *profiles*, defined in part 2 of the standard as "*pre-adapted packages of related software engineering standards*". From a practical point of view, Laporte et al. (2013) defined a profile as "*a matrix that identifies what elements are taken from existing standards, and which ones are excluded.*" Their definition is under the responsibility of the ISO/IEC JTC1. Documents were developed to provide the specifications of the different profiles (Entry, Basic, Intermediate, and Advanced) for software and systems engineering. Profiles specify management and engineering processes, expected outcomes, activities, tasks, and work products. Profiles reuse elements already defined in other standards called *base standards*. The selection of elements from base standards is made according to the recommendations and guidelines in ISO/IEC 29110-2. In general, the rational of the ISO/IEC 29110 series is intended to be used by VSEs that do not have experience or expertise in adapting and tailoring ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288 standards to the needs of a specific project.

Profiles are grouped into profile groups based on processes, activities, tasks, and capability levels common to different contexts. Until now, a single profile group called "Generic for VSEs that develop or maintain non-critical SW" has been defined by ISO. Profiles within this group are not related to a specific domain and could be adopted in any situation where non-critical software is developed.

2.1. ISO/IEC 29110 document structure

ISO/IEC 29110 was structured into five groups of documents, each one addressed to a different audience: a) one document that offers a general view of the standard, terms, and concepts; b) another one that specifies the elements common to all the profiles; c) one single document that defines the certification schema, recommendations for assessments and requirements to conduct compliant assessments; d) one or more documents with the specification of the profiles, and the description of what must be done for their adoption. (Laporte and O'Connor, 2016)

According to this approach, five separate parts of the standard ISO/IEC 29110 have been published (their current versions are referenced):

1. ISO/IEC 29110-1:2016 "Overview" with the purpose and scope of the standard and a glossary with terms from ISO/IEC 15504-1 and ISO 9000.
2. ISO/IEC 29110-2, which described the process to create profiles and introduce the Generic Profile Group and its four profiles (Entry, Basic, Intermediate, and Advanced). It was later divided into:
    - o ISO/IEC 29110-2-1:2015 "Framework and Taxonomy" that describe the process to create profiles and introduce the Generic Profile Group and its four profiles (Entry, Basic, Intermediate, and Advanced)
    - o ISO/IEC TR 29110-2-2:2016 "Guide for the development of domain-specific profiles."
3. ISO/IEC 29110-3, with information on how to conduct conformity assessments. It was divided into three separate documents:
    - ISO/IEC TR 29110-3-1:2020 "Process assessment guidelines".
    - ISO/IEC 29110-3-2:2018 "Conformity certification scheme Foreword".
    - ISO/IEC TR 29110-3-4:2015 "Autonomy-based improvement method".
4. ISO/IEC 29110-4-n, with separate documents for these profiles:
    - o ISO/IEC 29110-4-1:2018 "Software engineering - Profile specifications: Generic profile group"
    - o ISO/IEC 29110-4-3:2018 "Service delivery — Profile specification"
5. ISO/IEC TR 29110-5-n-m "Management and Engineering guides" – with recommendations to adopt and implement each profile. Today, there are recommendations published for the Entry, Basic, Intermediate and Advanced profiles published as 29110-5-1-1, 29110-5-1-2, 29110-5-1-3, and 29110-5-1-4. From a practical perspective, these guidelines constitute the core documents to guide the adoption of the standard. Other documents within this group include 29110-5-2-1 for Organizational management guidelines, and 29110-5-6-1, 5-6-2, 5-6-3, and 5-6-4 for systems engineering.

Only parts 2, 4, and 3-3 are normative, and the other ones are published as technical reports (TR). Parts 1 and 2 are unique documents, and parts 3, 4, and 5 have different subdivisions per and assessment models (part 3), profile group (part 4), and implementation guidelines (part 5). In parts 4 and 5, the numbering schema reflects the relationship between sub-parts: e.g. the code for the Generic profile group is 29110-4-1, and the guidelines for that group are identified as 29110-5-1-1 (Entry), 29110-5-1-2 (Basic), etc.

If we focus on the profiles within the Generic Profile group, it includes four profiles: Entry, Basic, Intermediate, and Advanced. Entry profile is not subject to certification, although it can be adopted following ISO/IEC 29110-5-1-1 guidelines for projects with an estimated effort of six man-months in start-ups (O'Connor and Laporte, 2010). The most relevant profile is the Basic, which was designed for entities developing a single software application with a single project team, with no risks due to special situational factors. It was defined using as inputs two base standards: ISO 12207:2008 "Systems and software engineering – Software life cycle processes" and ISO 15289:2008 "Systems and software engineering – Content of systems and software life cycle process information products (Documentation)". Entities adopting the Basic profile must fulfill four requirements listed in chapters 9.3.2.3 of ISO/IEC 29110-2 and chapter 6.7 of ISO/IEC 29110-4:

- There must be an explicit contract on the agreement, with a statement of work between the supplier and the client.
- The feasibility of the project must be assessed before committing to start the activities.
- Staff with the necessary training and skills must be assigned, including a project manager.
- The resources, services, and infrastructure needed to develop the project must be provided.

The software Basic profile establishes two processes: Project management (PM) and Software Implementation (SI). The selection of these processes was due to the evidence that most VSEs depend on their capability to execute projects on time and within budget. Guidelines for the Basic Profile implementation are given in the ISO/IEC TR 29110-5-1-2, with a detailed description of their objectives and outcomes, activities, tasks, and work products.

Although ISO/IEC 29110 series provide standards and guides for the development of software or systems made up of hardware and software, the tailoring for Space refers only to software development.

2.2. Adaptations of ISO/IEC 29110

Different proposals have discussed the integration of ISO/IEC 29110 with other process models, being the most significant the adaptation of the Basic profile to systems engineering activities. This initiative was launched in April 2009 by AFIS and INCOSE representatives from Canada, France, Germany, and the USA, using ISO/IEC/IEEE 15288 as the base standard instead of ISO/IEC 12207 (Laporte et al., 2012). The first ISO draft of the adaptation for systems engineering was released in 2012 and the profile was accepted and published in 2014. It defines two processes (Project Management PM and System Implementation SY). Recommendations with details for its interpretation and implementation were also defined in line with the *INCOSE Handbook for Systems Engineering* (INCOSE, 2015).

The integration of ISO/IEC 29110 with other models has been widely discussed in the literature. Mas and Mesquida (2013, 2014) proposed the combined use of ISO/IEC 29110 PM with a subset of practices from the PMI PMBOK® (Project Management Institute, 2017). The feasibility of using ISO/IEC 29110 with agile software development methodologies (ASDM) was studied by Galvan et al. (2015, 2018); although this approach is quite interesting - agile methodologies have been widely adopted by VSEs -, its work was restricted to the PM process. The combined use of ISO/IEC 29110 and the Unified Process's requirements management was analyzed by Alvarez and Hurtado (2014). Eito-Brun and Sicilia (2017) developed a process model that merged ISO 29110 processes with innovation management practices; its use in critical software projects has also been discussed (Diniz et al., 2019). But the development of a Maturity Model for VSEs in the Space domain (Rodríguez-Dapena and Lohier, 2017) is probably the most significant achievement together with the development of the systems engineering profiles previously discussed.

3. ISO/IEC 29110 in aerospace software engineering

The design of an assessment model for aerospace VSEs started in 2017 when different entities recognized the need of developing a maturity model to guide improvement programs and to assess small software development entities and projects. Involved participants included representatives of the European Space Agency (ESA), Centre National D'Etudes Spatiales (CNES), Deutsches Zentrum für Luft- und Raumfahrt (DLR), Japan Aerospace Exploration Agency (JAXA), European Organization for the Exploitation of Meteorological Satellites (EUMETSAT), Instituto Nacional de Pesquisas Espaciais (INPE) and the Softwcare company. With this model, VSEs in the aerospace sector would have a framework to demonstrate the quality of their processes and their compliance with an agreed, standardized subset of software development processes. The preliminary version of the model was published as a draft in 2018 and is now under evaluation with the collaboration of different companies. Target stakeholders of the model include VSEs, which can use the model to guide process improvements and complete self-assessments, and prime contractors interested in a common set of criteria to evaluate small subcontractors' capabilities. The model provides a clear set of criteria to guide assessments ensuring consistent results. The maturity model presents a structure similar to ISO/IEC 15504-5, SPICE for Space or Automotive SPICE, with a distinction between the Process Reference Model (PRM) and the Process Assessment Model (PAM).

At the time of writing this document, there are two ISO/IEC 29110 documents under development to provide processes for VSEs developing software for the Space domain: ISO/IEC 29110-6-1 - *Specific Space Profile Specifications* and ISO/IEC TR 29110-7-1 *Space-VSEs Profile Guidelines*.

3.1. The process reference model

PRM defines the process dimension and consists of a set of processes with their definition, scope, purpose, goals, and outcomes (i.e. observable results expected from the successful performance of the processes). The PRM does not detail specific methods or procedures nor prescribe data or documentation requirements, life cycle, use of tools, or development methodologies. It includes nine development processes and four organizational processes adopted from ISO/IEC 29110 and ECSS-Q-HB-80-02, "SPICE for Space". Table 1 summarizes the purpose of the different processes. The model adopts the SI and PM processes from the software Basic profile of ISO/IEC 29110-4-1 and from ISO/IEC 29110-4-2 (those in the *Organizational processes* group targeted to a higher maturity level that exceeds specific project activities). ECSS-Q-HB-80-02 Part 2A "SPICE for Space" processes are incorporated to deal with technical activities not covered by the software profiles of the ISO/IEC 29110.

The model gives organizations the possibility to decide which processes to apply, as "*particular projects or organizations may not need to use all of the processes provided by this document. Therefore, implementation of this document typically involves selecting a set of processes suitable to the organization or project*.", and allows supplementing the model with additional procedures, practices, and tools (ESA et al., 2018, p. 8)

3.2. The process assessment model

The PAM links the Process and the Capability dimensions. It responds to the same approach followed by SPICE process improvement models that measure the capability of the process for reaching predictable results "*by demonstrating the achievement of process attributes based on evidence related to indicators.*" (ESA et al., p. 17) SPICE-related standards define two types of indicators: a) process perfor-

Table 1
Maturity Model PRM Process (ESA et al., 2018).

Process Group	Process	Process name and purpose	Source Std.
Engineering	**ENG.1**	**Requirements Elicitation** To gather, process, and track evolving customer needs and requirements throughout the life of the product and/or service so as to establish a requirements baseline that serves as the basis for defining the needed work products.	1.1.1.1.1.1.1.1.1 ECSS-Q-HB-80-02
	SI	**Software Implementation** Systematic performance of the analysis, design, construction, integration, and tests activities for new or modified software products according to the specified requirements.	1.1.1.1.1.1.1.1.2 SO/IEC 29110-4-1
	ENG.8	**Software Testing** Confirm that the integrated software product meets its defined requirements.	ECSS-Q-HB-80-02
Management	**PM**	**Project Management** 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 expected quality, time and costs.	ISO/IEC 29110-4-1
Supporting	**SUP.1**	**Quality Assurance** Provide assurance that work products and processes comply with predefined provisions and plans.	ECSS-Q-HB-80-02
	SUP.2	**Verification** Confirm that each software work product and/or service of a process or project properly reflects the specified requirements.	ECSS-Q-HB-80-02
	SUP.8	**Configuration Management** Establish and maintain the integrity of the work products / items of a process or project and make them available to concerned parties.	ECSS-Q-HB-80-02
	SUP.9	**Problem Resolution Management** Ensure that all discovered problems are identified, analyzed, managed and controlled to resolution.	ECSS-Q-HB-80-02
	SUP.11	**Safety and Dependability Assurance** Ensure the required degree of safety and dependability of the software products.	ECSS-Q-HB-80-02
Organizational	**RM**	**Resource Management** Obtain and provide the organization with the necessary resources.	ISO/IEC 29110-4-2
	PSM	**Process Management** Establish and improve the organizational processes of the VSE.	ISO/IEC 29110-4-2
	PPM	**Project Portfolio Management** Generate projects for the VSE, provide technical content to establish the project's Formal Agreement, and supervise its performance while monitoring customer satisfaction.	ISO/IEC 29110-4-2
	OM	**Organizational Management** Make sure that value is delivered by the organization to the customer through planning, organizing, monitoring, and controlling organizational activities	ISO/IEC 29110-4-2

mance indicators and b) process capability indicators. Performance indicators are defined for each process and correspond to *significant activities or results associated with the achievement of the process*. From a practical perspective they are a) base practices and b) work products. Base practices are activities that help achieve the process purpose and expected outcomes; they indicate what should be done without specifying how. The PAM identifies a set of base practices for each process. Work products are the observable result of the execution of the process. The PAM lists fifty-three work products and their characteristics. According to that approach, VSEs adopting the model should provide evidence of the performance of the base practices and the presence of work products. Both types of performance indicators are used to measure the achievement of the PA.1 performance attribute that determines the achievement of the ALPHA maturity level (equivalent to a CL1 of SPICE). As an example, table 2 shows the Base Practices specified for the *ENG.1 Requirements Elicitation* process.

3.3. Process capability and organization maturity levels

The last point covered by the Maturity Model refers to the capability levels and the indicators used to decide if they are achieved or not. These indicators can be seen as requirements that entities must fulfill to reach a target capability level in a specific process. Capability levels achieved for each process can then be used to decide on the maturity level of the organization. The model defines three capability levels: Alpha, Beta, and Gamma.

- ALPHA or Performed process, means that the process achieves its purpose and outcomes. This level is achieved by executing the process-related base practices and producing its work products.
- BETA or Articulated process, means that the process is implemented in a managed, monitored way and follows a defined process. This level is built on top of the ALPHA level and is assessed using two process attributes: PA.2 Management of resources and PA.3 Defined process. It is worth mentioning that this approach makes a difference concerning other models based on SPICE, where different capability levels (2 and 3) are used to distinguish between the processes that are executed in a managed way and the processes that are executed based on a defined standard process.
- GAMMA or Aligned process, means that the process is aligned with organizational business goals and ensures customer satisfaction. This capability level is linked to the PA.4 process attribute, which requests the definition of business goals, collecting and analyzing data, and taking corrective and preventive actions.

Table 2
PAM at the Maturity Model (ESA et al., 2018, p. 20-21).

Process ID	ENG.1
Process Name	Requirements Elicitation
Process Purpose	The purpose of the Requirements Elicitation process is to gather, process, and track evolving customer needs and requirements throughout the life of the product and/or service so as to establish a requirements baseline that serves as the basis for defining the needed work products. Requirements elicitation may be performed by the acquirer or the developer of the system.
Process Outcomes	As a result of successful implementation of the Requirements Elicitation process: • continuing communication with the customer is established; • agreed customer requirements are defined and baselined; • a change mechanism is established to evaluate and incorporate changes to customer requirements into the baselined requirements based on changing customer needs; • a mechanism is established for continuous monitoring of customer needs; • a mechanism is established for ensuring that customers can easily determine the status and disposition of their requests; and • enhancements arising from changing technology and customer needs are identified and their impact is managed.
Base Practices	ENG.1.BP1: Obtain customer requirements and requests. Obtain and define customer requirements and requests through direct solicitation of customer and user input. [Outcomes: 1, 4] ENG.1.BP2: Understand customer expectations. Ensure that both supplier and customer understand each requirement in the same way. Review with customers and stakeholders their requirements and requests to better understand their needs and expectations and to check the feasibility and appropriateness of their requirements. [Outcomes: 6] ENG.1.BP3: Agree on requirements. Obtain agreement across teams on the customer requirements, obtaining the appropriate sign-offs by representatives of all teams and other parties contractually bound to work to these requirements. [Outcomes: 2] ENG.1.BP4: Establish customer requirements baseline. Formalize the customer requirements and establish as a baseline for project use and monitoring against customer needs. [Outcomes: 2, 3] ENG.1.BP5: Manage customer requirement changes. Manage all changes made to the customer requirements against the customer requirements baseline to ensure enhancements resulting from changing technology and customer needs are identified and that those who are affected by the changes are able to assess the impact and risks and initiate appropriate change control and risk mitigation actions. [Outcomes: 4, 5] ENG.1.BP6: Establish customer query mechanism. Provide a means by which the customer can be aware of the status and disposition of their requirements changes. [Outcomes: 5]

Based on the capability level of the individual processes, the Maturity Model defines – in chapter 6 - the rules to decide on the maturity level (ML) for the organization as a whole.

- Maturity Level 1 (CAT-D) is reached when at least the ALPHA level is reached for the following processes: Project management and Software Implementation from the ISO/IEC 29110 software Basic profile and, Software Testing, Quality Assurance, Configuration Management, Problem resolution management, and Resource management of the ECSS standards;
- Maturity Level 2 (CAT-C) is reached – on top of the previous one - when ALPHA is reached for additional processes: Requirements Elicitation, Verification, Safety and Dependability Assurance of the ECSS standard. Finally,
- Maturity Level 3 (Organizational) is reached when GAMMA is achieved for Project Management and Software Implementation and BETA for the rest of the processes including Organizational Management and Process Management from the ISO/IEC 29110-4-2.

These maturity levels are expected to be representative of the organizations' capability to develop aerospace software with different criticality levels as defined in the ECSS standard: ML1 entities are expected to be able to develop software with criticality D and ML2 with criticality level C. These criticality levels are defined in the ECSS standard ECSS-Q-ST-80; criticality levels are assigned considering the consequences at the system level of the non-execution or incorrect execution of the software: level C is assigned when consequences are major (Major mission degradation without loss of mission), and D when they are minor or negligible.

4. Incorporation and tailoring of ECSS requirements

The current version of the Maturity Model includes two work-in-progress annexes that are expected to provide the traceability between the model's base practices and work products and the software development requirements defined in the ECSS standards.

This research aims to complete the tailoring of the model by linking the activities and tasks derived from the analysis of the PAM with specific ECSS requirements in ECSS-E-ST-40C, which is the standard that governs software development activities. This work is aligned with the spirit of the model, which supports the tailoring and further specification of the model by the incorporation of requirements from other standards. From this perspective, the Maturity model establishes a framework where specific software development requirements can be plugged and put into a more general, common process context. The result of this analysis and integration is a detailed process model that helps VSEs ensure full understanding and compliance with specific aerospace standards like ECSS, Galileo Software Standard (GSWS-G), and others. The extension capabilities of the model would also allow VSEs to allocate their internal procedures on the process map defined by the Maturity model.

The first step of this activity is to derive a detailed process model from the Maturity model's base practices and work products. The resulting process model was represented using a semiformal process representation language. The use of a formal model representation notation is needed to accommodate additional extensions from ECSS requirements. The selected language to represent the process was SPEM (Software Process Engineering Modelling). With ISO/IEC 24744:2014, "Software Engineering – Metamodel for Development Methodologies" (SEMDM), SPEM offers a metamodel and a conceptual framework compliant with the Object Management Group (OMG) Meta Object Facility™ (MOF) to represent software development processes (Henderson-Sellers and Gonzalez-Perez 2008). Its scope is *"limited to the minimal elements needed to define any process and accommodate a large range of development methods and processes of different styles, cultural backgrounds, levels of formalism, life cycle models and communities." (SPEM, p. 2)*. The basic building blocks in SPEM are the method *content elements* that correspond to tasks, work products, roles, categories, and guidelines (checklists, estimation methods, or instructions on how to use the tools) that relate to each other: e.g. roles participate in tasks to generate work products. This language also supports the definition of capability patterns, which are process fragments that can be combined into delivery patterns (i.e. whole process from start to finish) and can be reused in different process instances using variability mechanism.

The development of the process framework has been completed using SPEM and the Eclipse Process Framework (EPF) tool developed by the Eclipse Foundation, using as inputs the maturity model's PAM and the ECSS standards for software development, configuration management, and software quality: ECSS-E-ST-40C and ECSS-M-ST-40C and ECSS-Q-ST-80C. This approach is in line with the purpose of the maturity model, which states that: *"an organization would adopt the standard and supplement it with appropriate procedures, practices, tools, and policies. A software project of the organization would typically conform to the organization's processes rather than conform directly to this standard." (p. 8)*. The resulting SPEM-based representation covers the whole set of processes and derived activities, tasks, and work products from the PAM modeled as independent capability patterns to allow their combination and reuse in different process instances at projects (see Fig. 1 for the PM process).

Each process is represented through a work breakdown structure and a graphical representation of the activities and tasks (see Fig. 2). The organization of activities and task in the form of a work breakdown structure that can be exported to project management tools ensures that different projects may benefit of a common plan that can be further tailored at the project level.

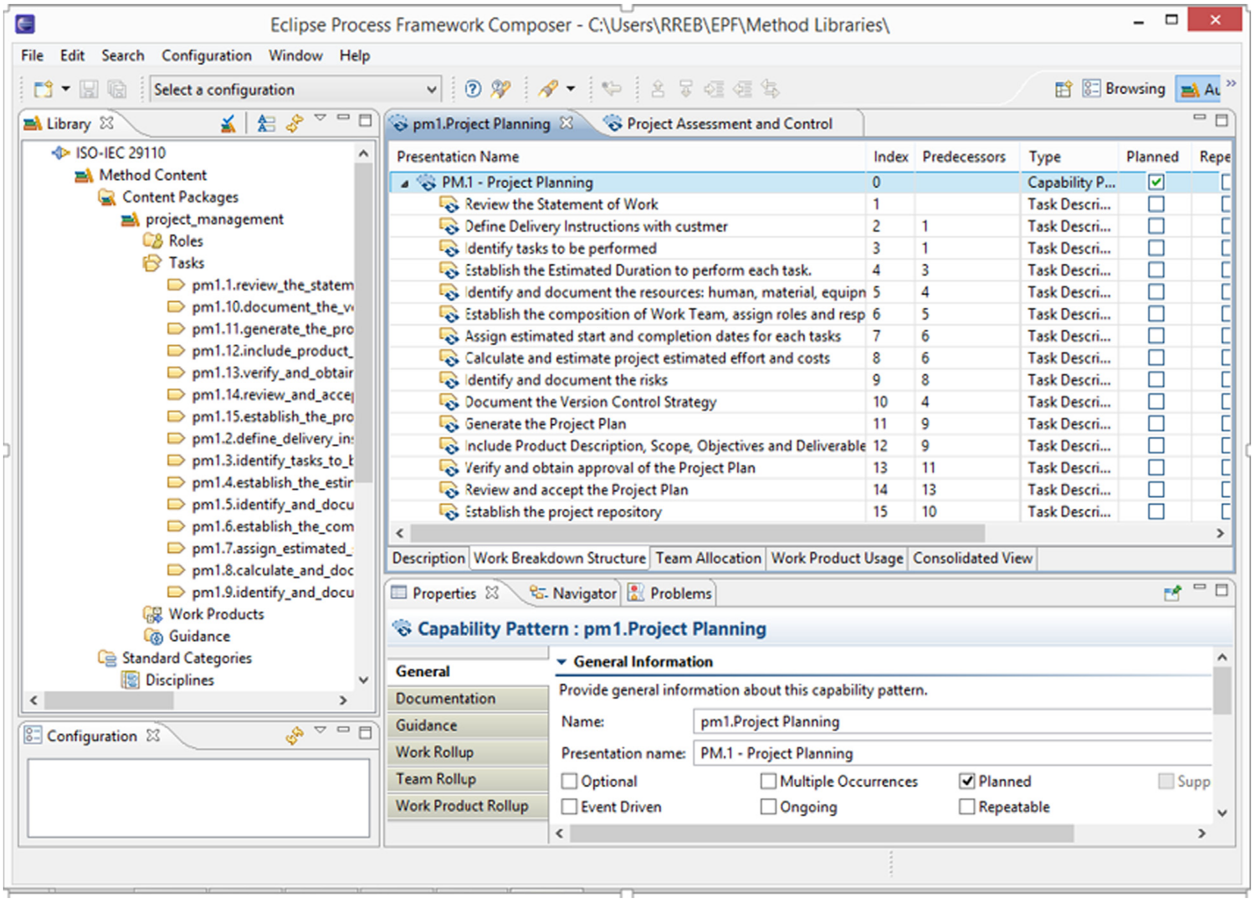

Fig. 1. SPEM-based Process Model for the PM process.

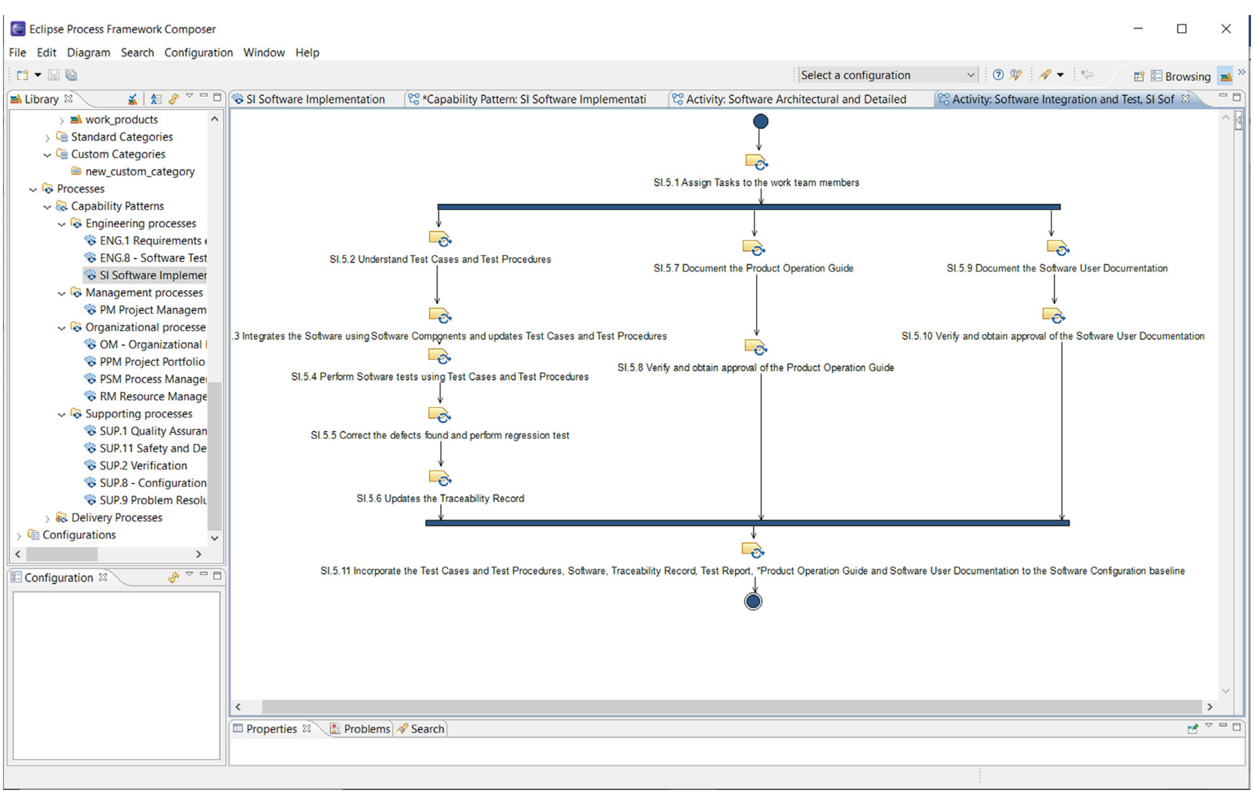

Fig. 2. Graphical representation for the SW Integration task.

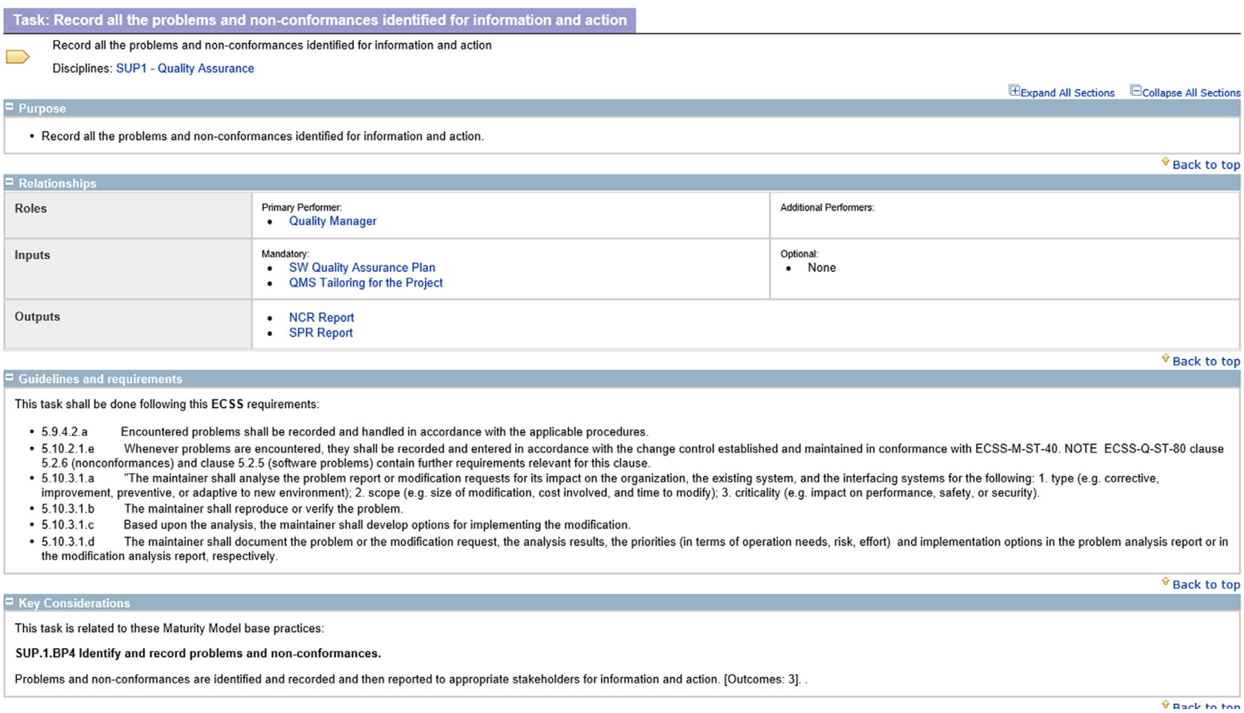

Fig. 3. Integrated view of Process Model with ECSS requirements.

ECSS standards for software engineering (ECSS-E-ST-40C) define 228 requirements that have been integrated into the process model. The incorporation of ECSS requirements is made by attaching them as guidelines to tasks and work products or by adding them as new activities or tasks. The linked graph provides a consolidated, single point-of access view of all the applicable requirements related to a specific process, activity, task, or work product, regardless of their provenance: ECSS, ISO/IEC 29110, or SPICE for Space (see Fig. 3) The incorporation of the ECSS requirements into the VSE MM for the Space Domain was made with the help of the ATLAS.ti tool, which provides the capability of tagging documents' fragments using a defined set of codes. These codes were grouped into Process, Tasks (typically derived from Base Practices), Guidelines, and Work products (according to the SPEM artifacts). ECSS requirements were tagged using these codes and later incorporated into the SPEM model in the different contexts where they are relevant.

Planned next steps in this initiative include the incorporation to the process model of additional requirements from aerospace software development standards (in particular ECSS-Q-ST-80C Rev.1 and Galileo Software Standard for Ground, GSWG), intending to facilitate VSEs a complete view of applicable process requirements contextualized with a well-defined set of activities, tasks and work products derived from the VSE Maturity Model's PAM.

5. Conclusions

The publication of the software profiles of the ISO/IEC 29110 can be considered a major milestone in the development of standard software processes suitable for VSEs. The incorporation of this standard into a maturity model for entities working in the aerospace industry is also a significant step to establish the ground for improvement programs and assessments, solving the traditional difficulties to apply more complex models like CMMI® or SPICE. The definition of this maturity model still offers opportunities for improvement: in particular, its integration with specific process requirements from software development standards like those defined in ECSS or ISO/IEC/IEEE 12207.

This research proposes an integration that results in a detailed, exhaustive process model where requirements are put into the context of a process framework to help practitioners gain a full understanding of the constraints that must guide software development efforts. Among other benefits, the resulting process model offers a unified view on all the process requirements defined in different sources and ensures that requirements are not overlooked regardless of their distribution through different documents and standards. These advantages are in part due to the use of a standard-based process representation model that supports the tailoring and extension of the model and the incorporation of additional aerospace software development standards, guidelines, and corporate procedures. Through the use of SPEM constructs, the proposed model can be easily expanded to ensure compliance with the regulatory framework applicable in different programs.

The resulting model includes the representation of thirteen processes - organized in four different groups (engineering, management, support and organizational) -, sixty-six sample work products traced to the ECSS document types – and one hundred eighty seven tasks that can be reused and tailored in the context of specific projects. The use of graphical representations for each process and activities helps ensure a better understanding of the individual process and their relationships and dependencies.

The benefits of the proposed approach include the availability of a single point of access to all the relevant information that is needed when planning and executing software projects and a better comprehension of the process requirements to be applied. The proposed contextualization of the ECSS requirements with the process defined in the VSE Maturity Model PAM ensures that VSEs working in the Aerospace industry have a single and easier-to-understand view of their stakeholders' expectations regarding software process implementation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alvarez, Jordi J., Hurtado, Julio A., 2014. Implementing the software requirements engineering practices of the ISO 29110-5-1-1 standard with the unified process. In: Computing Colombian Conference (9CCC), 2014 9th. IEEE, pp. 175–183.

Basri, Shuib, O'Connor, Rory V., 2010. Understanding the Perception of Very Small Software Companies towards the Adoption of Process Standards. In: Riel, A., O'Connor, Rory, Tichkiewitch, S., Messnarz, Richard, (Eds.), European Conference on Software Process Improvement: Systems, Software and Services Process Improvement. EuroSPI 2010. Communications in Computer and Information Science, vol. 99. Springer, Berlin, Heidelberg, pp. 153–164.

Castillón-Barraza, A., González-Angeles, A., Lara-Chavez, F., Mendoza-Muñoz, I., 2018. Tools to measure the technological capabilities of the aerospace industry. J. Ind. Eng. Manage. 11 (4), 769–775.

Cerezo-Narváez, A., et al. 2019. Standardizing innovation management: An opportunity for SMEs in the aerospace industry. Processes 7(5), art. no. 282.

Diniz, G.H., Ambrosio, A.M., Lahoz, C., 2019. Critical Software Processes Tailoring and Very Small Entities (VSE): A Literature Review. Int. J. Adv. Eng. Res. Sci. 6 (11).

Eito-Brun, Ricardo, Sicilia, Miguel-Angel, 2017. An innovation activity model for Very Small Entities in the software sector: an empirical study. R & D Manage. 47 (5), E13–E25. https://doi.org/10.1111/radm.12226.

ESA et al., 2018. Maturity Model for VSEs in the Space Domain. 98 p.

Felderer, M., Ramler, R., 2016. Risk orientation in software testing processes of small and medium enterprises: an exploratory and comparative study. Software Qual. J. 24 (3), 519–548.

Galvan, Sergio, Mora, Manuel, O'Connor, Rory V., Acosta, Francisco, Alvarez, Francisco, 2015. A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process. Procedia Comput. Sci. 64 (4), 188–195.

Galván, Sergio, Mora, Manuel, O'Connor, Rory V., 2018. A Means-Ends Design of SCRUM+: an agile-disciplined balanced SCRUM enhanced with the ISO/IEC 29110 Standard. In: Mejia, J. et al., (Eds.), Trends and Applications in Software Engineering. CIMPS 2017. Springer, Berlín, Heidelberg.

Henderson-Sellers, Brian, Gonzalez-Perez, Cesar, 2008. Standardizing Methodology Metamodeling and Notation: An ISO Exemplar. In: International United Information Systems Conference. Springer, Berlin, Heidelberg, pp. 1–12.

Hussein, A., Cheng, K., 2016. Development of the Supply Chain Oriented Quality Assurance System for Aerospace Manufacturing SMEs and Its Implementation Perspectives. Chin. J. Mech. Eng. 29 (6), 1067–1073. https://doi.org/10.3901/CJME.2016.0907.108.

INCOSE, 2015. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, fourth ed. Wiley.

Laporte, C.Y. et al., 2018. Applying Software Engineering Standards in Very Small Entities-From Startups to Grownups. IEEE Softw. 35 (1), 99–103.

Laporte, Claude Y., Alexandre, Simon, Renault, Alan, 2008. Developing international standards for very small enterprises. Computer 41 (3), 98–101.

Laporte, Claude Y., Ptack, Ken, Fanmuy, Gauthier, 2012. The Development of Systems Engineering International Standards and Support Tools for Very Small Enterprises. In: 22nd Annual International Symposium of the International Council on Systems Engineering, INCOSE 2012 and the 8th Biennial European Systems Engineering Conference, vol. 22, no. 1, pp. 1460–1487.

Laporte, C.Y., Mejia Miranda, J.M., 2020. Delivering Software and Systems Engineering Standards for Small Teams - Feedback from Very Small Entities, their customers, auditors and academia on ISO/IEC 29110. IEEE Comput. 53(8), 79–83.

Laporte, Claude Y., O'Connor, Rory V., Fanmuy, Gauthier, 2013. International systems and software engineering standards for very small entities. CrossTalk. 26 (3), 28–33.

Laporte, Claude Y., O'Connor, Rory V., 2016. Systems and Software Engineering Standards for Very Small Entities: Accomplishments and Overview. Computer 49 (8), 84–87.

Mas, Antonia, Mesquida, Antoni Lluis, 2013. Software project management in small and very small entities. In: Information Systems and Technologies (CISTI), 2013 8th Iberian Conference on, 2013. IEEE, pp. 1–6.

Mas, Antonia, Mesquida, Antoni-Lluís, 2014. A project management improvement program according to ISO/IEC 29110 and PMBOK®. J. Softw. Evol. Process. 26 (9), 846–854.

O'Connor, Rory V., Laporte, Claude Y., 2011. Deploying Lifecycle Profiles for Very Small Entities: An Early Stage Industry View. In: O'Connor, Rory, Rout, Terry, McCaffery, F., Dorling, Alec (Eds.), International Conference on Software Process Improvement and Capability Determination: Proceedings of the 11th International Conference, SPICE 2011, Dublin, Ireland, May 30 – June 1, 2011. Springer, Berlin, Heidelberg, pp. 227–230.

O'Connor, Rory V., Laporte, Claude Y., 2010. Towards the provision of assistance for very small entities in deploying software lifecycle standards. In: ACM International Conference Proceeding Series, pp. 4–7.

OCDE, 2019. OCDE Development, Small and Medium Enterprise Outlook. ECD Publishing, Paris, https://doi.org/10.1787/34907e9c-en.

Project Management Institute, 2017. A Guide to the Project Management Body of Knowledge (PMBOK® Guide), sixth ed. Project Management Institute, Newtown Square, Pa.

Raninen, A., Ahonen, J.J., Sihvonen, H.-M., Savolainen, P., Beecham, S., 2013. LAPPI: A light-weight technique to practical process modeling and improvement target identification. J. Softw. Evol. Process. 25 (9), 915–933.

Reed, F.M., Walsh, K., 2002. Enhancing technological capability through supplier development: A study of the U.K. aerospace industry. IEEE Trans. Eng. Manage. 49 (3), 231–242.

Ribaud, Vincent, Saliou, Philippe, 2010. Process assessment issues of the ISO/IEC 29110 emerging standard. In: ACM International Conference Proceeding Series, pp. 24–27.

Rodriguez´-Dapena, P., Buitrago-Botero, M.A. 2015. How to Certify the Very Small Entity Software Processes Using ISO/IEC 29110. In: Terry Rout, Rory V. O'Connor, Alec Dorling (Eds.), Software Process Improvement and Capability Determination: Proceedings 15th International Conference, SPICE 2015. Communications in Computer and Information Science, vol. 526. Springer, Berlin, Heidelberg, pp. 245–252

Rodríguez-Dapena, P., Lohier, P., 2017. How small organizations could participate in Space projects. In: 2017 IEEE International Workshop on Metrology for Aerospace (MetroAeroSpace), Padua, 2017, pp. 11–15. doi: 10.1109/MetroAeroSpace.2017.7999557.

Sánchez-Gordon,´ Mary-Luz, O'Connor, Rory V., 2016. ''Understanding the gap between software process practices and actual practice in very small companies. Software Qual. J. 24(3), 549–570.

Takeuchi, Motoko, Kohtake, Naohiko, Shirasaka, Seiko, Koishi, Yumi, Shioya, Kazunori, 2014. Report on an assessment experience based on ISO/IEC 29110. J. Software: Evol. Process 26 (3), 306–312.

Further Reading

ECSS-Q-HB-80-02 Part 2A, 2010. Space product assurance—Software process assessment and improvement – Part 2: Assessor instrument. ESA-ESTEC Requirements and Standards Division, 8 October 2010.

European Commission, 2015. User guide to the SME Definition. Luxembourg: Publications Office of the European Union.

ISO/IEC, 2008. Systems and software engineering – Software life cycle processes. ISO/IEC 12207:2008.

ISO/IEC, 2018. Software engineering – Lifecycle profiles for Very Small Entities (VSEs) – Part 4-1: Profile specifications: Generic profile group. ISO/IEC 29110-4-1:2018.

ISO/IEC, 2011. Software engineering – Lifecycle profiles for Very Small Entities (VSEs) – Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile. ISO/IEC TR 29110-5-1-2:2011.

ISO/IEC, 2014. Software engineering – Metamodel for development methodologies. ISO/IEC 24744:2014.

ISO/IEC, 2015. Systems and software engineering – Lifecycle profiles for Very Small Entities (VSEs) – Part 5-6-1: Systems engineering – Management and engineering guide: Generic profile group: Entry profile. ISO/IEC TR 29110-5-6-1:2015.

OMG, 2008. Software & Systems Process Engineering Metamodel™ Specification (SPEM™) Version 2.0. Available at: http://doc.omg.org/formal/08-04-01.pdf.