Received: 11 March 2019 | Revised: 6 June 2019 | Accepted: 15 June 2019
DOI: 10.1002/smr.2212

WILEY Journal of Software: Evolution and Process

RESEARCH ARTICLE - METHODOLOGY

Quantitative evaluation of tailoring within SPICE-compliant security-informed safety-oriented process lines

Barbara Gallina

School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden

Correspondence
Barbara Gallina, School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden.
Email: barbara.gallina@mdh.se

Funding information
EU ECSEL AMASS project, Grant/Award Number: 692474

Abstract

In the context of SPICE-compliant and (security-informed) safety processes, efficient process tailoring is necessary due to the increasing proliferation of requirements, which, if not systematised, may become an unmanageable cognitive overload leading to process degradation instead of improvement. Recently, security-informed safety-oriented process line engineering (SiSoPLE) has been proposed as a sound solution to systematise common and variable process elements in the context of security-informed safety-oriented processes described within security as well as safety-related standards. SiSoPLE represents an extension of safety-oriented process line engineering (SoPLE). The gain of the application of SoPLE in terms of efficient tailoring via reuse was measured in a previous work, where the GQM⁺ Strategies model, an extension of the goal/question/metric (GQM) paradigm, was adopted to develop a measurement model for achieving quantitative evidence. In this paper, we develop further our previously proposed measurement model to achieve quantitative evidence regarding the benefits of using process line engineering extended to SPICE-compliant security-informed safety processes. We then apply our extended GQM⁺ Strategies model on a SPICE for space-compliant SiSoPL to illustrate and assess its usefulness. Finally, we discuss our findings and provide our perspectives on quantitative evaluation of tailoring in the context of critical-systems engineering.

KEYWORDS

change management, ECSS normative scheme, GQM⁺ Strategies, security-informed safety-oriented process line engineering (SiSoPLE), software process improvement (SPI), SPICE for Space

1 | INTRODUCTION

Management, engineering, quality, and capability-maturity–focused, descriptive as well as prescriptive, standards impose an increasing and considerably overlapping number of requirements on development processes. In the context of SPICE-compliant and security-informed safety life cycles for software engineering, process improvement via security-informed safety-oriented process line engineering (SiSoPLE) seems to be feasible. SiSoPLE represents a solution to manage change/efficient tailoring in the context of security-informed safety-critical systems development processes. Change management is a key component in process improvement infrastructures.[1] Implementation of SiSoPLs in an organisation, however, needs to be planned and justified. SiSoPLE represents an extension of safety-oriented process line engineering (SoPLE). The gain of the application of SoPLE in terms of efficient tailoring via reuse was measured in a previous work,[2] where the GQM⁺ Strategies model, shortened GQMPS, an extension of the goal/question/metric (GQM) paradigm, was adopted to develop a measurement model for achieving quantitative evidence. Metrics to measure the effectiveness of SiSoPLE have been so far neglected. A lack of metrics can impede their adoption. Organisations considering adoption of SiSoPLs are faced with the upfront questions regarding the selection of the right processes for conversion to derive the maximum benefits in the shortest time frame. Resources can be allocated to an endeavour only in the presence of objective justification of the economic benefits. To provide such justification, we need an appropriate measurement methodology. In this paper, we develop further our previously proposed model to achieve quantitative evidence regarding the benefits of using process line engineering extended to SPICE-compliant security-informed safety processes for software engineering. We then apply our extended GQM⁺ Strategies model on a SPICE for Space–compliant SiSoPL to illustrate and assess its usefulness.

The rest of this paper is organised as follows. In Section 2, we provide background information. In Section 3, we present our GQM+ model for SPICE-compliant SiSoPLE evaluation. In Section 4, we evaluate the benefits of a SPICE for Space (S4S)–compliant SiSoPL and discuss our findings including threats to validity. In Section 5, we discuss the synergies between the SPI Manifesto and SPICE-compliant SiSoPLE-targeted GQMPS. In Section 6, we provide more general perspectives on quantitative evaluation of tailoring in the context of critical-systems engineering. In Section 7, we discuss the related work. Finally, in Section 8, we conclude the paper and sketch future research directions.

2 | BACKGROUND

2.1 | ECSS standards: focus on software development

The ECSS normative scheme for space programmes and projects[3] comprises a root standard (ECSS-S-ST-00[4]) from which three major branches develop, namely, project management (whose standards are denoted with an M), engineering (whose standards are denoted with an E), and product assurance (whose standards are denoted with an Q). In addition to norms, guidelines are provided within handbooks (denoted with HB).

In this paper, we focus on the software-related standards and handbooks depicted in Figure 1. To make the paper self-contained, in what follows, we briefly recall essential information from these standards/handbooks.

- ECSS-S-ST-00[4]: The purpose of this standard is to give an introduction to the three branches of applicability and to the disciplines covered by the set of ECSS standards and the processes involved in generating and using ECSS standards. ECSS-S-ST-00C also states that the ECSS system can be adapted to specific domains of application by use of tailoring activities, which can be driven by, eg, cost, risk, and maturity capability.
- ECSS-M-ST-10[5]: The purpose of this standard is to provide key elements of project planning and implementation. In this paper, we mention the key element represented by the project breakdown structures, which break the project down into manageable elements. One of these structures is the work breakdown structure (WBS), which provides a framework for managing cost, schedule, and technical content. ECSS-M-ST-10 also defines a seven-phase-space project life cycle (PLC), whose phases are denoted with alphabetical letters. Phase B (preliminary definition) is the one in focus in this paper.
- ECSS-E-ST-40C[6]: The purpose of this standard is to cover all aspects of space system software engineering including requirements definition, design, production, verification and validation, transfer, operations, and maintenance. Tailoring rules are provided in a specific annex, *Annex R (normative)*, to enable manufacturers and suppliers to customise their engineering processes. The tailoring is conducted based on the software criticality, which ranges from A to D. Different customisations, performed by the different customers, can be seen as different single processes within a family of processes.

In this section, we limit our attention to a very small portion of ECSS-E-ST-40C, Clause 5.2 (Software related system requirement process). According to Annex R, all requirements related to Clause 5.2 are applicable for all four criticality levels. We recall that this process is constituted of a series of activities. Each of these activities consists of various tasks, which in turn contain various steps.

- Overview (5.2.1)
- Software-related system requirements analysis (5.2.2)
  - Specification of system requirements allocated to software (5.2.2.1)
    - The customer shall derive system requirements allocated to software from an analysis of the specific intended use of the system and from the results of the safety and dependability analysis. (5.2.2.1a)
  - Identification of observability requirements (5.2.2.2)

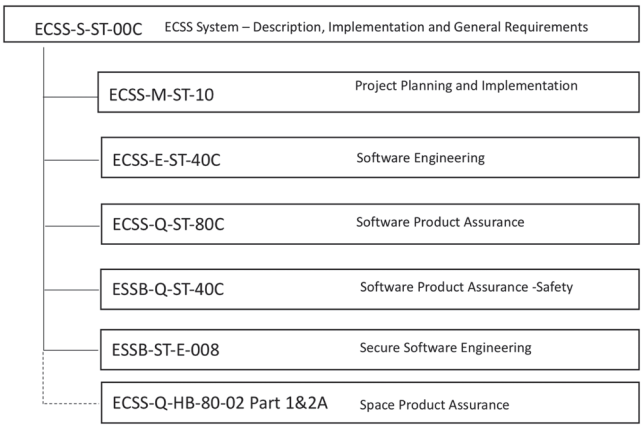

FIGURE 1 ECSS scheme

* The customer shall specify all software observability requirements to monitor the software behaviour and to facilitate the system integration and failure investigation. (5.2.2.2a)

- Software-related system verification (5.2.3)
- Software-related system integration and control (5.2.4)
- System requirements review (5.2.5)

In this paper, specifically, we focus on Clause 5.2.2.1. This limited process portion exemplifies what is typically required in terms of process engineering, ie, complying with the requirements while tailoring.

- ESSB-ST-E-008[7]: The purpose of this internal standard is to provide a collection of secure software engineering processes intended to incorporate security engineering practices within an ESA (European Space Agency) software development project and especially through use of a Secure Software Development Lifecycle (SSDLC). The necessity of developing security standards for ESA projects was discussed by Fischer et al.[8] It is worth to note that ESSB-ST-E-008 does not overwrite existing standards but instead it incrementally adds requirements specifically related to security. The tailoring of ESSB-ST-E-008 is conducted based on the software criticality, as well as the Evaluation Assurance Levels (EAL) and the strength of function. The definition of EAL, from EAL1 (functionally tested - less strict) to EAL7 (formally verified design and tested—most strict) is provided within the Common Criteria (CC),[9] which establishes a security assurance scale. In this paper, we limit our attention to clause 5.2, which constitutes the point of alignment with ECSS-E-ST-40C, Clause 5.2. ESSB-ST-E-008 increments ECSS-E-ST-40C, clause 5.2.2.1, by adding three new requirements, which are as follows:

  The customer shall use an available security requirements catalogue to refine the security requirements profile. (5.2.2.1.b)
  The customer shall use security assurance requirements to endorse the project specific security requirements and to assure to meet them. (5.2.2.1.c)
  The customer shall document the resulting security requirements profile, and the security strength of function requirements inside the software system specification (SSS). The customer shall provide a traceability matrix to trace the security requirements to the risks identified in the cyber-security risk assessment. (5.2.2.1.d)

  In addition, it also refines requirement 5.2.2.1.a by stating that the analysis should also consider the security requirements profile.
- ECSS-Q-ST-40C[10]: The purpose of this standard is to define the safety programme and the safety technical requirements aiming at protecting humans and the environment from hazards associated with European space systems. Clause 6.2 (safety requirements identification and traceability) states that "Safety requirements shall be identified and traced from the system level into the design and then allocated to the lower levels." This clause constitutes the point of alignment with ECSS-E-ST-40C. ECSS-Q-ST-40C defines the severity categories, necessary for safety risk assessment. It also defines the criticality of the functions and the criticality category assignment for software products versus function criticality. It does provide guidance for tailoring safety requirements based on their applicability, given the software product under planning/engineering.
- ECSS-Q-ST-80C[11]: The purpose of this standard is to define a set of software product assurance requirements to be used for the development and maintenance of software for space systems. Clause 6.3 of this standard (software-related system requirements process) constitutes the point of alignment with ECSS-E-ST-40C.
- ECSS-Q-HB-80-02 Part1[12] and Part 2[12]: The purpose of this two-part handbook is to customise SPICE (Software Process Capability dEtermination) for Space (S4S). S4S addresses the software process capability maturity in space. To determine maturity, the process assessment model selects the process reference model and augments it with indicators. These indicators are used to identify if the process outcomes (PO), the result of the execution of the process, and the process attribute outcomes (PA), the result of the achievement of a specific process attribute, are present. Base practices (BP) (activity-oriented PAs) must be evaluated to establish the capability of the process to be achieved. In this paper, we limit our attention to one BP, specifically BP-1 (Specify software requirements), for the process (ENG.4). BP-1 mainly restates ECSS-Q-ST-40C-requirements related to 5.2.2.1.a and refers to ECSS-Q-ST-80C. Thus, it mainly makes sure that a SPICE assessor would ensure the quality required by ECSS.

2.2 | SoPLE, SiSoPLE, and SPICE-compliant SiSoPLE

SoPLE was introduced by Gallina et al[13,14] and applied in various domains (see, for instance, Gallina et al[15] for its usage in the context of tool qualification recertification and Varkoi et al[16] for its usage in the nuclear domain). SoPLE consists of a two-phase method. The first phase is aimed at engineering the domain from a process perspective, ie, identifying and systematising process-related commonalities and variabilities in order to concurrently engineer a set of processes (SoPL). The second phase is aimed at deriving single safety-oriented processes via selection and composition of commonalities and variabilities. To deal with security-informed safety processes, SoPLE was extended. In particular, Gallina et al[17] introduced SiSoPLE, which consists of process line engineering focused on the alignment of safety and security aspects for certification purposes. As part of the domain engineering, SiSoPLE requires the systematisation of the commonality within the terminological framework pertaining to security-informed safety (SiS). The potential usefulness of SiSoPLE was illustrated in the avionics[17] and automotive[18] domains. In a

TABLE 1 Potential benefits of process-related commonality at different stages

Phase	Benefit	Rational
Strategy	Enable faster variant time to certificate	The common portion of the process design already built, only the unique portion needs to be designed
Strategy	Enter niche way of working	Design un-forecast variants on top of the common skeleton enables the company to recognise and enter ways of working as they appear
Design	Shared engineering cost (intended commonality)	Reduced engineering effort required for later variants
Design	Reuse of already designed capability patterns and processes (unintended commonality)	Design effort does not need to be repeated
Manufacture	Shared tooling	Tooling cost, if any, can be spread over more processes
Manufacture	Learning curve benefits	Fewer hours/unit required
Assessment, Compliance and Delivery	Reduced assessment, compliance checking and delivery time	Learning in assessment, compliance procedures for later variants
Assessment, Compliance and Delivery	Shared compliance checking equipment	Compliance checking equipment can be spread over more processes
Assessment, Compliance and Delivery	Reduced external assessment, certification	Reuse of type certificates or regulatory approval

similar manner, process line engineering can be extended to embrace different types of standards: quality and process improvement standards. For instance, in this paper, a SPICE-compliant SiSoPLE consists of process line engineering focused on the alignment of safety, security, and software process improvement aspects for certification.

From a tooling perspective, SoPLE, SiSoPLE, and SPICE&SiSoPLE can be supported by the integration of Eclipse Process Framework (EPF) Composer,* (recently migrated to Eclipse Neon 4.6.3[19]), and Base Variability Resolution (BVR) Tool,† where the EPF Composer is used to model the base process and its related library, while the BVR Tool is used to model (VSpec), resolve (Resolution), and realise (Realization) the variability. More precisely, VSpec permits users to model the variability in a feature diagram-like fashion. Resolution permits users to configure (make choices at variation points, where desired variants can be selected) their process. Finally, Realization permits users to bind the conceptual representation of the variable elements with the concrete elements in the base model. The integration of EPF Composer and BVR Tool is described in more details by Javed et al.[20] Here, we want to point out that the integration of EPF Composer and BVR Tool offer, via the VSpec, Resolution, and Realization models, a coordination/negotiation ground among the different teams within a company. Teams, when not a unique process engineer is nominated, have the possibility to propose changes but also see the impact (eg, constraints violation and potential commonality reduction) that their changes may entail and take actions to coordinate effectively with other teams and thus avoid commonality erosion.

In the literature regarding product line engineering or platform engineering, potential benefits of the commonality are highlighted. Cameron et al[21] summarise those benefits in a tabular format. In what follows, specifically in Table 1, we provide a semantic translation of those benefits in the context of process line engineering, and we limit our attention to four phases: strategy, design, manufacture, and assessment.

2.3 | Reuse-related metrics

Berger et al[22] define several metrics that provide different perspectives for assessing the suitability for setting up product lines. Among such metrics, in this subsection, we recall Size of Commonality (SoC) and Product-related Reusability (PrR).

1. *SoC* measures the number of reusable/identical components in a product line. For evaluating a given set of similar products constituted of n products, each product pi with $1 \leq i \leq n, n \geq 2$ is decomposed into a set Cpi of m so-called reasonable atomic pieces, where each piece is denoted cj with $1 \leq j \leq m, m \geq 1$. To perform a decomposition, all components/atomic pieces must be identified and formally specified. An annotated, directed graph is specified for each product, which reflects the dependencies (called signatures) between all components. These dependencies can be logical or communicative. SoC is determined by comparing the component signatures. A syntactic comparison of signatures is made based on the names of the components while a semantic comparison is made from the behavioural profiles of the components which capture behavioural constraints. If the signatures are identical, the components are identical. *SoC* is computed as shown in Equation (1)(a), where p_i represents the products of the product line, i ranges from *1* to n, and C_{pi} represents the set of components of the product i.

2. *PrR* measures the extent of reusability of the common components for a specific product. *PrR* is computed as shown in Equation (1)(b).

$$(a)\ SoC = \left| \bigcap_{1}^{n} Cpi \right| \quad (b)\ PrRi = \frac{SoC}{|Cpi|}. \tag{1}$$

*https://www.eclipse.org/epf/
†https://github.com/SINTEF-9012/bvr

2.4 | GQM+ Strategies

The GQMPS model links measurement programs to higher level organisation goals and strategies.[23] GQMPS is built as an extension of the GQM paradigm, a top down approach, in which measurements are based on measurement goals.[24] The GQM paradigm consists of three levels: the conceptual level (Measurement Goal) where the objectives are defined, the operational level (Question) where the questions are made, and the quantitative level where the metrics are defined. These levels are also hierarchically organised in a pyramid structure. The apex of the pyramid is represented by a measurement goal, which specifies the purpose of measurement, the object which is being measured, the issue to be measured, and the viewpoint from which the measurement is taken. This measurement goal is refined by a set of questions that breaks down the goal into its significant elements. Each question is further refined into one or more metrics. These metrics may either be objective or subjective in nature. Further, a particular metric may be used to answer more than one question.

The GQMPS model helps organisations to align multilevel organisation goals and strategies to the measurement goals. It consists of two perspectives, the Organisational and Planning Perspective (OPP) and the Control Perspective (CP). The OPP and CP structures help incorporate dependencies among different levels of the organisation. The OPP is concerned with the organisational goals and strategies while the CP is concerned with the measurements. The structure of the OPP resembles a pyramid with the top goal of the organisation at the apex. The top goal is broken down into one or more strategies. Each strategy can be further split into one or more goals and associated strategies until the strategies cannot be further split into lower goals. The CP structure is built using the GQM paradigm. Each organisational goal is linked to a GQM structure in the CP via a measurement goal. These links enable alignment of organisational goals and strategies with measurement goals. This ensures that organisations invest resources only in meaningful and essential data collection and analysis activities.

2.5 | SoPLE-targeted GQM+ Strategies

In a previous work,[2] we developed a GQM+ Strategies model to measure the gain of the application of SoPLE in terms of efficient tailoring via reuse. Our proposed SoPLE-targeted GQM+ Strategies model is recalled in Figure 2.

The OPP structure reflects the organisational goals and strategies starting with the overall organisation goal at the top and the related strategies broken down below, reflecting the SoPLE organisation goals and their related strategies. We show the association of the top organisational goal (*G1*) to the strategy (*S3*) of "Exploit commonality of safety-oriented processes," which, as seen in Section 2.2, if significant may entail a series of benefits. The strategy *S3* is further reduced to the software development organisation (SDO) goal *G2*, that of "identify candidate SoPLs for reusability" supporting the strategy *S6*, "Engineer SoPLs." Strategy *S3* is reduced to a single SDO goal for illustration purposes, though it may be reduced to additional goals, such as productivity and quality related goals. The CP part of the model links the goal *G2* to the measurement goal *MG1*, "Assess suitability to form a SoPL." The viewpoint is that of the SoPL manager, who belongs to the SDO and has overall organisational responsibility for software development with formation of SoPLs as the object. *MG1* is progressively refined into the question *Q1* and the metrics *M1* and *M2* addressing the extent of commonality via the quantitative evaluation of the size of commonality and the product (process in this context) reusability. The higher the commonality, the stronger is the evidence for adoption and potential organisational goal achievement and entailment of the potential benefits.

3 | GQM+ STRATEGIES MODEL FOR SPICE-COMPLIANT SISOPLE EVALUATION

In this section, we extend our previously proposed SoPLE-targeted GQM+ Strategies Model for SPICE-compliant SiSoPLE with a focus on efficient tailoring via reusability of SPICE-compliant security-informed safety-oriented processes. A consolidated view of the extended SPICE-compliant SiSoPLE-targeted GQMPS model is shown in Figure 3. Syntactically, the extension is minimal. Figure 3, indeed, does not expose a considerable syntactical difference from Figure 2. From a semantic perspective, however, this extension has the purpose to consider all drivers used for

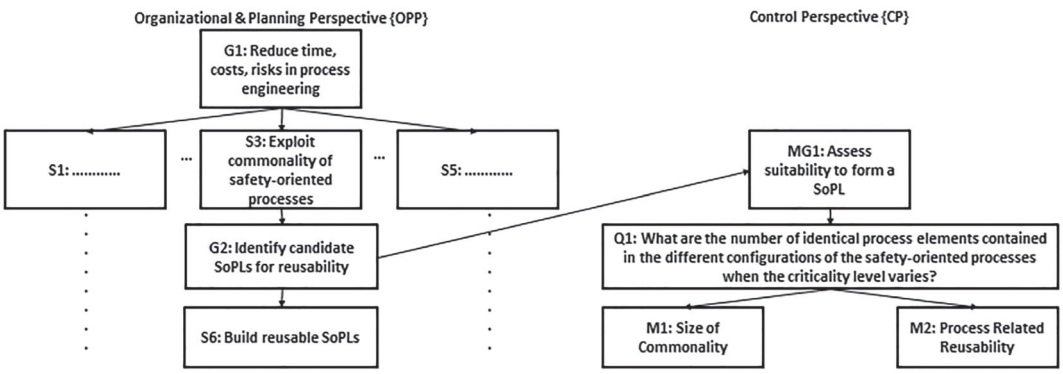

FIGURE 2 SoPLE-targeted GQM+ Strategies model

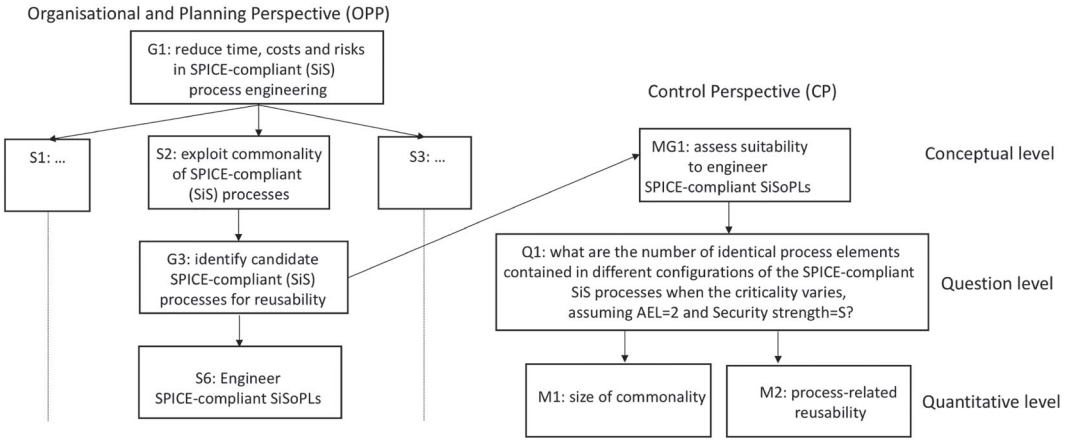

FIGURE 3 SPICE-compliant SiSoPLE-targeted GQM$^+$ Strategies model

tailoring. As the idiomatic expression "the devil is in the detail" warns and as previous studies let emerge, a strategy focused on high-level commonality exploitation might be biased during the upfront planning and blinded by the high-degree of commonality present in conceptual design. As a consequence, during the adoption, as the design progresses, changes are made that cause a continual drift away from commonality. Even if each change may be small, the net effect, the "divergence," can be great. This is why it is crucial to learn from previous studies[21] and learn from the previously identified canonical commonality strategies in order to properly tune the commonality planning towards a commonality culture, meant as ability to weigh the options in a divergence decision, ie, ability to ponder the right level of commonality knowing the potential need for customisations. Since the extended focus embracing also SPICE-compliance and security-related compliance may imply potential customisations, the purpose of the extended SPICE-compliant SiSoPLE-targeted GQMPS model is to capture the manager and process engineer attention when considering all drivers. Given this extension, the manager and the process engineer could consider evaluating the GQM-portion not only on the entire family (including SPICE and security requirements) but also on every subfamily and on every clause and subclause with well-identified responsibility for each acting role to be able to ensure a proper upfront planning.

Similar to what presented for SoPLE-targeted GQM$^+$ Strategies Model, also the SPICE-compliant SiSoPLE organisation may be concerned with goals related to the feasibility of establishing SPICE-compliant SiSoPLs (pre-SPICE-compliant SiSoPL) or goals related to assessing the effectiveness of the established SPICE-compliant SiSoPLs (post-SPICE-compliant SiSoPL). In this paper, we limit our discussion to pre-SPICE-compliant SiSoPL goals, established SPICE-compliant SiSoPLs are not in place yet.

4 | EVALUATING THE BENEFITS OF AN ECSS-S4S-COMPLIANT SISOPL

In this section, we evaluate the benefits of SPICE-compliant SiSoPLE. Our evaluation, conducted on a single illustrative case, has the purpose to explore the suitability of SPICE-compliant SiSoPLE in the space domain and the generalisability of the findings in other domains.

4.1 | ECSS-compliant S4S&SiSoPL

Based on the information recalled in Section 2.1 and in Section 2.2, in this section, we model the ECSS-S4S-compliant SiSoPL related to the *software-related system requirement process*. To model it, we use BVR/VSpec.

Figure 4 presents our S4S-compliant SiSoPL for *software-related system requirement process (5.2)*. As mentioned in Section 2.1, the activity 5.2.2 is composed of two tasks (5.2.2.1 and 5.2.2.2). Task 5.2.2.1 is further structured.

The variants of task 5.2.2.1 depend on the associated values for AEL and strength of function. This results in a certain number of reusable elements and zero or more variable elements of the software-related system requirement process (5.2) depending on the associated values. The choices in our S4S-compliant SiSoPL are the activities and tasks described in Section 2.1, the criticality levels, and the presence or absence of security concerns. One and only one of the four criticality levels is valid for any S4S-compliant SiSoPL variant. The constraints are enforced when resolving a particular S4S-compliant SiSoPL variant. For instance, in our S4S-compliant SiSoPL, constraint c1 enforces the requirement that security-related steps are required only when the values for AEL and strength of function are set.

4.2 | Applying SPICE-compliant SiSoPLE-targeted GQMPS

An application of our SPICE-compliant SiSoPLE-targeted GQMPS model is shown in Figure 5. Goals *G1* and *G2* are the AMASS consortium organisation goals[25] while *G7* is an S4S-compliant SiSoPLE organisation goal, namely, that of the Software Development Organisation (SDO)

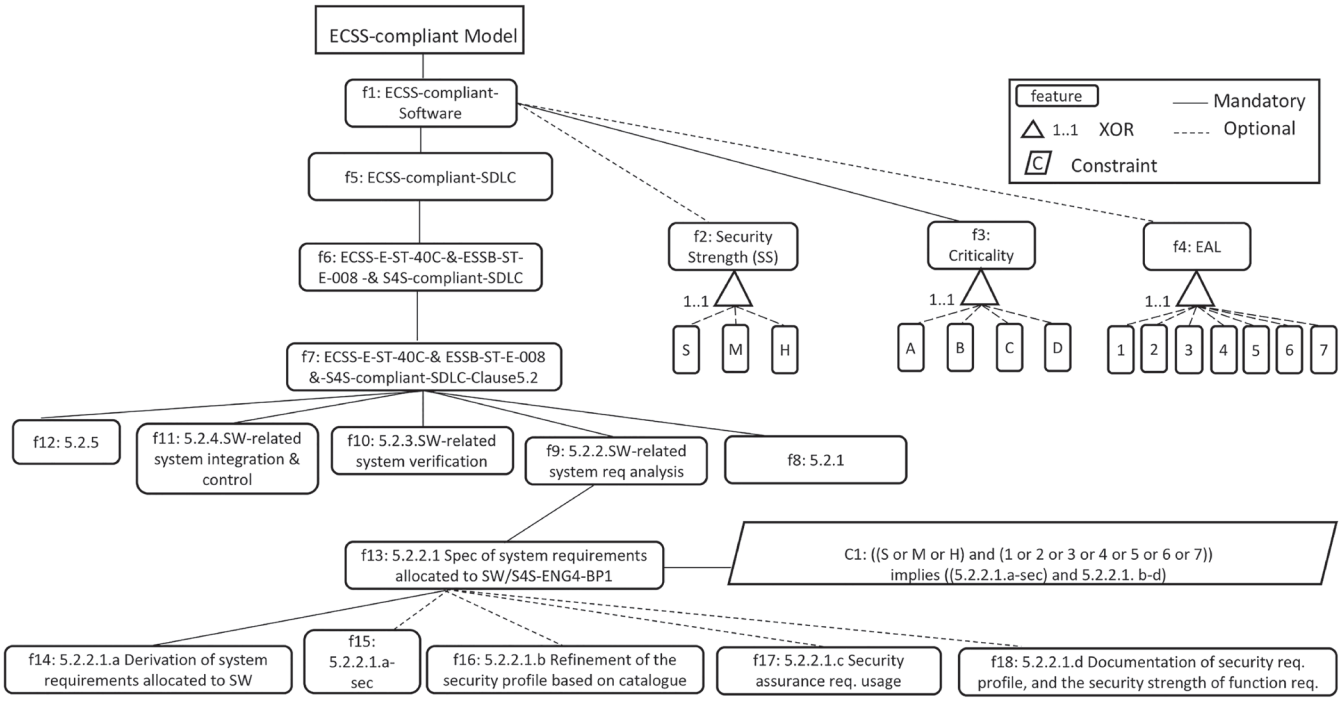

FIGURE 4 S4S-compliant SiSoPL

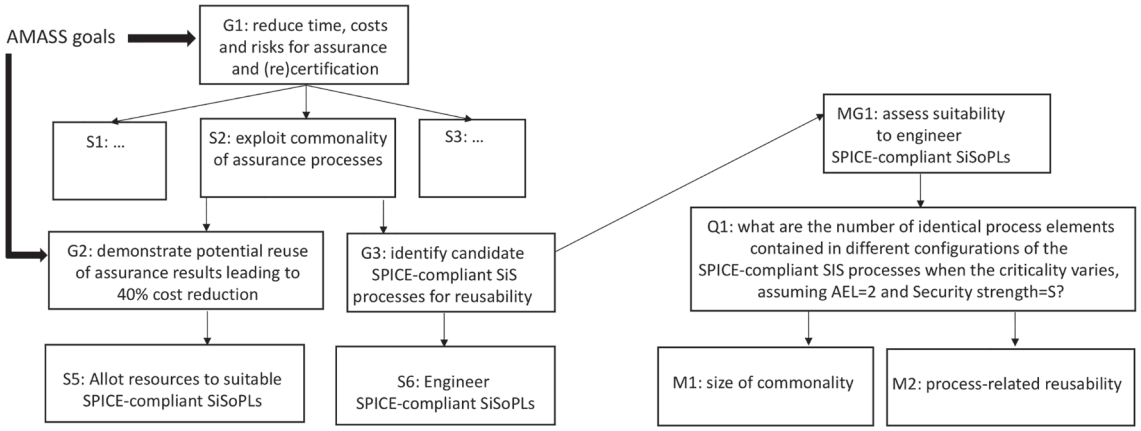

FIGURE 5 Application of our SPICE-compliant SiSoPLE-targeted GQM$^+$ Strategies model

of the space-related partner.[26] The pre-S4S-compliant SiSoPL implementation goal of the SDO is *Identify candidate S4S-compliant SiSoPLs for reusability*. The SDO goal *G7* links to the measurement goal *MG1*, *Assess suitability to form a S4S-compliant SiSoPL*. Question *Q1* refines *MG1* and addresses process element counting based on the semantic signature of the processes subject to constraints imposed by criticality levels of the processes. *Q1* pertains to the commonality of the process elements. We can logically infer that the higher the commonality, the more likely is the suitability for adopting a process line approach. We define two metrics (*M1* and *M2*), regarding the question *Q1*, which are *SoC* and *PrR* respectively interpreted for processes, ie, *SoC* measures the number of reusable/identical components in a S4S-compliant SiS process line and *PrR* measures the extent of reusability of the common components for a specific process.

Software related system requirement process constitutes our S4S-compliant SiSoPL and is treated in a similar manner to the product line referred to in Section 2.3. The activities that make up our S4S-compliant SiSoPL constitute the elements and are similar to the components in the product line. The single processes can be configured/tailored based on the applicable criticality level, the assurance evaluation level, the strength of function, and the associated constraints. Focusing on feature 6 onwards, as we have 8 common elements (see features f7, f8, f9, f10, f11, f12, f13, and f14), *SoC* computes to 8. The metric *PrR* is computed for each single process. Since AEL is typically set to 2 and since Strength of function is typically considered as standard, the tailoring space is drastically reduced. More specifically, we consider the set of processes constitute of four single processes due to the four criticality levels (with no security) plus four additional processed due to the criticality level and default values for AEL and Strength of function. Since the change of the criticality level has no impact on the software-related system requirement process, the

number of elements in the single processes is equal for all criticality levels A, B, C, and D. Specifically, the elements are 8 (f7 through f14). The number of elements in the single processes with security concerns is equal for all criticality levels A, B, C, and D. Specifically, the elements are 12 (f7 through f18). Thus, *PrRs* for single processes A, B, C, and D without security concerns is computed as 1, while *PrRs* for single processes A, B, C, and D with security concerns is computed as 0.6. For the sake of clarity, it should be stated that the variability due to SPICE requirements was not considered since negligible for this case.

4.3 | Threats to validity

In this section, we discuss the threats to validity, ie, to what extent our results are true and not biased by our subjective point of view.[27]

Construct validity refers to quality of choices about the forms of the independent and dependent variables, more precisely about the choice of the treatment. In our case, the independent variable taken into consideration are the process elements mentioned within the standards. To treat (extract and model) those elements, we have considered the definitions of process elements provided in the SPEM2.0 specification and the glossary provided by ECSS. We believe that by considering standardised definitions for extracting and modelling process elements, the extraction and modelling's reliability is sufficiently high (ie, the repeatability is high), under the assumption that those definitions are understood in a similar way via basic training. Concerning dependent variable, our treatment relies on previously accepted work within product line engineering, where metrics were elaborated.

External validity is concerned with to what extent it is possible to generalise the findings and to what extent the findings are of interest to other people outside the investigated case. The work conducted in large EU projects dealing with certification-related challenges has shed light on the, to some extent, recurring structure of standards, the recurring intradomain and cross-concern dependencies among standards, the recurring criticality-based hierarchical structure, which drives the increase of prescription in terms of objectives, activities, methods to be compliant with, and the recurring standardisation process itself. The gathered experience and inferred knowledge, confirmed through informal discussions with assessors and practitioners as well as through pieces of formal evidence by qualitative adopting family oriented process line engineering in various dependability-critical contexts, support the generalisability of the findings: A high degree of commonality is present and the adoption of SPICE-compliant-SiSoPLE has the potential to be beneficial not only in the space domain but also in other domains. This support is guaranteed under the assumption that the certification schemes do not change. Obviously, in the face of unanticipated technological progress substantially impacting on the current certification schemes, the generalisability of the current results would be invalidated. However, given the way new standards and customisations are currently introduced, this support seems to be guaranteed. As stated in ECSS-Q-HB-80-02, space software development processes are not substantially different from software processes in some other application domains (eg, defence and public transport). S4S, for instance, uses the material provided in ISO/IEC 1[28] "as is" as much as possible with minor modifications. Security standards are introduced as refinement/addition of software engineering and quality standards. As a consequence, the illustration of the alignment of standards for efficient tailoring via process line engineering, shown in this paper, can be performed in other domains and SPICE-compliant SiSoPLE can be generalised to them. This support is also guaranteed under the assumption that the creation of a common and commonly understood terminology is possible. Given the ongoing proliferation of concern-specific and domain-specific standards, numerous research works have been conducted by pools of experts (see, for instance, Baufreton et al[29] and Blanquart et al[30]) with the aim of proposing a common terminology and performing comparative studies and extract a meta-standard. The final stated aim is to move step by step towards convergence (the meta-standard). Thus, even if the current assumption does not hold when considering multiconcern standards, due to the absence of a consensus-based SIS terminological framework, it is likely that it will hold or that it could hold within niches promoting convergence.

4.4 | Discussion

In addition to the discussion related to the threats of validity, in this section, we discuss the findings related to the application of SPICE-compliant SiSoPLE-targeted GQMPS. Given the similarities of the models in terms of purpose and stage of development, our findings are analogous to those, which were discussed in our previous work.[2] The discussion covers the following three main bolded aspects.

General soundness: Despite the simplicity of the considered example, we can state that the application of SPICE-compliant SiSoPLE-targeted GQMPS for measuring SPICE-compliant SiSoPLE effectiveness is sound since it has the potential to generate objective justifications for its adoption. We have employed our SPICE-compliant SiSoPLE-targeted GQMPS model to measure SPICE-compliant SiSoPL effectiveness with reference to the AMASS consortium organisation and the space-related partner. Our rather small S4S-compliant SiSoPL and the computed metrics establish a case for the adoption of SPICE-compliant SiSoPLE to the ECSS software development process. The *SoC* determined for our S4S-compliant SiSoPL indicates a high degree of commonality despite the variability, which might be introduced by the criticality levels/AEL/Strength of function. We observe from the computed *PrRs*, the extent of reusability of the common elements for S4S-compliant SiSoPLs.

Maturity: SPICE-compliant SiSoPLE-targeted GQMPS's maturity varies with respect to the examination's angle. In terms of motivation and definition, SPICE-compliant SiSoPLE-targeted GQMPS is fairly mature. However, in terms of validation, it is still in its embryo stage. A true in-depth validation in industrial settings has not yet been carried out. We need to further examine our findings and the impact of the other process elements, which have a bearing on the safety-oriented process line such roles, tools employed, input/output work products. With respect to roles, as known, processes are nothing if not performed by people[31] and people skills are recognised to be crucial and need to be audited.

According to ECSS-M-ST-10C,[5] 5.2.1.2.e, for instance, the supplier shall demonstrate that the key personnel have the necessary qualification, skills, and experience to perform the task for which they are allocated. Clearly, the necessary skills may vary based on the criticality of the tasks to be performed. With respect to tools employed, their reusability will be constrained by the existence of evidence in favour of their usage to automate a specific critical task. A tool qualified for a certain level will need to be re-qualified to be used to automate a task of a higher criticality level. With respect to input/output work products, in the context of SPICE-compliant SiSoPLE, only their syntactical reusability is expected to be considered. Their semantic content is instead expected to be addressed via product line engineering. In addition to the space domain, other domains could/should be explored.

Tool support: SPICE-compliant SiSoPLE-targeted GQMPS is not yet tool supported. The computation of Size of Commonality and Product-related Reusability could be easily implemented as additional functionalities of BVR Tool. Similarly, an Eclipse-based editor could be developed to be able to edit SPICE-compliant SiSoPLE-targeted GQMPS applications.

5 | SPI MANIFESTO AND S4S&SISOPLE-TARGETED GQMPS: SYNERGIES

SPICE-compliant SiSoPLE-targeted GQMPS enables the measurement of SPICE-compliant SiSoPLE effectiveness and thus permits process engineers to achieve an objective justification regarding the economic benefits related to the application of SPICE-compliant SiSoPLE as software process improvement strategy. SPICE-compliant SiSoPLE represents a solution to manage change in the context of SPICE-compliant security-informed safety-critical systems development processes. Change management is a key component in process improvement infrastructures.[1]

As discussed in Section 4.4, when applying SPICE-compliant SiSoPLE, not only the breaking down of the work should be considered but all other relevant process elements, which may have an impact on SPICE-compliant SiSoPLE evaluation as well as on Software Process Improvement. Concerning roles, in addition to the technical skills (as per standards used in safety-critical software engineering), management skills[31,32] and social responsibility-related skills[33] should also be considered in order to ensure SPI-readiness attitude. Moreover, as observed by Gallina et al[34] in the Swedish context, culture may also influence the SPI readiness. Thus, technical skills alone are not sufficient and thus they cannot be analysed in isolation when reasoning about the reusability of a role from one process to a different one since cultural-, managerial-, and social responsibility-related (lack of) skills may (hinder/) foster his/her successful deployment. Existing dependencies between skills and other process elements should be specified. To this purpose, the integration between EPF Composer and BVR Tool offers an adequate tool support for SPICE-compliant SiSoPLE. Via EPF Composer, process engineers have the possibility to model a single process of the SPICE-compliant SiSoPL and then make it vary via BVR Tool (more specifically, via the VSpec, Resolution, and Realization editors). As shown in Figure 4, process engineers have also the possibility to specify constraints that limit the selection and composition of reusable process elements. Skill-related constraints could/should be specified in order to properly constrain the role-reusability based on contextual information.

6 | PERSPECTIVES ON QUANTITATIVE EVALUATION OF PROCESS/PRODUCT/ASSURANCE CASE TAILORING

In critical-product line engineering, as discussed by Gallina,[35] changes in the criticality of the products have an impact on the stringency of the processes used to plan/develop them, and as a consequence, these product/process changes have an impact on the corresponding subarguments of the assurance cases used to argue about process compliance and product's dependability. Within the AMASS project, BVR Tool has been

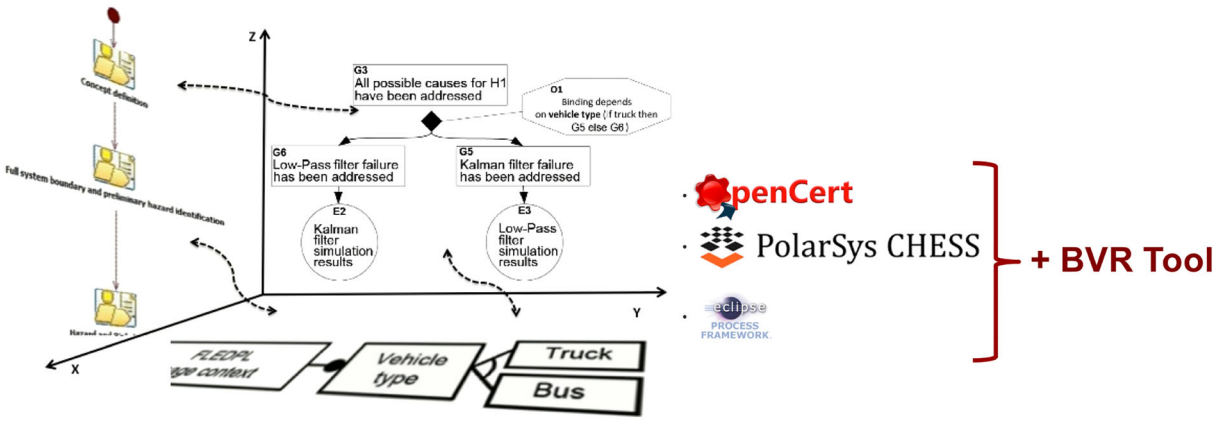

FIGURE 6 Three-dimension–oriented variability management, adapted from Gallina[35]

integrated not only with EPF Composer for managing the variability at process level but also with CHESS Toolset[‡] to manage the variability at product level and with OpenCert[§] to manage the variability at argumentation level. Thus, the seminal ideas, presented by Gallina,[35] were made concrete within the AMASS platform. Figure 6 depicts on the left the three-dimension–oriented conceptual variability management and on the right the tooling infrastructure available within the AMASS platform to support it. As a consequence, a VSpec model containing the variability for the three different dimensions (product/process/assurance case) can be represented. For an exemplification and details, the interested reader may refer to the AMASS deliverable D6.3.[36] Given such a VSpec model, the size of commonality and the product reusability could be considered for the three dimensions separately and see how changes in product's criticality affect the metrics. The calculated metrics could then be traced back to not only a goal focused on process-related time, costs, and risks reduction (see G1) but also goals on product and on assurance case-related time, costs, and risks reduction. Thus, achieving a more global view on quantitative evaluation of tailoring in the context of critical product line engineering. This, however, would not call for an additional extension of our proposed measurement model but, instead, would call for its parametrisation towards turning it into a pattern for quantitative evaluating the tailoring of families and their relationships.

7 | RELATED WORK

In the literature, as already discussed in our previous work,[2] other works have tackled the necessity of measuring reuse. However, none has conceived a methodological framework for measurement. The novelty of this paper, with respect to our previous one, consists in its orientation to the quantitative evaluation of multiple drivers and dimensions.

8 | CONCLUSION AND FUTURE WORK

In the context of security-informed safety life cycles mandated by standards and in the context of software process improvement-related standards and handbooks, process improvement via SPICE-compliant SiSoPLE seems to be feasible. Implementation of SPICE-compliant SiSoPLs in an organisation, however, needs to be planned and justified. This requires the ability to measure the effectiveness of the implementation at all stages of the development effort. In this paper, we have proposed a SPICE-compliant SiSoPLE-targeted GQMPS model to enable the measurement of SPICE-compliant SiSoPLE effectiveness. Then, we have applied it and demonstrated the effectiveness of S4S-compliant SiSoPLE in the context of ECSS standards. We may logically infer that because we measured and have the evidence for appropriateness of S4S-compliant SiSoPLE, implementation should enable process improvement. Our S4S-compliant SiSoPL is built on a very small part of the ECSS specifications and adopts an organisation goal/strategy linked measurement model. In addition, our study has been performed in relation to an organisation's specific processes and the associated standards and we examined only pre-S4S-compliant SiSoPL implementation metrics. Despite these limitations, as discussed, given the evident overlapping amount of process requirements, the efficient tailoring via SPICE-compliant SiSoPLE seems sound. We have also discussed the validity's threats and elaborated some perspectives towards achieving a more global view on quantitative evaluation of tailoring in the context of critical product line engineering.

In the near term future, the proposed SPICE-compliant SiSoPLE-targeted GQMPS model requires to be validated by assessing other SPICE-compliant SiSoPLEs. To that purpose, we plan to build on top of our previous work conducted in the automotive domain and apply SPICE-compliant SiSoPLE-targeted GQMPS to the Automotive SPICE-compliant SiSoPL at software design level. We also intend to consider its potential parametrisation towards turning it into a pattern for the quantitative evaluation of tailoring in all the dimensions of interest in the context of critical-product line engineering. Finally, in a long-term future, tool support will be developed.

ACKNOWLEDGEMENTS

This work is supported by EU and VINNOVA via the ECSEL JU under grant agreement no. 692474, AMASS project.[37]

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

FINANCIAL DISCLOSURE

None reported.

ORCID

Barbara Gallina https://orcid.org/0000-0002-6952-1053

[‡]https://www.polarsys.org/chess/
[§]https://www.polarsys.org/proposals/opencert

REFERENCES

1. Pries-Heje J, Johansen J, eds. Alcala, Spain: MANIFESTO Software Process Improvement eurospi.net; 2010. http://www.iscn.com/Images/SPI_Manifesto_A.1.2.2010.pdf. (Last accessed:June 19, 2019).
2. Gallina B, Iyer S. Towards quantitative evaluation of reuse within safety-oriented process lines. In: CCIS 896 of Proceedings of the 25th European Conference Systems, Software and Services Process Improvement - EuroSPI Bilbao, Spain September 5-7, 2018, Communications in Computer and Information Science. Bilbao, Spain: Springer International Publishing; 2018:469-479.
3. Jones M, Gomez E, Mantineo A. Mortensen UK. Introducing ECSS Software-Engineering Standards within ESA. http://www.esa.int/esapub/bulletin/bullet111/chapter21_bul111.pdf; August 2002.
4. European Cooperation for Space Standardization (ECSS). ECSS-S-ST-00C - ECSS system - Description, implementation and general requirements ESA Requirements and Standards Division. ESA-ESTEC; 2008.
5. European Cooperation for Space Standardization (ECSS). ECSS-M-ST-10C Rev. 1, Space project management - Project planning and implementation. ESA Requirements and Standards Division, ESA-ESTEC; 2009.
6. European Cooperation for Space Standardization (ECSS). ECSS-E-ST-40C, Space Engineering-Software. ESA Requirements and Standards Division, ESA-ESTEC; 2009.
7. ESA Standardization Steering Board. ESSB-ST-E-008 Secure Software Engineering Standard; 2016.
8. Fischer D, Spada M. Ready for secure software: secure software engineering for space missions, space operations: innovations, inventions, and discoveries. In: Proceedings of the 13th SpaceOps Conference, Pasadena, California, Progress in Astronautics and Aeronautics. Pasadena, California; 2014:333-351.
9. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). ISO/IEC 15408-1 - Information technology - Security techniques - Evaluation criteria for IT security - Part 1: Introduction and general model; 2014.
10. European Cooperation for Space Standardization (ECSS). ECSS-Q-ST-40C Rev.1, Space product assurance-Safety. ESA Requirements and Standards Division, ESA-ESTEC; 2017.
11. European Cooperation for Space Standardization (ECSS). ECSS-Q-ST-80C Rev.1, Space product assurance-Software product assurance. ESA Requirements and Standards Division, ESA-ESTEC; 2017.
12. European Cooperation for Space Standardization (ECSS). ECSS-Q-HB-80-02-Part1, Space product assurance - Software process assessment and improvement- Part 1: Framework. ESA Requirements and Standards Division, ESA-ESTEC; 2010.
13. Gallina B, Kashiyarandi S, Martin H, Bramberger R. Modeling a safety-and automotive-oriented process line to enable reuse and flexible process derivation. In: Proceedings of the 38th IEEE International Computer Software and Applications Conference Workshops (COMPSACW). Västerås, Sweden; 2014:504-509.
14. Gallina B, Sljivo I, Jaradat O. Towards a safety-oriented process line for enabling reuse in safety critical systems development and certification. In: Proceedings of the 35th Annual IEEE Software Engineering Workshop (SEW). Heraclion, Crete, Greece; 2012:148-157.
15. Gallina B, Kashiyarandi S, Zugsbrati K, Geven A. Enabling cross-domain reuse of tool qualification certification artefacts. In: LNCS 8696 of Proceedings of the1st InternationalWorkshop on DEvelopment, Verification and VAlidation of cRiTical Systems. Florence, Italy: Springer International Publishing; 2014:255-266.
16. Varkoi T, Mäkinen T, Galllina B, Cameron F, Nevalainen R. Towards systematic compliance evaluation using safety-oriented process lines and evidence mapping. In: Communications in Computer and Information Science, vol 748. Ostrava, Czech Republic: Springer, Cham; 2017:83-95.
17. Gallina B, Fabre L. Benefits of security-informed safety-oriented process line engineering. In: Proceedings of the 34th IEEE/AIAA Digital Avionics Systems Conference(DASC); 2015; Prague:8C11-8C19. Czech Republic, 13-17 September.
18. Castellanos Ardila JP, Gallina B. Towards efficiently checking compliance against automotive security and safety standards. In: Proceedings of the 28th IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). Toulouse, France; 2017:317-324.
19. Javed MA, Gallina B. Get EPF Composer back to the future: A trip from Galileo to Photon after 11 years. In: EclipseCon. Toulouse, France; 2018. June 13-14.
20. Javed MA, Gallina B. Safety-oriented Process Line Engineering via Seamless Integration between EPF Composer and BVR Tool. In: Proceedings of the 22Nd International Systems and Software Product Line Conference (SPLC); 2018; New York, NY, USA:23-28.
21. Cameron BG, Crawley EF. Crafting platform strategy based on anticipated benefits and costs. In: Advances in Product Family and Product Platform 49 Design: Methods & Applications. New York, NY: Springer; 2014.
22. Berger C, Rendel H, Rumpe B. Measuring the Ability to Form a Product Line from Existing Products. arXiv preprint arXiv:1409.6583; 2014.
23. Basili V, Trendowicz A, Kowalczyk M, et al. *GQM+ strategies in a Nutshell*. In: Aligning Organizations Through Measurement. Springer International Publishing; 2014:9-17.
24. Basili VR, Caldiera G, Rombach HD. The Goal Question Metric Approach. *Enc Softw Eng*. 1994;I:528-553.
25. AMASS Consortium. AMASS Objectives. http://www.amass-ecsel.eu/content/objectives. (Last accessed: June 19, 2019).
26. AMASS. Deliverable D1.1. Case Studies Description and Business Impact. https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/documents/D1.1_Case-studies-description-and-business-impact_AMASS_Final.pdf; 2017. (Last accessed: June 19, 2019).
27. Runeson P, Höst M. Guidelines for conducting and reporting case study research in software engineering. *Empir Softw Eng*. 2009;14(2):131-164.
28. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). ISO/IEC 15504 - Information technology - Process assessment; 2017.
29. Baufreton P, Blanquart JP, Boulanger JL, et al. Multi-domain comparison of safety standards. In: Proceedings of the 5th European Congress on Embedded Real Time Software and Systems(ERTS). Toulouse,France; 2010.
30. Blanquart JP, Ledinot E, Gassino J, et al. Software safety - a journey across domains and safety standards. In: Proceedings of the 9th European Congress on Embedded Real Time Software and Systems (ERTS), hal-01734621, HAL CCSD; Toulouse, France: 2018. January.
31. Korsaa M, Johansen J, Schweigert T, et al. The people aspects in modern process improvement management approaches. *J Soft Evol Process*. 2013;25(4):381-391.

32. Korsaa M, Biro M, Messnarz R, et al. The SPI Manifesto and the ECQA SPI Manager Certification Scheme. *J Soft Evol Process*. 2012;25(5):525-540.
33. Messnarz R, Sicilia MA, Biro M, et al. Social Responsibility Aspects Supporting the Success of SPI. *J Sof Evol Process*. 2014;26(3):284-294.
34. Gallina B, Nyberg M. Reconciling the ISO 26262-compliant and the Agile Documentation Management in the Swedish context. In: Proceedings of the 3rd International Workshop on Critical Automotive applications: Robustness & Safety (CARS), Matthieu Roy, Paris, France, HAL; 2015.
35. Gallina B. Towards enabling reuse in the context of safety-critical product lines. In: Proceedings of the 5th IEEE/ACM International Workshop on Product Line Approaches in Software Engineering (PLEASE), Florence, Italy, May 19; 2015.
36. Consortium A. Design of the AMASS tools and methods for cross/intra-domain reuse (b). https://www.amass-ecsel.eu/content/deliverables;2018. (Last accessed: June 19, 2019).
37. AMASS Consortium. AMASS (Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems). http://www.amass-ecsel.eu; (Last accessed: June 19, 2019).

How to cite this article: Gallina B. Quantitative evaluation of tailoring within SPICE-compliant security-informed safety-oriented process lines. *J Softw Evol Proc*. 2019;e2212. https://doi.org/10.1002/smr.2212