This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

Safety Practices in Requirements Engineering: The Uni-REPM Safety Module

Jéssyka Vilela, Jaelson Castro, Luiz Eduardo G. Martins, and Tony Gorschek

***Abstract*—Context: Software is an important part in safety-critical system (SCS) development since it is becoming a major source of hazards. Requirements-related hazards have been associated with many accidents and safety incidents. Requirements issues tend to be mitigated in companies with high processes maturity levels since they do their business in a systematic, consistent and proactive approach. However, requirements engineers need systematic guidance to consider safety concerns early in the development process. Goal: the paper investigates which safety practices are suitable to be used in the Requirements Engineering (RE) process for SCS and how to design a safety maturity model for this area. Method: we followed the design science methodology to propose Uni-REPM SCS, a safety module for Unified Requirements Engineering Process Maturity Model (Uni-REPM). We also conducted a static validation with two practitioners and nine academic experts to evaluate its coverage, correctness, usefulness and applicability. Results: The module has seven main processes, fourteen sub-processes and 148 practices that form the basis of safety processes maturity. Moreover, we describe its usage through a tool. Conclusions: The validation indicates a good coverage of practices and well receptivity by the experts. Finally, the module can help companies in evaluating their current practices.**

***Index Terms*—Safety-critical systems, Requirements Engineering, Maturity Models, Uni-REPM, Safety Engineering.**

I. Introduction

Safety-critical systems consist of a set of hardware, software, process, data and people whose failure could result in accidents that cause damage to the environment, financial losses, injury to people and loss of lives [1], [2].

In this context, the literature reports that software has contributed to deaths and injuries in many safety incidents and safety-related catastrophes [1], [3]–[7] and several studies have identified problems with the RE process for SCS [8]–[11]. Currently, software is being used to implement and/or control an increasing number of traditional as well as innovative functions [12]. Furthermore, software also handles functions that were controlled by humans [12].

Therefore, software is becoming a major source of risks and hazards since it can give wrong instructions to system hardware, through actuators, that can lead to accidents and hurt people [12]. Hence, considering the relevance of maintaining high confidence in safety-critical software [13], a consensus in academia and industry is being established that safety concerns should be addressed early in the system lifecycle [1], [2], [12], [14].

Companies that develop SCS face some issues during system development such as (i) absence of systematization of available safety actions/practices [15], [16]; (ii) need of integration among safety, RE and the broad context of product development, management and corporate strategy [1], [2], [14]; (iii) lack of a model to guide them on how to apply their efforts systematically to achieve safety goals and to maintain continuous improvement in safety implementation [15], [16], which can continually drive actions toward higher safety levels; (iv) difficulties in determining and assigning priorities to safety actions/practices to be adopted [15].

In order to ensure a systematic safety processes evolution, various aspects (e.g. organizational, technical, strategic) have to be addressed [10]. Aiming to unify the development process and give guidance to companies, some safety standards are available. Nevertheless, standards do not have the notion of process capability or maturity [17]. Moreover, they have contributed to the development of systems historically depicted as mature or highly-evolved. This makes their implementation challenging for those companies starting to follow standards aiming to increase their systems safety [18].

In this context, determining the capability of safety processes [16] has been identified as necessary to have more technical results that can be used in a continuous process improvement [16], [17].

Recent literature reports an increasing interest in maturity models [19], [20] to fill the gap of safety standards [12]. Some safety maturity models have been proposed such as +SAFE-CMMI-DEV [21] and ISO 15504-10 [22]. Though these models cover the whole software development process, they are not complete and detailed enough to guide SCS companies during the RE phase.

Addressing safety concerns early in software development contributes to ensuring that safety problems do not propagate through subsequent phases [23]–[25]. Furthermore, if changes in the system are stopped early, it is more likely to have the opportunity to obtain a stable and error-free software version [1], [25], [26].

Therefore, the early consideration of safety concerns in RE should be a top priority in the development of SCS since RE is essential for software quality [27], and effectiveness of software development process [10]. Moreover, high safety levels are typically best achieved by addressing safety from the beginning; not by trying to add protection components

J. Vilela is with Universidade Federal do Ceará (UFC) and Universidade Federal de Pernambuco (UFPE), Brazil, e-mail: jessykavilela@ufc.br.

J. Castro is with Centro de Informática of Universidade Federal de Pernambuco (UFPE), Recife-PE, Brazil, e-mail: jbc@cin.ufpe.br.

L. E. G. Martins is with Departamento de Ciência e Tecnologia of Universidade Federal de São Paulo (UNIFESP), São José dos Campos, Brazil, email: legmartins@unifesp.br.

T. Gorschek is with Blekinge Institute of Technology (BTH), Sweden, email: tony.gorschek@bth.se.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

and additional complexities after system has been developed [1], [28].

Mature companies do their business in a systematic and proactive approach [20], [29]. On the other hand, immature companies can and do produce good quality requirements documents, but they may not be able to do so consistently or when working with tight deadlines. Such companies lay emphasis on fixing problems right away and only obtain their results through the efforts of determined subjects, while deadlines and budgets are often exceeded [20].

Hence, a mature and practicable process contributes to eliminate errors [25] and requirements problems [30] from the beginning. Furthermore, process improvement and process assessment frameworks allow to transfer research results into practice [31].

There are some RE assessment frameworks [32]–[34] that allow companies to evaluate the strengths and weaknesses [20] regarding the RE process. However, these maturity models do not cover both market-driven and bespoke RE at the same time [35]. To fill this gap, Uni-REPM was proposed, but it did not originally consider the safety practices required for the development of a safety-critical system.

This paper proposes a safety maturity module for Uni-REPM, called Uni-REPM SCS, that companies can use it as a guide to assess and improve their current safety practices and processes. It is relevant to note that as the original Uni-REPM, the safety module is not purely prescriptive, but rather both the evaluation aspect and the improvement part of the model are context aware, i.e. companies can define based on the project context what is relevant for them to use, and what is not.

The goal of the safety module is not only to offer to companies that develop SCS a way to evaluate and improve their their RE processes, but also to enable previously non-SCS development companies to incorporate SCS aspects into their RE processes.

We used multiple information sources to collect data and to define the practices to be included in a RE module for SCS, including two Systematic Literature Reviews (SLR), one large interview study with companies, and an inventory and extraction from several safety standards.

We evaluated the safety module in terms of coverage, correctness, usefulness, and applicability. This evaluation relied on the feedback of two practitioners and nine academic experts regarding the structure and safety practices presented in the module.

This paper is organized as follows. Section II introduces background and related work. The research methodology followed to develop the Uni-REPM Safety module is presented in Section III. The module structure and contents are described in Section IV. The module evaluation is discussed in Section V. Discussions about the module are exposed in Section VI. Our conclusions are presented in Section VII.

II. Background and Related Work

In this section, we define some concepts used in this work to ensure consistency throughout this paper and discuss related works.

A. Background

1) Maturity Models: Maturity can be classified as the state of being complete, perfect or ready [20], [36]. Accordingly, this concept suggests an evolutionary progress from an initial to a desired stage [17], [20], [36]–[39].

In software engineering, the notion of maturity is used by maturity models (MM) to assess the capabilities of a company [36], [40], [41].

According to Wendler [39], a maturity model is *"a structured collection of elements that describe the characteristics of effective processes at different stages of development. It also suggests points of demarcation between stages and methods of transitioning from one stage to another"*.

These models make easy the assessment of companies by outlining anticipated, typical, logical, and desired evolution paths [36]. MM define best actions/practices for software life-cycle processes, based on good engineering and process-management principles, and process-attribute sets for capability/maturity design aspects [42]. Therefore, the objective in using MM is to detect and delete bad capabilities [19], but also detect good capabilities are missing and must be added. Thus, the application of this concept is not limited to any particular domain [39].

Typically, maturity models:

- define capability areas, process areas or design objects [17], [36]–[39], [41];
- consist of sequential maturity stages [17], [20], [36]–[39];
- have a hierarchical progression from an initial to a desired stage [17], [20], [36], [37], [39];
- involve a wide range of organizational activities and actions/practices [36], [38], [39], [41], [42];
- have an assessment instrument that can either be qualitative or quantitative [36], [37].

2) Uni-REPM: Uni-REPM is an universal lightweight model to evaluate the maturity of a RE process structured in two views: Process Area and Maturity Level [35].

The model hierarchy has three levels, namely Main Process Area (MPA), Sub-Process Area (SPA) and Action. On the top level of the model, there are seven Main Process Areas (Organizational Support, Requirements Management Process, Elicitation, Requirements Analysis, Release Planning, Documentation and Requirements Specification, and Requirements Validation) corresponding to RE main activities.

Each MPA is further broken down into several SPAs and, on the bottom level, an Action denotes a certain activity that should be done or a certain item that should be present. Each Action has a maturity level (1, 2 or 3) that corresponds to "Basic", "Intermediate", or "Advanced" level.

The Uni-REPM has an assessment instrument in which the appraiser can mark one of three options: "Incomplete" (vital action performed partially or not at all in the RE process), "Complete" (action performed completely), and "Inapplicable" (action was not necessary).

The model was designed with the ideas of being lightweight, and also a self-assessment and improvement tool. It can be used by professionals themselves acting as evaluators - making improvements based on recent lessons learned.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

B. Related Work

Many software process capability/maturity models have been elaborated, expanded and modified over the past years. Accordingly, some SLRs about maturity models have been conducted [20], [39], [43].

The SLRs show that there is a clear trend to propose maturity models customized to specific domains, small and medium enterprises, testing and quality assurance, security engineering, extreme programming, e-government, medical systems, space, telecommunications, software development among other domains [42].

1) Software maturity models: Generic software process improvement frameworks such as Capability Maturity Model Integration (CMMI) [44], SPICE [45], ISO9000 [46] have been proposed and adopted by companies. Although they address RE in some extent, they do it shallowly since their scope is to cover all phases of development process having a much bigger scope than just RE [35].

The above maturity models emphasize bespoke RE which is related to the development of a customized software system for a specific customer [34]. Nevertheless, they have not been updated with RE actions/practices in industry [35]. According to Svahnberg et al. [35], there are practices not handled at all by these models, and other actions are classified as being very advanced whereas in current state of practice they are the common procedure.

2) RE maturity models: There are some RE assessment frameworks, for example, the Requirements Engineering Good Practice Guide (REGPG) [32], Requirement Engineering Process Maturity Model (REPM) [33], Market-Driven Requirements Engineering Process Maturity Model (MDREPM) [34], and others that allow companies to evaluate the strengths and weaknesses [20] regarding the RE process.

The REGPG guide [32] suggests a model based on 66 good requirements practices, where 36 are classified as basic, 21 as intermediate and 9 as advanced. The advanced practices are concerned with formal specification which is recommended for critical systems development. However, the model proposes only 9 practices for SCS being very succinct. Moreover, in our opinion, its implementation is very challenging for small/immature companies aiming to increase the safety of their systems.

The MDREPM model [34] is an evolution of REPM [33] to consider market-driven practices. Market-driven RE is applicable to software companies that develop software to a determined market, which can be a combination of a number of known customers or, on another extreme, a mass market where customers cannot be clearly indicated [34].

However, REGPG, REPM, and MDREPM do not cover both market-driven and bespoke RE as required by industry [35]. To fill this gap, the Unified Requirements Engineering Process Maturity model (Uni-REPM) [35] was proposed but it does not consider safety actions required for the development of a safety-critical system. Therefore, in this work, we propose a safety module for Uni-REPM.

3) Safety maturity models: Safety culture maturity models are available in literature [47], [48]. However, safety culture is a characteristic of groups and companies that handle organizational collective practices to avoid accidents during the work in factories [48] and not about developing SCS.

Some safety maturity models have been developed, for example, +SAFE-CMMI-DEV [21] and ISO 15504-10 [22]. However, these models are too general, usually adopted by safety engineers, and do not provided detailed specific safety practices for RE as well as the particularities of these two areas as in our work.

III. Research Methodology and Module Creation

We followed the design science methodology [49] as a research method to develop the safety module. This methodology requires that solutions must be iteratively proposed, refined, evaluated, and, if necessary, enhanced [40], [49].

We have done several works on safety-critical systems and with companies involved in SCS development to define the sources of information for this module:

1) performed SLRs (SLR1, SLR2) [2], [14], [50];
2) conducted empirical studies with 11 companies (INTERVIEW-STUDY) [51], [52] and elaborated technical reports (TECH-REPORT) [53];
3) examined safety standards (SAFETY-STD) [18], [54]–[68];
4) analyzed existing maturity models (EXISTING-MATURITY-MODS) [21], [22], [69], [70];
5) conducted comprehensive analyis of important authors in the field (STATE-OF-THE-ART) [1], [3], [7], [32], [71]–[82].

A. Module construction process

The steps of the methodology adopted to construct the Uni-REPM SCS are presented in Figure 1. The module construction process involved nine steps, each building on the ones before it. Although methodologies for creating maturity models are available in literature [19], [37], [40]–[42], and inspired our model, we added new steps, as needed based on based on the our experience/opinion, along the way.

1) Knowledge Acquisition: Our first step was to investigate the literature available about RE in safety-critical systems to become familiar with the domain. In this step, we studied the problem space, comprehend the concepts involved in the safety domain as well as to investigate the problems in the integration of these two areas.

We also analyzed the ISO/IEC 25010 [56] international quality standard that is part of Systems and software Quality Requirements and Evaluation (SQuaRE) System and software quality models. ISO/IEC 25010 cancels and replaces ISO/IEC 9126-1 [57] and extends quality models to include computer systems, and quality in use from a system perspective.

2) Problem definition: After the comprehensive investigation of the domain (STATE-OF-THE-ART) and based on the identified existing safety maturity models (EXISTING-MATURITY-MODS) [21], [22], [69], [70], we investigated the problem relevance, i.e. the actual demand for the maturity model [40]. Based on this, we concluded that there was a need for proposing the Uni-REPM SCS according to the objectives and scope of our work.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

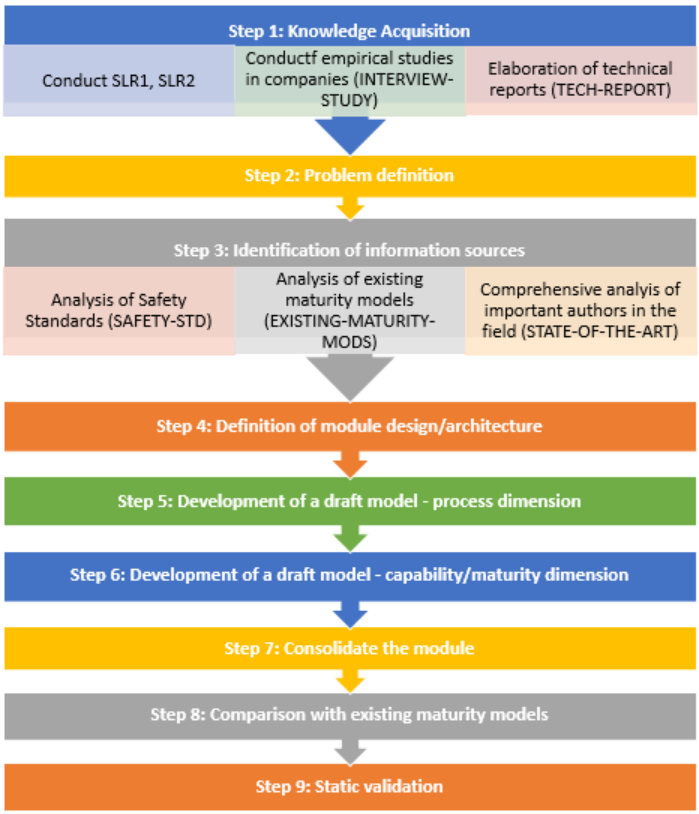


Fig. 1. Methodology for creating Uni-REPM SCS.

3) Identification of information sources: We performed a comprehensive literature review (STATE-OF-THE-ART) to identify and select the information sources for the safety actions/practices.

We also reviewed safety standards (SAFETY-STD) [18], [54]–[68] as a way to knowledge acquisition since in the interviews conducted (INTERVIEW-STUDY), the practitioners highlighted the need and importance of following an adequate safety standard:

"*the certification process pushes us to define requirements with more details*" [51].

"*We have a set of international specific standards we have to follow to build the machines. We really need to identify what parts of the standards are applicable for the machinery we are designing (cutting, laminator, printing machine, and so on)*" [52].

"*the certification process improves the relationship with suppliers, because everyone must follow the same safety standards, which are used as artifacts of communicating between the company and its suppliers*" [51].

Finally, we also analyzed the works of important authors in the field (STATE-OF-THE-ART) [1], [3], [7], [32], [71]–[82]. The list of information sources is presented in Section IV-A.

4) Definition of module design/architecture: After the analysis of the information sources and data extraction, we established the design/architecture of the module [40]. The comparison of existing maturity models with the problem definition suggests the enhancement of an existing model as a design strategy.

Accordingly, we proposed a module following the structure of Uni-REPM. We opted to follow its structure since we are designing a safety module for an universal lightweight maturity model, capable of evaluating the maturity of a RE process, that has been used and well accepted in companies. Moreover, the SPAs already present in the Uni-REPM cover the main processes involved in the RE phase.

In the module definition process, existing safety maturity models (EXISTING-MATURITY-MODS) were used as a starting point for the design process because they already cover some RE aspects for the safety-critical domain. In this context, we identified the features that maturity models typically have.

The safety module follows the same hierarchical structure of multiple layers adopted by many maturity models [37], [41] and Uni-REPM [35] and features listed in Section II-A1.

5) Development of a draft model - process dimension: The next step consisted in the development of a draft module in the process dimension. Hence, we defined the 14 sub-processes areas of the module, their respective actions as well as how they would be connected to the Uni-REPM model.

6) Development of a draft model - maturity dimension: We updated the draft module considering the maturity dimension by assigning a fixed number of maturity levels for the actions already determined.

We opted to maintain the likert scale [83] with three levels of Uni-REPM (Basic, Intermediate, Advanced), as adopted by other MM [32], [84], considering the difficulties that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

users have in choosing among five options with very discrete differences as adopted in many maturity models.

Accordingly, we want users to be aware and clearly distinguish among the stages, reducing implications on module application and improving interpretation of stages. This reduced number of maturity levels makes easier for practitioners to understand what it means for their RE is assessed to be on a particular maturity level [35].

7) Consolidate the module: Then, we performed the consolidation of the module in a way understandable for the target group [19] by discussing it and refining the module several times.

8) Comparison with existing maturity models: We performed a comparison among existing maturity models in safety (EXISTING-MATURITY-MODS), whose results are described in Section VI-B. All maturity models analyzed cover the entire project lifecycle. Hence, they do not go into detail into any particular practice area, such as RE. Therefore, the maturity model we propose is more descriptive and detailed because it was designed specifically for safety in RE and contains a comprehensive assessment instrument.

9) Static validation: We adopted the technology transfer framework proposed by Gorschek et al. [85] to perform the safety module validation. This model was proposed through a partnership between academia and industry and it has been cited more than 180 times on Google scholar.

In this step, we collected feedback regarding the contents of the module and its coverage of safety practices from 9 subjects (7 academic experts and 2 practitioners). The results of this validation are presented and discussed in Section V.

In the next section, we describe the structure, contents and usage of the Uni-REPM SCS.

IV. The Uni-REPM Safety module

RE issues such as vague initial requirements, ambiguities in requirements specification, undefined requirements process, requirements growth, requirements traceability, and confusion between methods and tools [31], [86]–[88] have a huge impact on the quality of a SCS.

In this context, there is a consensus that the most cost-efficient place to correct many problems is in the RE phase [31], [89], [90]. Despite this, RE remains somehow a neglected area [31], [86], [87], [91], [92].

Requirements problems are less frequent in companies with high maturity levels [30]. Therefore, Uni-REPM SCS aims to reduce issues in RE during SCS development by addressing safety actions/practices that should be covered in the RE process to reduce the gap between these areas.

In the next sections, we describe the sources of actions, module structure, its contents and how to use it to evaluate the maturity level of an organization.

A. Sources of the actions

Uni-REPM SCS is based on several sources as described in Section III: Systematic Literature Reviews (SLR1, SLR2), empirical studies (INTERVIEW-STUDY), technical reports (TECH-REPORT), safety standards (SAFETY-STD), existing maturity models (EXISTING-MATURITY-MODS), and comprehensive literature review (STATE-OF-THE-ART).

The traceability information for the safety actions/practices is presented in Table I. This information is presented so that the reader may locate the sources of individual actions in the module.

B. Module overall structure

The Uni-REPM safety module follows the dual-view-approach of Uni-REPM: Process Area and Maturity Level.

The process area view allows to visualize the hierarchy of processes that consist the model and consult actions/practices of the same group. The maturity level view, on the other hand, classifies the practices by level where the actions in one level supports each other as well as the more advanced practices on the next level [35].

1) Process area view: The module follows the same hierarchy of Uni-REPM that defines three levels: Main process area (MPA), Sub-process area (SPA) and Action. Figure 2 presents the safety module and its relationship with Uni-REPM. The module extends Uni-REPM by adding new SPAs highlighted through dashed lines. Existing process outcomes were not altered and none were removed.

Since we want to integrate safety in the RE process, we maintained the seven MPAs of Uni-REPM that were defined considering well-adopted processes such as Kotonya and Sommerville [93]. The MPAs are described below:

1) *Organizational Support (OS)*: assesses the quantity of support provided to RE practices from the surrounding companies.
2) *Requirements Process Management (PM)*: contains activities to manage, control requirements change as well as to assure that the process is being followed.
3) *Requirements Elicitation (RE)*: handles actions for discovering and understanding the necessities and desires of costumers in order to communicate them to others stakeholders.
4) *Requirements Analysis (RA)*: contains activities to detect errors, create detailed view of requirements as well as to estimate information needed in later activities of RE process.
5) *Release Planning (RP)*: comprises important actions to define the optimal set of requirements for a certain release in order to accomplish defined/estimated time and cost goals.
6) *Documentation and Requirements Specification (DS)*: addresses how a company structures the requirements and other information collected during elicitation into consistent, accessible and reviewable documents.
7) *Requirements Validation (RV)*: includes checking the requirements against defined quality standards and the real needs of the several stakeholders. Its aim is to assure that the documented requirements are complete, correct, consistent, and unambiguous.

Sub-process area (SPA) contains closely related actions, which help to achieve a bigger goal. The unique identifier assigned to each SPA is composed of the MPA identifier

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

TABLE I
SAFETY UNI-REPM TRACEABILITY INFORMATION.

Type	Author	Reference	Actions	Total Number of actions
STATE-OF-THE-ART	Leveson	[3]	13,14,16,19,20,23,25,41,43,44,52,53,54,55,56,57,62, 65,69,71,75,76,77,80,81,82,86,87,91,94,95,108,133,145	32
	Sommerville and Sawyer	[32]	3,13,16,18,22,27,28,31,32,34,37,42,43,58,65,75,114,118, 132,137,138,139,140,142	24
	Kontogiannis et al.	[80]	7,8,19,24,51,90,95,98,107,108,109,111,114,133	14
	Nancy Leveson (2011)	[1]	7,8,17,30,37,38,58,68,85,87,138,144	12
	Lami et al.	[82]	5,66,147,148	4
	Jim Whitehead	[76]	134,135,137	3
	Firesmith	[79]	35,36,37	3
	Schedl and Winkelbauer	[71]	25,77,94	3
	Kazaras and Kirytopoulos	[73]	37,38	2
	Kim, Nazir and overgard	[72]	37,38	2
	Leveson et al. (2002)	[7]	10,16	2
	Thomas Grill and Margit Blauhut	[78]	9	1
	Ekberg et al.	[75]	106	1
	Pernstal et al.	[81]	5	1
	Jon G. Hall, Andr Silva	[77]	8	1
SAFETY-STD	ISO 61508	[54]	18,21,24,28,37,39,46,59,61,62,63,64,67,72,78,90,92,93, 95,102,104,105,110,112,118,119,120,122,123,124,125,126,127,143	34
	ECSS-E-HB-40A	[61]	4,11,12,45,49,60,66,70,73,100,101,103,104,128,129,140,146,147	18
	MIL-STD-882C	[65]	19,32,45,57,58,62,77,79,84,85,87,90,95,96,98,131	17
	ISO/TS 15998-2	[59]	31,32,39,47,48,89	6
	MIL-STD-882E	[67]	29,32,33,70,90	5
	ISO 13849-1	[63]	4,11,15,39,141	5
	ECSS-E-ST-40C	[62]	74,146,147,149	4
	ISO 14639-1	[18]	107,109,111	3
	MIL-STD-882D	[66]	69,84	2
	ISO 13849-2	[64]	50,62	2
	ISO 26262-6	[55]	67	1
	ISO 15998	[58]	88	1
	ISO/TR 14639-2	[68]	109	1
INTERVIEW-STUDY	Martins and Gorschek	[51]	2,6,9,25,26,31,32,34,42,67,70,84,86,105,106,107,113,114,115, 124,130	21
EXISTING-MATURITY-MODS	+Safe-CMMI-DEV	[21]	1,25,28,31,32,33,39,56,58,77,83,85,94,59,95,120,124,133,135	19
	SW-CMMI	[69]	1,2,3,5,61,62,66,74,76,85,93,96,98,99,118,121,127	17
	ISO 20474-1	[60]	7,8,9,10,11,12	6
	ISO/IEC TS 15504-10	[22]	93,123	2
	SE-CMM	[70]	76,99	2

to which the SPA attaches and its abbreviation. For example, "OS.SKM" represents a sub-process area called "Safety Knowledge Management (SKM)" which resides under MPA "Organizational Support".

We propose fourteen SPAs to be connected to the seven MPAs described above.

- **Safety Knowledge Management (SKM):** provides transparency in the development process by making sure that projects and the company have the required knowledge and skills to accomplish project and organizational objectives.
- **Safety Tool support (STO):** is responsible for facilitating the appropriate execution of the corresponding tasks and manage all safety-related information that should be created, recorded and properly visualized.
- **General Safety Management (GSM):** covers project safety management activities related to planning, monitoring, and controlling the project.
- **Safety Planning (SP):** provides the safety practices and establishes a safety culture in the company.
- **Safety Configuration Management (SCM):** addresses the control of content, versions, changes, distribution of safety data, proper management of system artifacts and information important to the organization at several levels of granularity.
- **Safety Communication (SCO):** aims to improve the safety communication sub-process by establishing actions related to many safety terms, methods, process to support the safety analysis and assurance processes.
- **Safety Traceability (ST):** handles the traceability among artifacts helping to determine that the requirements affected by the changes have been completely addressed.
- **Supplier Management (SM):** is responsible to manage the acquisition of products and services from suppliers external to the project for which shall exist a formal agreement.
- **Preliminary Safety Analysis (PSA):** addresses the realization of a preliminary safety analysis to avoid wasting effort in next phases of system development.
- **Failure Handling (FH):** handles failures in system components that can lead to hazardous situations, addition of redundancy as well as protection mechanisms.
- **Safety Certification (SC):** has actions related to system certification.
- **Human Factors (HF):** handles issues regarding system's users and operators that can lead to hazards and shall be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

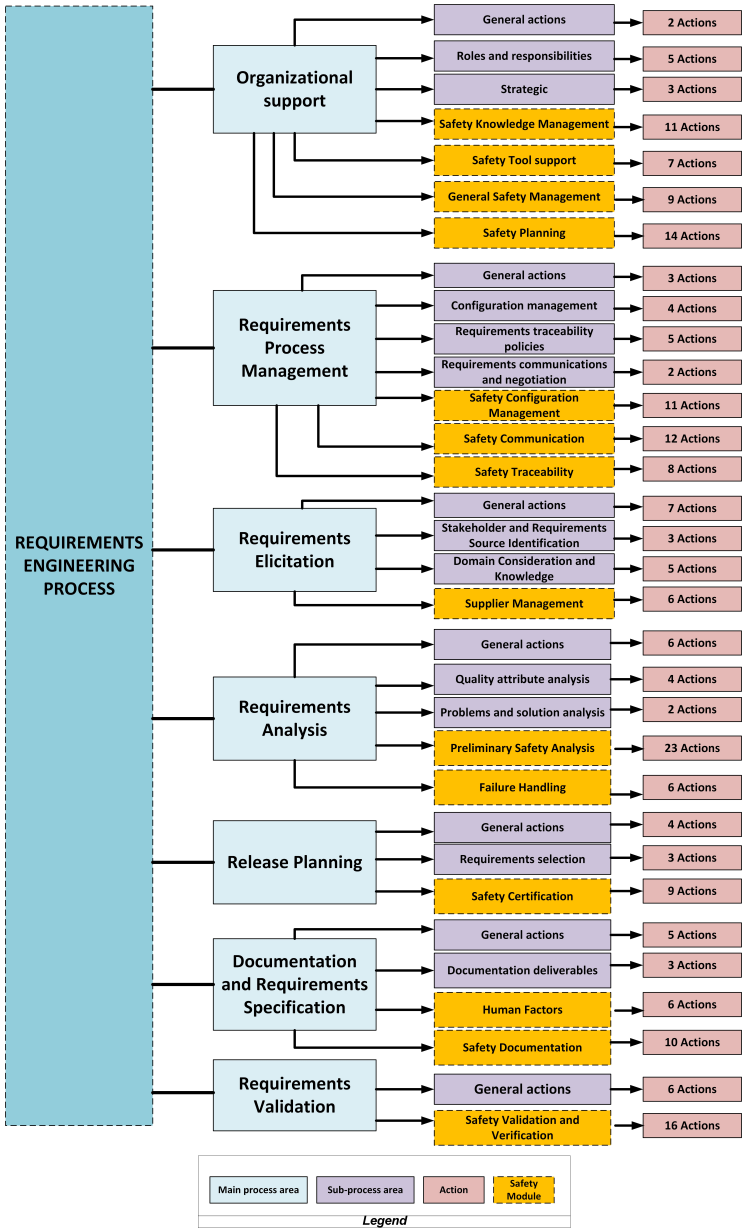


Fig. 2. Safety module and its relationship with Uni-REPM.

considered during the RE stage of safety-critical system development.

- **Safety Documentation (SDO):** has practices to record all information related to system's safety produced in RE phase.
- **Safety Validation and Verification (SVV):** contains actions for requirements validation and the definition of strategies for the verification of requirements aiming to obtain requirements clearly understood and agreed by the relevant stakeholders.

At the low-level of module structure, we have "actions" that represent a specific good practice. By performing the action, the organization can improve their process and gain certain benefits [35].

For example, an action "*Develop a safety information system to share knowledge in the organization*" once implemented in the organization will enable practitioners to share knowledge in the organization improving the communication among them.

In the safety module, actions also follow the same format assigned to sub-processes to define their unique identifiers and also used to define safety extensions [16]. Actions are identified by the MPA/SPA under which they reside, followed by an "a" which stands for "action" and their position in the group. For example, "*OS.SKM.a1*" means the first action under MPA "*Organizational Support*" and SPA "*Safety Knowledge Management*".

Besides the description of each action, there can be recommendations and supporting actions [35]. Recommendations provide suggestions on well-adopted techniques or supporting tools to practitioners implementing an action. On the other

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

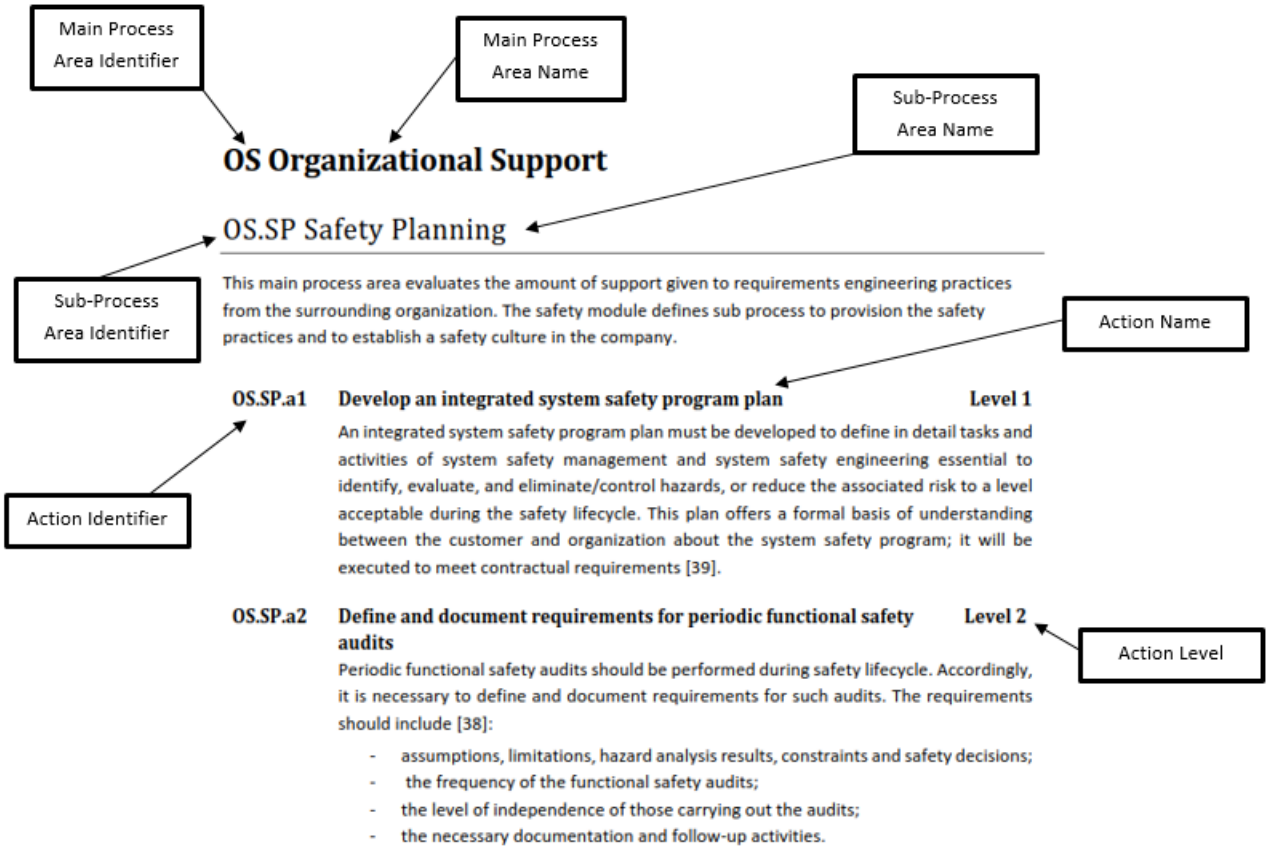


Fig. 3. Example of an Uni-REPM SCS action.

hand, supporting actions define links to other actions which will benefit the practitioners if they are implemented together.

The complete description of the module as well as an online software tool developed to conduct RE/Safety evaluations using Uni-REPM and the safety module can be found on the project homepage: www.unirepm.com.

2) Maturity Level view: It is developed by assigning a level to each action in a likert scale from 1 to 3, corresponding to "Basic", "Intermediate", and "Advanced" level where 3 represents the highest level of maturity. The maturity levels are:

1) 1 (Basic): The aim of this level is to achieve a rudimentary repeatable safety requirements engineering process. The process in this level is defined and followed. Basic usability and interface aspects are considered, basic safety-related information is incorporated into system artifacts, responsibility, accountability and authority are identified, a lifecycle for projects artifacts is defined, and it also contains initial practices to establish a safety culture in the company. Moreover, in this level, preliminary safety and hazard reports, a maintenance plan are constructed, the system behavioral model and restart-up procedures are specified, and a preliminary safety analysis is performed. Initial project monitoring and take corrective actions are implemented and a common nomenclature is established. This level corresponds to "present-state"; meaning that there is no activity performed to collect and analyze data/feedback for future improvement of the process.
2) 2 (Intermediate): in this level, the process is more rigorous because it involves various perspectives and is led by product strategies/goals. Operator task models are evaluated, ergonomic principles are considered. Life-cycle and modification activities, system development methodology, competence requirements, safety policy and safety goals are clearly defined and documented. Safety manual is elaborated, hazard and risk analysis results are maintained throughout the overall safety lifecycle, the hazards auditing and log file as well as working groups and structures are established, safety experience on similar systems are considered, and tools are used to support the processes. The preliminary level of safety achieved by the system and preliminary compliance with safety standards are demonstrated, safety audits are conducted and initial traceability mechanisms are considered in system artifacts. Also considered are requirements for the avoidance of systematic faults and fault-detection procedures. Moreover, external systems and safety-related software concerns as well as system integration procedures are handled, and communication among stakeholders is also considered.
3) 3 (Advanced): it denotes the most mature process. The improvements in the process are shown in the advanced way of documenting lessons learned, sharing knowledge

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

in the organization, specifying the general safety control structure, formal agreements among organization and suppliers are established and maintained. Moreover, formal communication channels among different organizational levels and common safety information system for system specification and safety analysis are used.

It is important to note that Uni-REPM SCS is not intended to be used as part of a product assessment. While safety standards require the definition of safety integrity level of the system under development to set requirements for the project and the system, the module provides a way to evaluate the capability of safety-related processes as well as a scheme for their improvement.

Therefore, the maturity level achieved is not related with the safety integrity level the project has to fulfill. Accordingly, an evaluation based on safety maturity models, such as Uni-REPM SCS, +SAFE or ISO 15504-10, is not analogous to a functional safety assessment. Hence, using a maturity model does not provide any guarantee of compliance with any safety standard.

Moreover, Uni-REPM SCS does not prescribe any specific technique, method or tool. Its goal is to consider the process (the "what") and it does not require the adoption of any specific technique or method (the "how").

These different levels of maturity were specified according to the difficulty to implement the action, how essential it is for the RE process, dependencies among actions, the frequency they appear in different information sources as well as the ability of optimizing the safety processes considering our experience and the results of literature reviews and safety standards.

This view shows the actions from all process areas which the organization should implement in order to achieve a specific maturity level. We list the practices by level in Tables II, III, and IV.

C. Examples of definition of actions

The definition of the actions considered the sources of information listed in Table I. The different sources and their terminologies were reconciled by reading the full description of action and we adopted the most common term. For example, the description of action *OS.SP. Safety Planning* is presented in Figure 3. This action was identified through both SAFETY-STD [65], EXISTING-MATURITY-MODS [21], and STATE-OF-THE-ART [3], [71].

In the SAFETY-STD source, the MIL-STD-882C [65] requires the development of a System Safety Program Plan (SSPP) that should specify in detail activities required to identify, evaluate, and eliminate/control hazards, or reduce the associated risk to an acceptable level.

+SAFE [21] of EXISTING-MATURITY-MODS category declares that elements of the plan for performing the safety engineering process are part of the safety plan that may take on various formats according to the requirements of regulatory agencies.

Studies in the STATE-OF-THE-ART also address the need of conducting a proper safety planning. Schedl and Winkelbauer [71] say that the process related to customer requirements have to be assessed and resources planning should be performed and detailed in a System Safety Plan. Leveson [3] also states that in the early stage of system development, the system safety program plan should be developed.

Another example of an action is *"Identify and document responsibility, accountability and authority"* that was inspired in SAFETY-STD, EXISTING-MATURITY-MODS, STATE-OF-THE-ART.

In the SAFETY-STD category, the IEC 61508 safety standard [54] requires that a company developing an electrical/electronic/programmable electronic safety-related system must designate one or more persons to assume responsibility for safety activities. In the same way, the MIL-STD-882C [65] defines that a safety plan should depict the responsibility and authority of safety team, other contractor organizational elements involved in the system safety effort, subcontractors, and system safety groups.

EXISTING-MATURITY-MODS also requires evident specification of responsibility. +SAFE [21] demands that who or what should be assigned responsibility and authority for performing the processes, developing the work products, and providing the services of the safety management process.

Responsibility and authority are also a concern reported in STATE-OF-THE-ART. Leveson [3] determines in her proposal of elements in a safeware program that a safety policy should include a well-defined assertion of responsibilities, authority, accountability, and scope of activities. Finally, Kontogiannis et al. [29] states that risk management should be comprehensive and clear about accountability.

In the next sections, we explain some examples of how we grouped the actions in sub-process areas of Uni-REPM SCS.

D. Examples of definition of SPAs

1) SPA: Safety Knowledge Management (SKM): The SKM sub-process provides transparency in the development process by making sure that projects and the company have the required knowledge and skills to accomplish project and organizational objectives. The definition of this SPA was inspired in the presence of safety practices related with knowledge sharing in an organization as described in the works of the SAFETY-STD, INTERVIEW-STUDY, STATE-OF-THE-ART categories.

In the SAFETY-STD, for example, we noticed the need of an infrastructure to share knowledge (OS.SKM.a1) in [18], control access mechanisms to the safety information system (OS.SKM.a3) in [18], [68], and maintain employees' competence information (OS.SKM.a4) [54].

In the INTERVIEW-STUDY [51], the practitioners highlighted the need to define and maintain a strategy for reuse (OS.SKM.a7), reuse the stored knowledge (OS.SKM.a8), and document the use of stored knowledge (OS.SKM.a9). On the other hand, the development of a safety information system to share knowledge in the organization (OS.SKM.a2) is emphasized by [3], [80] in the STATE-OF-THE-ART.

Finally, we propose two actions: *OS.SKM.a10 Notify users about problems, new versions and exclusions of artifacts in use*, and *OS.SKM.a11 Manage assets*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

TABLE II
ACTIONS OF UNI-REPM SCS (PART 1).

#	ID	Description	Level	Refs
	RE	Requirements Elicitation		
	RE.SM	Supplier Management		
2	RE.SM.a2	Identify and document the products to be acquired	1	
3	RE.SM.a3	Select suppliers and record rationale	1	
	DS	Documentation and Requirements Specification		
	DS.HF	Human Factors		
7	DS.HF.a1	Construct operator task models	1	[1], [60], [80]
8	DS.HF.a2	Document human factors design and analysis	1	[1], [60], [77], [80]
12	DS.HF.a6	Specify Human Machine Interface requirements	1	[60], [61]
	DS.SDO	Safety Documentation		
120	DS.SDO.a1	Record Safety decisions and rationale	1	
14	DS.SDO.a2	Ensure that safety requirements are incorporated into system and subsystem specifications, including human-machine interface requirements	1	[3]
16	DS.SDO.a4	Develop and document training, operational and software user manuals	1	[3], [7], [32]
17	DS.SDO.a5	Document System Limitations	1	[1]
	RA	Requirements Analysis		
	RA.PSA	Preliminary Safety Analysis		
25	RA.PSA.a3	Identify and document system hazards	1	[3], [21], [51], [71]
26	RA.PSA.a4	Identify and document hazards, hazardous situations and harmful events due to interaction with other equipment or systems (installed or to be installed)	1	[51]
28	RA.PSA.a6	Obtain and document information about the determined hazards (causes, probability, severity, duration, intensity, toxicity, exposure limit, mechanical force, explosive conditions, reactivity, flammability etc.)	1	[21], [32], [54]
29	RA.PSA.a7	Identify and document hazardous materials	1	[67]
30	RA.PSA.a8	Identify and document consequences of hazards, severity categories and affected assets	1	[1]
31	RA.PSA.a9	Conduct risk estimation	1	[21], [32], [51], [59]
32	RA.PSA.a10	Conduct risk evaluation for each identified hazard	1	[21], [32], [51], [59], [65], [67]
35	RA.PSA.a13	Identify and document pure safety requirements	1	[79]
39	RA.PSA.a17	Identify and document safety functional requirements	1	[21], [54], [59], [63]
40	RA.PSA.a18	Identify and document operational requirements	1	
42	RA.PSA.a20	Prioritize hazards and safety requirements	1	[32], [51]
	RA.FH	Failure Handling		
48	RA.FH.a3	Specify Restart-up procedures	1	[59]
49	RA.FH.a4	Document the system behavioral model	1	[61]
50	RA.FH.a5	Identify and document Common-Cause Failures (CCF) and how to prevent them	1	[64]
	RP	Release Planning		
	RP.SC	Safety Certification		
54	RP.SC.a3	Evaluate the threat to society from the hazards that cannot be eliminated or avoided	1	[3]
55	RP.SC.a4	Construct preliminary safety and hazard reports	1	[3]
59	RP.SC.a8	Document the division of responsibility for system certification and compliance with safety standards during safety planning	1	[21], [54]
60	RP.SC.a9	Specify a maintenance plan	1	[61]
	RV	Requirements Validation		
	RV.SVV	Safety Validation and Verification		
62	RV.SVV.a2	Define the safety verification plan	1	[3], [54], [64], [65], [69]
64	RV.SVV.a4	Define pass/fail criteria for accomplishing software validation and verification	1	[54]
65	RV.SVV.a5	Develop safety test plans, test descriptions, test procedures, and validation and verification safety requirements	1	[3], [32]
67	RV.SVV.a7	Validate safety-related software aspects	1	[51], [55], [61], [65]
68	RV.SVV.a8	Ensure that there is no potentially hazardous control actions	1	[1]
69	RV.SVV.a9	Perform safety evaluation and verification at the system and subsystem levels	1	[3], [66]
70	RV.SVV.a10	Conduct joint reviews (company and customer)	1	[51], [67]
72	RV.SVV.a12	Document discrepancies between expected and actual results	1	[54]
75	RV.SVV.a15	Perform safety inspections	1	[3], [32]
	OS	Organizational Support		
	OS.SP	Safety Planning		
77	OS.SP.a1	Develop an integrated system safety program plan	1	[3], [21], [71]
79	OS.SP.a3	Define and document the interface between system safety and all other applicable safety disciplines	1	[65]
80	OS.SP.a4	Delineate the scope of safety analysis	1	[3]
84	OS.SP.a8	Identify any certification requirements for software, safety or warning devices or other special safety feature	1	[51], [65], [66]
88	OS.SP.a12	Specify operating conditions of the machine and installation conditions of the electronic parts	1	[58]
89	OS.SP.a13	Determine the required performance level	1	[59]
90	OS.SP.a14	Identify and document the hazard analysis to be performed; the analytical techniques (qualitative or quantitative) to be used; and depth within the system that each analytical technique will be used (e.g., system level, subsystem level, component level)	1	[54], [65], [67], [80]
	OS.GSM	General Safety Management		
92	OS.GSM.a2	Identify and document safety lifecycle for the system development	1	[54]
95	OS.GSM.a5	Identify and document responsibility, accountability and authority	1	[3], [21], [54], [65], [80]
96	OS.GSM.a6	Define system safety program milestones and relate these to major program milestones, program element responsibility, and required inputs and outputs	1	[65], [69]
99	OS.GSM.a9	Monitor project and take corrective actions	1	[69], [70]
	OS.STO	Safety Tool support		
103	OS.STO.a4	Record information of the tools in the baseline	1	[61]
	PM	Requirements Process Management		
	PM.SCM	Safety Configuration Management		
118	PM.SCM.a1	Maintain accurately and with unique identification all safety configuration items and safety information (hazards, safety requirements, risks, etc.)	1	[32], [54], [69]
124	PM.SCM.a6	Perform safety impact analysis on changes	1	[21], [51], [54]
126	PM.SCM.a8	Document the procedures for initiating modifications to the safety-related systems, and to obtain approval and authority for modifications	1	[54]
128	PM.SCM.a10	Appoint all deliverable documents according to the rules defined in the Configuration Management Plan	1	[61]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

TABLE III
ACTIONS OF UNI-REPM SCS (PART 2).

	# ID	Description	Level	Refs
	PM.SCO	Safety Communication		
132	PM.SCO.a3	Establish a common nomenclature	1	[32]
139	PM.SCO.a10	Identify, record and resolve conflicts	1	[32]
	PM.ST	Safety Traceability		
146	PM.ST.a5	Define and maintain bi-directional traceability between system and subsystem verification results and system specification	1	[61], [62]
147	PM.ST.a6	Define and maintain bi-directional traceability between validation results and system specification	1	[61], [62], [82]
	RE	Requirements Elicitation		
	RE.SM	Supplier Management		
4	RE.SM.a4	Specify all external systems and safety-related software	2	[61], [63]
5	RE.SM.a5	Establish and maintain detailed system integration procedures for the external systems and safety-related software	2	[69], [81], [82]
6	RE.SM.a6	Define the safety standards that suppliers must follow	2	[51]
	DS	Documentation and Requirements Specification		
	DS.HF	Human Factors		
9	DS.HF.a3	Evaluate prototypes, requirements and technical UI restrictions	2	[51], [60], [78]
10	DS.HF.a4	Model and evaluate operator tasks and component black-box behavior	2	[7], [60]
11	DS.HF.a5	Define interfaces considering ergonomic principles	2	[60], [61], [63]
	DS.SDO	Safety Documentation		
15	DS.SDO.a3	Document all lifecycle and modification activities	2	[63]
18	DS.SDO.a6	Provide a safety manual	2	[32], [54]
20	DS.SDO.a8	Ensure that safety-related information is incorporated into user and maintenance documents	2	[3]
21	DS.SDO.a9	Maintain hazard and risk analysis results for the system throughout the overall safety lifecycle	2	[54]
22	DS.SDO.a10	Include a summary of safety requirements	2	[32]
	RV	Requirements Validation		
	RV.SVV	Safety Validation and Verification		
61	RV.SVV.a1	Define the safety validation plan for software aspects of system safety	2	[54], [69]
63	RV.SVV.a3	Define the technical strategy for the validation of external systems and safety-related software	2	[54]
66	RV.SVV.a6	Define and maintain a software integration test plan	2	[61], [69], [82]
71	RV.SVV.a11	Ensure that the stakeholders understand software-related system safety requirements and constraints	2	[3]
74	RV.SVV.a14	Ensure that software requirements and interface specification are consistent	2	[62], [69]
76	RV.SVV.a16	Identify and fix inconsistencies in safety requirements specification	2	[3], [69], [70]
	RA	Requirements Analysis		
	RA.PSA	Preliminary Safety Analysis		
23	RA.PSA.a1	Identify and document safety-critical computer software components and units	2	[3]
24	RA.PSA.a2	Simulate the process	2	[54], [80]
27	RA.PSA.a5	Specify the type of initiating events that need to be considered	2	[32]
33	RA.PSA.a11	Identify and document risk mitigation procedures for each identified hazard	2	[21], [67]
34	RA.PSA.a12	Collect safety requirements from multiple viewpoints	2	[51]
36	RA.PSA.a14	Identify and document safety-significant requirements and safety integrity levels	2	[79]
37	RA.PSA.a15	Identify and document safety constraints and how they could be violated	2	[1], [32], [54], [72], [73], [79]
38	RA.PSA.a16	Identify and document possible control flaws and inadequate control actions	2	[1], [72], [73]
41	RA.PSA.a19	Perform and document the feasibility evaluation of safety functional requirements	2	[3]
43	RA.PSA.a21	Identify and document analysis and verification requirements, possible safety-interface problems, including the human-machine interface, and operating support requirements	2	[3], [32]
45	RA.PSA.a23	Consolidate preliminary system safety technical specification	2	[61], [65]
	RA.FH	Failure Handling		
46	RA.FH.a1	Define requirements for the avoidance of systematic faults	2	[54]
47	RA.FH.a2	Specify Fault-detection procedures	2	[59]
51	RA.FH.a6	Perform reliability and system performance analysis	2	[80]
	RP	Release Planning		
	RP.SC	Safety Certification		
52	RP.SC.a1	Conduct safety audits	2	[3]
53	RP.SC.a2	Demonstrate the preliminary level of safety achieved by the system	2	[3]
56	RP.SC.a5	Construct preliminary safety cases	2	[3], [21]
57	RP.SC.a6	Demonstrate preliminary compliance with safety standards	2	[3], [65]
58	RP.SC.a7	Ensure that the hazard report is updated with embedded links to the resolution of each hazard, such as safety functional requirements, safety constraints, operational requirements, and system limitations	2	[1], [21], [32], [65]
	OS	Organizational Support		
	OS.SP	Safety Planning		
78	OS.SP.a2	Define and document requirements for periodic functional safety audits	2	[54]
81	OS.SP.a5	Establish the hazards auditing and log file	2	[3]
82	OS.SP.a6	Establish working groups and structures	2	[3]
83	OS.SP.a7	Define and document the regulations and safety standards to be followed	2	[21]
85	OS.SP.a9	Define and document requirements completeness criteria and safety criteria	2	[1], [21], [65], [69]
86	OS.SP.a10	Review safety experience on similar systems	2	[3], [51]
87	OS.SP.a11	Specify the general safety control structure	2	[1], [3], [65]
	OS.GSM	General Safety Management		
91	OS.GSM.a1	Identify and document the system development methodology	2	[3]
93	OS.GSM.a3	Identify and document competence requirements for the safety activities	2	[22], [54], [69]
94	OS.GSM.a4	Set safety policy and define safety goals	2	[3], [21], [71]
98	OS.GSM.a8	Prepare progress reports in a period of time defined by the project	2	[65], [69], [80]
	OS.STO	Safety Tool support		
100	OS.STO.a1	Use of verification and validation tools	2	[61]
101	OS.STO.a2	Specify justifications for the selection of the off-line support tools	2	[61]
102	OS.STO.a3	Assess offline support tools which can directly or indirectly contribute to the executable code of the safety related system	2	[54]
104	OS.STO.a5	Use of tools with support to cross reference and maintain the traceability among safety information in the software specification	2	[54], [61]
105	OS.STO.a6	Use of computer-aided specification tools	2	[51], [54]
	OS.SKM	Safety Knowledge Management		
111	OS.SKM.a5	Document a strategy to manage the knowledge	2	[18], [80]
112	OS.SKM.a6	Define a lifecycle for projects artifacts	2	[54]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

TABLE IV
ACTIONS OF UNI-REPM SCS (PART 3).

#	ID	Description	Level	Refs
	PM	Requirements Process Management		
	PM.SCM	Safety Configuration Management		
119	PM.SCM.a2	Define and document change-control procedures	2	[21], [54]
121	PM.SCM.a3	Define and document safety configuration items to be included in the baseline	2	[69]
122	PM.SCM.a4	Document configuration status, release status, the justification (taking account of the impact analysis) for and approval of all modifications, and the details of the modification	2	[54]
123	PM.SCM.a5	Document the release of safety-related software	2	[22], [54]
125	PM.SCM.a7	Specify and follow the template for software modification request	2	[54]
127	PM.SCM.a9	Maintain and make available the software configuration management log	2	[54], [69]
	PM.SCO	Safety Communication		
131	PM.SCO.a2	Define a method of exchanging safety information with the suppliers	2	[65]
133	PM.SCO.a4	Train people continuously in system engineering and safety techniques (education)	2	[3], [21], [80]
135	PM.SCO.a6	Keep stakeholders updated regarding the progress of all safety-related activities	2	[21], [76]
136	PM.SCO.a7	Construct a repository of common hazards	2	Proposed
137	PM.SCO.a8	Define and follow templates for system artifacts	2	[32], [76]
138	PM.SCO.a9	Document how conflicts will be resolved	2	[1], [32]
140	PM.SCO.a11	Produce all the deliverables documents based on the official document templates	2	[32], [61]
141	PM.SCO.a12	Make available safety-related software specification to every person involved in the lifecycle	2	[63]
	PM.ST	Safety Traceability		
142	PM.ST.a1	Define and maintain traceability policies	2	[32]
143	PM.ST.a2	Define and maintain bi-directional traceability between the system safety requirements and the software safety requirements	2	[54]
144	PM.ST.a3	Define and maintain bi-directional traceability between the safety requirements and the perceived safety needs	2	[1]
145	PM.ST.a4	Link and maintain bi-directional between environmental assumptions and the parts of the hazard analysis based on the assumption	2	[3]
148	PM.ST.a7	Define and maintain bi-directional traceability among system hazards into components	2	[82]
149	PM.ST.a8	Justify reasons for not traced software requirements	2	[62]
	RE	Requirements Elicitation		
	RE.SM	Supplier Management		
1	RE.SM.a1	Establish and maintain formal agreements among organization and suppliers	3	[21], [69]
	RA	Requirements Analysis		
	RA.PSA	Preliminary Safety Analysis		
44	RA.PSA.a22	Perform interface analysis, including interfaces within subsystems (such as between safety-critical and non-safety-critical software components)	3	[3]
	RV	Requirements Validation		
	RV.SVV	Safety Validation and Verification		
73	RV.SVV.a13	Verify the behavioral model	3	[61]
	DS	Documentation and Requirements Specification		
	DS.SDO	Safety Documentation		
19	DS.SDO.a7	Document lessons learned	3	[3], [65], [80]
	OS.STO	Safety Tool support		
106	OS.STO.a7	Define and use tools to support the safety process and workflow management	3	[51], [75]
	OS.SKM	Safety Knowledge Management		
107	OS.SKM.a1	Establish and maintain an infrastructure to share knowledge	3	[18], [51], [80]
108	OS.SKM.a2	Develop a safety information system to share knowledge in the organization	3	[3], [80]
109	OS.SKM.a3	Define control access mechanisms to the safety information system	3	[18], [68], [80]
110	OS.SKM.a4	Maintain employees' competence information	3	[54]
113	OS.SKM.a7	Define and maintain a strategy for reuse	3	[51]
114	OS.SKM.a8	Reuse the stored knowledge	3	[32], [51], [80]
115	OS.SKM.a9	Document the use of stored knowledge	3	[51]
116	OS.SKM.a10	Notify users about problems, new versions and exclusions of artifacts in use	3	
117	OS.SKM.a11	Manage assets	3	Proposed
	OS.GSM	General Safety Management		
97	OS.GSM.a7	Use of indicators on engineering documentation to assess the product properties and the development progress	3	Proposed
	PM	Requirements Process Management		
	PM.SCM	Safety Configuration Management		
129	PM.SCM.a11	Upload all documents on the safety information system	3	[61]
	PM.SCO	Safety Communication		
130	PM.SCO.a1	Establish formal communication channels among different organizational levels	3	[51]
134	PM.SCO.a5	Use of a common safety information system for system specification and safety analysis	3	[76]

2) SPA: Safety Tool support (STO): This sub-process is responsible for facilitating the appropriate execution of the corresponding tasks and manage all safety-related information that should be created, recorded and properly visualized. We included this SPA in the module considering that some SAFETY-STD request, for example, the specification of the reasons for the selection of the off-line support tools (OS.STO.a2) [61], the assessment of such tools that can directly or indirectly contribute to the executable code of the safety related system (OS.STO.a3) [54], the use of tools with support to cross reference and maintain the traceability among safety information in the software specification (OS.STO.a5) [54], [61].

In the INTERVIEW-STUDY [51] and STATE-OF-THE-ART [75], the requirement of defining and using tools to support the safety process and workflow management (OS.STO.a7) is presented.

3) SPA: General Safety Management (GSM): GSM covers project safety management activities related to planning, monitoring, and controlling the project. We defined this SPA

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

considering the practices presented in works of STATE-OF-THE-ART, SAFETY-STD, and EXISTING-MATURITY-MODS categories.

The works of STATE-OF-THE-ART cite, for example, the identification and documentation of the system development methodology (OS.GSM.a1) [3], responsibility, accountability and authority (OS.GSM.a5) [3], [80] as well as setting safety policy and defining safety goals (OS.GSM.a4) [3], [71].

SAFETY-STD [54], [65] and EXISTING-MATURITY-MODS [22], [69] require the identification and documentation of safety lifecycle for the system development (OS.GSM.a), the competence requirements for the safety activities (OS.GSM.a3), and defining system safety program milestones and relate these to major program milestones, program element responsibility, and required inputs and outputs.

4) SPA: Safety Planning (SP): This is one of the most important SPA in the module since it provides the safety practices and establishes a safety culture in the company. The development of a SCS requires a careful planning and safety analysis (described in next section). Therefore, the SP was proposed considering the actions in SAFETY-STD, EXISTING-MATURITY-MODS, STATE-OF-THE-ART, and INTERVIEW-STUDY.

Several actions are cited in different categories such as develop an integrated system safety program plan (OS.SP.a1) which is presented in the STATE-OF-THE-ART [3], [71] and EXISTING-MATURITY-MODS [21]. Moreover, the identification of certification requirements for software, safety or warning devices or other special safety feature (OS.SP.a8) is presented in the INTERVIEW-STUDY [51] as well as in SAFETY-STD [65], [66].

Finally, the identification and documentation of the hazard analysis to be performed; the analytical techniques (qualitative or quantitative) to be used; and depth within the system at which each analytical technique will be used (e.g., system level, subsystem level, component level) (OS.SP.a14) is required in SAFETY-STD [54], [65], [67] and STATE-OF-THE-ART [80].

5) SPA: Preliminary Safety Analysis (PSA): The PSA is the sub-process with more actions in the module since we grouped all actions related to performing a preliminary safety analysis. Someone may argue that some actions may be in the other SPA like elicitation for example. We agree that they may be included in such SPA, but we argue that the elicitation and analysis are generally done in parallel.

Furthermore, PSA is usually performed when some system functionalities are already elicited and documented, i.e. using a preliminary system specification as a basis. Therefore, we opted to classify all practices of safety analysis only in this group aiming to improve understandability of the module.

This SPA has practices from SAFETY-STD such as conduct risk evaluation for each identified hazard (RA.PSA.a10) [59], [65], [67], EXISTING-MATURITY-MODS like identify and document safety functional requirements (RA.PSA.a17) [21], [54], [59], [63], STATE-OF-THE-ART such identify and document safety constraints and how they could be violated (RA.PSA.a15) [1], [32], [72], [73], [79], and INTERVIEW-STUDY like identify and document system hazards (RA.PSA.a3) [51].

E. Module usage and tool support

Uni-REPM SCS aims to assess the safety maturity in the RE process; hence, it can be used by people who are involved in this stage of software development, deeply understand it and be in charge of process improvement in general.

Examples of users are:

- Requirements Engineer
- Safety Engineer
- Software Engineer
- Quality assurance engineer
- Project manager
- Product manager

The safety module was designed to be used through an assessement instrument as adopted by Uni-REPM [35]. The instrument has a query to evaluate each safety action in which the evaluator can select one of three options:

1) "Incomplete" (IC) - the action was deemed vital but was done partially or not done at all in the RE process.
2) "Complete" (C) - the action was done in the RE process.
3) "Inapplicable" (IA) - the action does not apply to the process.

Safety-critical systems are developed in several domains by companies with different sizes, infrastructure and maturity. Hence, some companies may not benefit from implementing all the actions in the module or some actions are deemed unnecessary to be done in particular situations of companies.

For example, in small systems, prototypes may be not useful since the system can be very simple. In this case, the action "*DS.HF.a2 Document human factors design and analysis (Basic Level)*" might not be useful for some companies. If we consider it as "Incomplete", the process may not reach the Basic level because not all actions in this level would be fulfilled. This would bias the company's evaluation results and would be more critical if all other actions in higher maturity levels were completed.

Accordingly, safety process of the companies should not be "devalued" if they do not perform a certain nonessential action (in their point of view). In order to consider situations like these, the option "Inapplicable" is provided. In this way, the module fits more real process and the evaluation result is less distorted.

Therefore, in some cases, the organization may find some actions only applicable in one of the settings. Whether an action is "Inapplicable" or not is solely based on the judgment of the project evaluator. Reasons for marking the action with this option should be considered carefully to avoid accidentally skipping an important action. Moreover, lack of time, resource or unawareness can not be used as a reason to mark an action as being "Inapplicable".

The assessment instrument is implemented in an online software tool aiming to facilitate and automate the evaluation process. The tool is available at www.unirepm.com. The tool supports three types of user: (1) external evaluator - this user can insert companies, projects and perform Safety/RE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

evaluations; (2) internal evaluator - this user only can insert projects to the company he/she belongs and perform Safety/RE evaluations; (3) admin - besides the functionalities of external evaluators, this user can manage users and different versions of RE/Safety models.

In Figure 4, we present an example of the Uni-REPM SCS assessment instrument. The Action ID in the checklist links the question(s) to the associated action in the model. This helps the users in case they need to locate the item for further information or clarification.

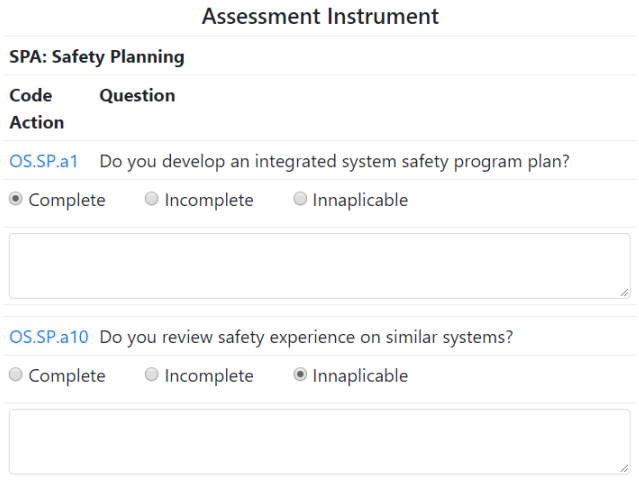
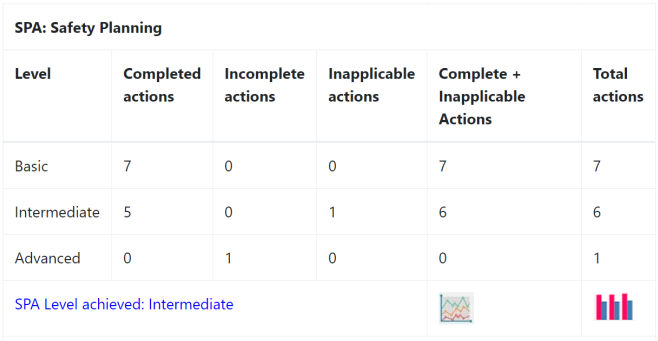


Fig. 4. Partial view of Uni-REPM SCS assessment instrument.



SPA: Safety Planning

Level	Completed actions	Incomplete actions	Inapplicable actions	Complete + Inapplicable Actions	Total actions
Basic	7	0	0	7	7
Intermediate	5	0	1	6	6
Advanced	0	1	0	0	1
SPA Level achieved: Intermediate					

Fig. 5. Assessment results of SPA "Safety Planning".

Fig. 6. Example of a graphical presentation of assessment results of SPA "Safety Planning".

After answering all questions present in assessment instrument, the tool determines and presents the evaluation results. To define the maturity level, the tool consider that: (1) for each SPA, all actions at a certain level must be Completed (or Inapplicable) in order to the MPA achieve such level; (2) for the whole process, all actions at a certain level must be Completed (or Inapplicable) to the process achieve such level.

There special cases where a SPA does not have actions at a certain level: (1) if the SPA does not have advanced actions, the maximum level possible is intermediate; (2) if the SPA does not have basic actions and the company completed at least one action at the intermediate level, it is classified as Basic, otherwise, the company is not classified as Basic and remains in the level Zero. The same principles are applied to MPAs and, consequently, to determine the maturity of entire project.

The evaluators should perform a careful analysis about the reasons that contribute to an action to be marked as incomplete, as they indicate which activities should be considered for process improvement efforts [35].

An example of evaluation results of SPA *"Safety Planning"* is described in Figure 5. The assessment results can be presented in a graphical way, as for example in Figure 6. These results do not correspond to real data, it is just presented with illustration purposes. Such representation provides a better view, allowing the organization to benchmark their maturity and to monitor their development.

It is important to highlight that the graphical presentation of Figure 6 does not include cumulative action counts. It only shows the total number of actions and the number of completed + inapplicable actions by maturity level. So, even the company may complete all actions of higher levels, for example intermediate level, but did not complete the actions of basic level, the company does not achieve the basic level, remaining at level zero.

The blue line presents actions which were *Complete*. In this case, 7 actions in the Basic Level were *Complete*, 6 actions in Intermediate level and 0 actions in the Advanced level. The purple line presents *Complete* actions together with *Inapplicable*.

The distance between the blue line and the purple line is called the module lag, which represents the number of Inapplicable actions. Hence, the module lag is important as it indicates to what extent the module works for a specific company and the context of that company. In this case, the module lag (i.e. the absolute numerical limit) is fairly small with only 1 Inapplicable action. This means a high applicability of the module.

Besides, the green line presents the total actions that should be completed in *"Safety Planning"* SPA. For example, at Advanced level, there is 1 action that should be finished. The difference between the purple and green lines is important because it denotes the improvement area of the process. It shows how many additional actions should be conducted to achieve a certain level of maturity.

Overall, the graph denotes that, in this SPA, the process has not done all the actions at Advanced level. Hence, the SPA received Intermediate Level. In order to reach the Advanced

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

level, 1 more action has to be done. Similar work can be done with other SPAs to achieve the result for the whole process.

V. Static Validation

Aiming to validate Uni-REPM SCS and ensure that the model quality is suitable for companies to pilot, we performed a static validation following the technology transfer framework [85]. After we formulated the candidate solution, which was the version 0.1 of UniREPM SCS, we conducted a static validation. This type of validation proposes collecting experts' opinion in order to find out whether the knowledge in literature was reasonably transferred and presented in the model [94].

The static validation provides an early feedback that helps to identify potential problems without using industry resources [94]. Hence, it is possible to improve the candidate solution and prepare it for the next step - dynamic validation [85].

A. Static validation design

We conducted the validation considering the following aspects:

- Coverage: to make sure the module presents the necessary safety practices and to detect others that might not be captured in the sources of information used to propose the module.
- Correctness: to ensure the names and maturity levels of the proposed practices are correctly presented.
- Usefulness and Applicability: to collect the experts' opinion about the module application in industrial settings and to what extent.

Considering the aspects above, the validation design was guided by the research questions below:

RQ1: Does Uni-REPM SCS have a sufficient coverage of safety practices?

RQ2: To what extent is Uni-REPM SCS suitable for companies, in terms of its correctness, usefulness and applicability?

RQ3: What improvements can be done to Uni-REPM SCS based on the findings in RQ1 and RQ2?

To answer these research questions, we interviewed two industry practitioners and conducted a survey with nine domain experts that work in academia that have partnerships with industry. The subjects provided their opinion about the SPAs, actions, and their maturity levels to validate its accuracy and adequacy.

The interviews were conducted by face-to-face meetings and the survey consisted in contacting domain experts using self-administered questionnaires sent by email. All subjects answered the same profile and module evaluation questionnaires and received a copy of Uni-REPM SCS full description.

We adopted the self-administered questionnaire survey strategy because the subjects could spend their most convenient time to analyze the module and answer the module evaluation questionnaire. Besides, we collected subject's background aiming to better contextualize the feedback received. These questions aimed to extract information about their knowledge and experience in the area as well as to help to draw a better view about them.

The subjects were selected using random sampling after a previous analysis of their profile. We considered their research interest in SCS, embedded systems and RE, their experience in these areas, and if they have publications in these areas. We searched for experts from many sources (publications, personal recommendation) and contacted them through emails. After four months, 11 out of 137 experts contacted accepted to participate and returned their feedback.

B. Subjects' Profile

Eleven subjects participated in this validation. Considering their affiliation, they are from four countries: Brazil (seven subjects), Norway, Poland, and South Korea with one subject each. The majority of them has PhD (eight subjects) and three have Master's degree. In order to easily refer to expert's comments, an ID that consisted of S# was assigned to each subject.

They have experience in academia, industry (working or with partnership), research institute, and spin-off company as demonstrated in Table V. We have mainly academics participating in this static validation as required by the technology transfer model [85] that defines that we should formulate a candidate solution, evaluate it in academia and improve it before using/validating in industry. Although the subjects are mainly academics, many of them have previous industrial experience and also work actively with industry collaborators in the field. Considering that they have such experience, we asked them about the module usefulness and if they would adopt it to get preliminary insights.

Besides, the time of experience with safety-critical systems is relevant (mean of 7 years) which can increase our confidence in their opinion. It is important to highlight that Subject #4 did not inform (NI) the time of experience with SCS, but he/she does have experience as safety case editor.

TABLE V
Time of experience - by type and with safety-critical systems (years).

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	Mean
Academia		8	9	15	16	16	7	2	6	10	7	8.73
Industry	8	15							7	2	7	3.55
Research Institute	6		3		31							3.64
Spin-off company				3								0.27
Industry - Partnership						10						0.91
Experience in safety-critical systems	12	15	6	NI	6	16	7	2	6	6	1	7

The subjects have experience with different domains (see Figure 7) and several roles (see Table VI) which contributed to analyze the coverage of practices and to have indications of its applicability in different domains.

Observing the need of certification by regulatory entities of the developed systems, we also questioned the subjects' experience with safety standards which is depicted in Table

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

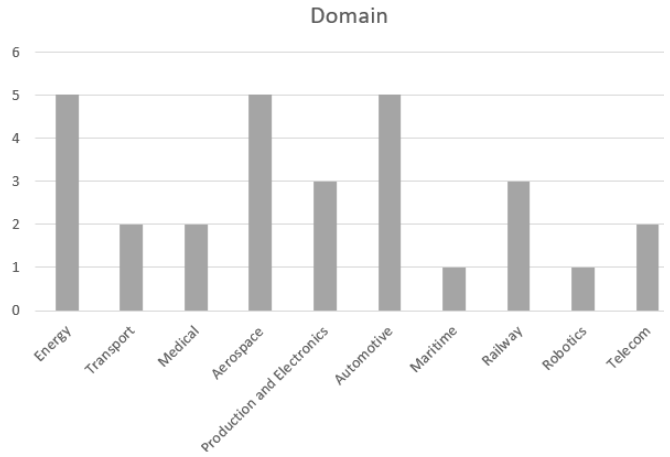


Fig. 7. Domain of experience.

TABLE VI
EXPERIENCE - ROLES.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	%
Requirements engineer		x	x	x				x	x		x	54.55
Designer (architecture and detailed design)	x	x						x				27.27
Developer/programmer	x				x	x			x		x	45.45
Tester				x							9.09	
Safety analyst/expert (internal to the company)		x	x	x	x			x				45.45
Independent assessor (consultant or external to the company)		x										9.09
Project leader or manager	x		x		x	x			x		x	45.45
Researcher	x		x	x	x	x	x	x		x	x	81.82
Teacher (Professor, lecturer etc.)		x	x	x	x	x	x		x	x	x	81.82
System engineer					x	x		x	x			36.36

VII. Except for two subjects, all of them have experience with standards. Subject #6 stated that "*I never get caught up in the rules during the developments. I have always been concerned with the problem itself and the rules of application, not development.*" This corroborates the question about the domains the subjects have experience and it is also an important result for this evaluation.

Questions about maturity models were also considered. First, we asked if the experts had already followed a maturity model (Figure 8). The majority did not follow as already expected since they are mostly from academia, and two subjects (S2 and S11) had followed.

It is important to note that subject S2 is one of the two practitioners interviewed and this expert had followed the CMMI [44] maturity model. S11 had followed the CMMI and MPS.BR [95] maturity models.

In case they have followed, we asked if the expert prefers

TABLE VII
EXPERIENCE WITH SAFETY STANDARDS.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
DO-178		x	x							x	
EN 50126			x					x			
EN 50128			x								
EN 50129			x								
FDA's guidelines for infusion pumps				x							
ISO 14971				x							
ISO 26262			x		x			x		x	
IEC 60601	x			x							
IEC 61508			x	x	x					x	
ISO/IEC 62279								x			
ISO 62304		x									
MIL-STD-498		x							x		
MIL-STD-882		x						x			
MIL-STD-2167									x		
None						x	x			x	

a generic maturity model or a specific maturity model. S2 answered "*I prefer a specific maturity model to avoid conflicts of interpretation*".

S11 said : "*Maturity models serve as guides for customizing processes and selecting practices. One problem with the most popular models (MPS.Br and CMMI) is because they are generic in nature. This makes them extensive, and this is reflected in the need for a great deal of effort to study and evaluate what should really be incorporated into each organization. More specific maturity models can provide a leaner and more cohesive set of recommendations and practices suggestion.*"

These statements contribute to the literature [21] [22] about the need of a specific maturity model for SCS.

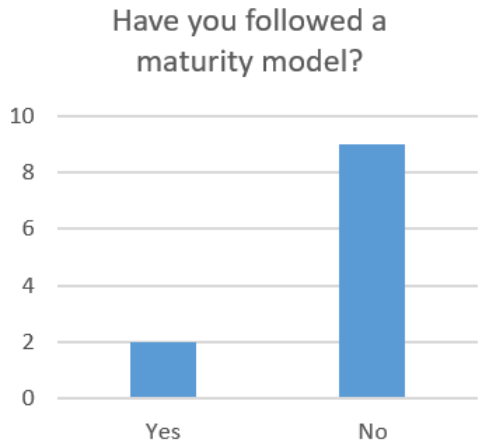


Fig. 8. Results of the question whether subjects had already followed a maturity model.

Regarding their opinion of the importance of such models, shown in Figure 9, the great majority (81.82%) considered them important, one expert did not answer and only one

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

considered it unimportant.

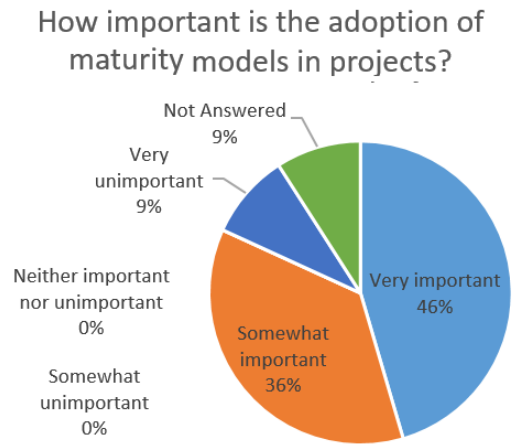


Fig. 9. Opinion of subjects about the importance of maturity models.

C. Results from module validation

After collecting the subjects' background, the module evaluation questions aimed to uncover improvement areas in terms of coverage, correctness, usefulness and applicability. They were about the sub-process areas, the coverage of practices, the maturity levels assigned and subjects' opinion about the module.

The first element we asked the experts was their opinion about the safety processes (or focus areas), presented in the module, they consider important and should be considered in RE process. Table VIII shows these results.

TABLE VIII
OPINION OF EXPERTS ABOUT THE SPAS OF UNI-REPM SCS.

SPA	Don't know (%)	Not Needed (%)	Desirable (%)	Essential (%)
Safety Knowledge Management			63.64	36.36
Safety Tool support	9.09	9.09	54.55	27.27
General Safety Management		9.09	18.18	72.73
Safety Planning		18.18	36.36	45.45
Safety Configuration Management	9.09	18.18	18.18	54.55
Safety Communication			54.55	45.45
Safety Traceability	9.09		36.36	54.55
Supplier Management	18.18	9.09	63.64	9.09
Preliminary Safety Analysis	9.09		36.36	54.55
Failure Handling	9.09		9.09	81.82
Safety Certification	18.18	18.18	45.45	18.18
Human Factors			27.27	72.73
Safety Documentation			45.45	54.55
Safety Validation and Verification			18.18	81.82

The results demonstrate that all SPAs in Uni-REPM SCS are considered desirable or essential by the majority of experts. This indicates that the reviews we performed and sources of information used are updated and reflect current industry needs.

The subjects reported if they think that other safety-related process are also important for the development of a safety-critical system. Our response actions to their answers, presented below, are discussed in Section V-D.

S1: *The existing usability needs to be considered in the development of a new critical system. The change in the way of using the system can generate failures.*

S5:*Environment description safety is always a relationship between a system and its environment.*

S6: *Durability tests - a fully functional and bug-free prototype can fail in a short time due to poor component quality or wrong handling in the welding process, for example.*

S8: *The proposed module supports reliability based safety engineering along with system thinking for safety critical systems but still some process are missing while generating inadequate control actions in Requirements Analysis.*

S10: *Since safety standards in both aerospace (DO-178C and SAE ARP 4754A) and automotive (ISO 26262) domains recommend or mandate the development of an Assurance/Safety Case (Kelly, 2003) as a requirement for certification of a safety-critical system, The Safety Assurance process should be considered during the development of a given safety-critical system to obtain certification credits. An Assurance Case is a clear, comprehensive, and justifiable argument supported by a body of evidence that a system is acceptably safe to operate in a particular context.*

S11: *No, the process relationship is already comprehensive.*

The easiness of understand the SPAs in the module in experts opinion is presented in Figure 10.

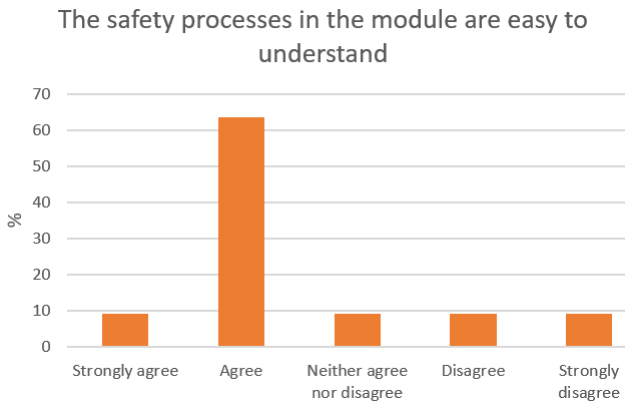


Fig. 10. Opinion of subjects whether the SPAs are easy to understand.

We observed that great majority (72.73%) agree that the SPAs are easy to understand. This was a concern we have when defining the SPAs since we wanted to adopt the terms that are in fact used in the SCS domain. This may be an indication that such goal was achieved.

The subjects opinion about the practices of Uni-REPM SCS is shown in Table IX.

All experts, including the industry practitioners interviewed, considered the great majority of safety practices of the module desirable or essential. The exceptions were S4 that did not

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

TABLE IX
OPINION ABOUT THE PRACTICES.

Subject	Not Answered	Don't know	Not Needed	Desirable	Essential
S1	0	10	1	62	75
S2	0	0	0	24	124
S3	0	0	0	62	86
S4	148	0	0	0	0
S5	0	0	14	41	93
S6	0	10	23	65	50
S7	0	146	0	0	2
S8	1	0	11	80	56
S9	4	18	8	69	49
S10	0	0	2	24	122
S11	0	12	0	70	66

answer this question and S6 that opted to not answer once he/she said that "*I do not have practical experience to answer this question*".

We asked the subjects whether there are actions important for RE process of safety-critical systems that are missing in the module. Our response actions to their answers, described below, are discussed in Section V-D.

S1: *Verification of existing products.*

S3: *For assessing the completeness of the module, safety standards could be checked. For example, some prescribe traceability between requirements and code.*

S4: *Security analysis - it is hard to find a domain/application where those 2 topics wouldnt interrelated. Here, the main focus is: can safety be compromised by malicious actions.*

S5: *Formal communication is mentioned several (3) times in the document. However IMHO, the informal communication in a project is more important, mainly because it is much more frequent.*

S8: *In my view, all the processes are well reflected in proposed module.*

S9: *It would be interesting to define a set of basic requirements that should be verified in each new software version (shakedown). There is a compromise between cost and coverage so there is a minimum test for releasing a new version.*

The easiness of understanding the actions in the module in experts opinion is presented in Figure 11. We noticed that the actions were considered as easy to understand by the subjects. We believe that the understanding of actions across domains is not compromised since we provide explanations for each action and examples. This is an important result since most of them analyzed the actions considering only their names and not the description. We already expected that they would not analyze the full description once the module has 148 actions (resulting in a long questionnaire) and the experts' schedules are busy. Therefore, we did not ask them how many times they consulted the actions' descriptions.

The maturity level we defined to each action was also evaluated by the subjects. The number of changes proposed by

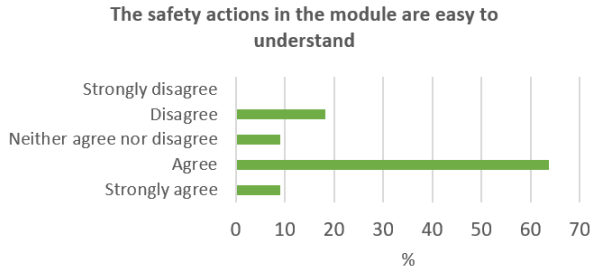


Fig. 11. Opinion of subjects whether the actions are easy to understand.

them is presented in Table X. Excepting S4, S5, S8, and S11 there was a tendency in the experts to reduce the maturity level we proposed to the actions. This occurred mostly with the two practitioners (S1 and S2) that during the interviews constantly repeated that the actions are the common procedure and are required by safety standards. Nevertheless, we observed a tendency of academic experts in proposing to increase the maturity level. It is important to highlight that the total number of changes is higher than 148 once many suggestions referred to the same action by different subjects.

TABLE X
SUGGESTIONS ABOUT THE MATURITY LEVEL OF THE PRACTICES.

Subject	Proposes to increase the maturity level	Proposes to decrease the maturity level	Total number of changes
S1	1	17	18
S2	0	15	15
S3	11	37	50
S4	30	28	58
S5	40	9	50
S6	0	0	0
S7	0	2	2
S8	34	6	41
S9	1	0	1
S10	26	5	31
S11	0	0	0
Total	143	119	262

During the analysis of subject's opinions, we observed some conflicting suggestions for the same action (increase and decrease). The number of actions in conflict per SPA is listed in Table XI. The number of conflicts was low, only 29 actions in a total of 148. We did not believe that there were misunderstandings during the interpretation of an action, however, this may be occurred due to the differences in subjects' experiences.

To answer RQ2, a query about the extent they believe the safety module will help requirements engineers to perform safety-related activities or tasks in the project (Figure 12) and whether they would adopt the module (Figure 13) was made to the experts.

We observed that they consider the contribution of Uni-REPM SCS significant to industry and they would adopt it case they would work in industry. Considering that the majority

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

TABLE XI
NUMBER OF ACTIONS IN CONFLICT BASED ON SUBJECTS' OPINION PER SPA.

SPA	# actions with conflicts
Safety Knowledge Management	3
Safety Tool support	0
General Safety Management	1
Safety Planning	1
Safety Configuration Management	2
Safety Communication	0
Safety Traceability	3
Supplier Management	0
Preliminary Safety Analysis	10
Failure Handling	0
Safety Certification	0
Human Factors	0
Safety Documentation	2
Safety Validation and Verification	7
Total	29

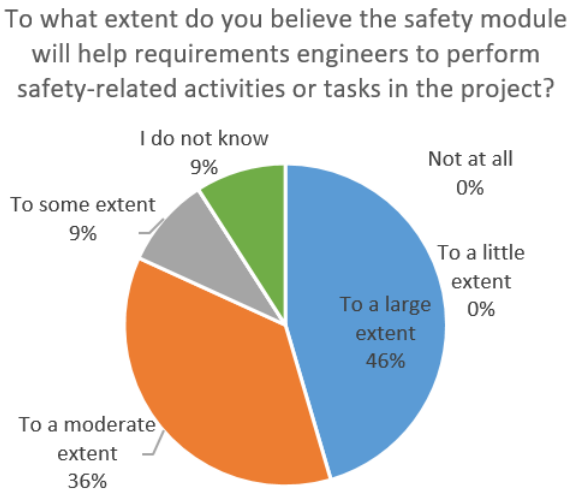


Fig. 12. Opinion of subjects about the usefulness of Uni-REPM SCS.

of experts are from academia, we have many answers of inapplicable.

D. Response Actions and Model Improvements

The feedback results were analyzed, response actions were decided and the model was refined and improved (RQ3).

1) Regarding new SPAs: We discarded the suggestions to add new SPAs:

- usability needs (S1) - there is no need to add a new SPA specific to the issues because they are already addressed in the Human Factors SPA;

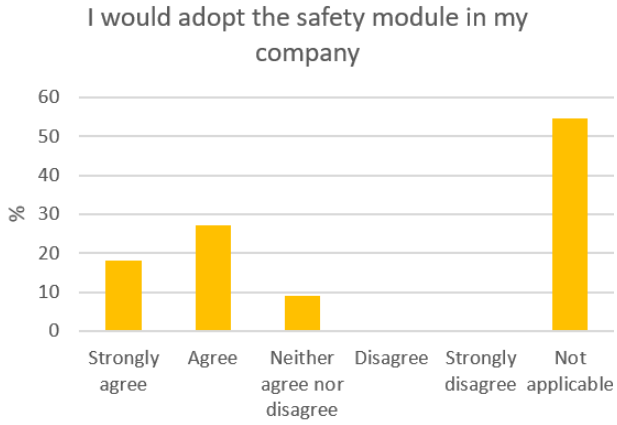


Fig. 13. Opinion of subjects about the adoption of Uni-REPM SCS.

- environment description (S5) - there is no need to add a new SPA to handle this because preliminary safety analysis, failure handling, and safety planning SPAs have already practices related to this comment;
- durability tests (S6) - because the required performance level is presented in the safety planning SPA;
- inadequate control actions (S8) - because preliminary safety analysis SPA already contain practices to handle this comment.
- safety assurance (S10) - the module already has the safety certification and safety planning SPAs that contain practices related to safety cases.

The subjects did not considered that the SPAs are insufficient or deficient for usability needs or environment description since they did not propose to add new actions. The comments are related to separate these issues in specific SPAs.

2) Regarding new actions: We discarded two suggestions to add actions that are: traceability between requirements and code (S3) - we are concerned with RE phase and code is produced in later stages; and informal communications (S5) - because it is a practice already performed in any project and generally SCS projects are large and involve several stakeholders that it would be impossible to manage informally. In a project that formal communication are not necessary, the company can mark it as inapplicable.

Some comments were left as future work considering the estimated resource and time taken to implement them. They are: verification of existing products (S1); security analysis (S4) that although is out of Uni-REPM SCS scope it is related with safety; and requirements for shakedown testing (S9) which basically refers to the fast/high level testing that is performed to an application after it has been migrated or deployed to a given environment to assure that is up and running without major glitches, which means that is ready for being tested.

Finally, according to S8: *In my view, all the processes are well reflected in proposed module*. Considering the experts' comments about the actions, we believe that the module has a good coverage of safety practices. These results contribute to increase our confidence in its coverage (RQ1). However, we highlight that this aspect will be analyzed again in future

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

studies that we will conduct with companies in the dynamic validation.

3) Regarding the maturity level of actions: Considering the goal of the model and the knowledge acquired in the several sources of information used, we tried to accommodate as much feedback as possible. As a result, 98 out of 120 suggestions to change the maturity level of practices were implemented. In Table XII, we show the number of implemented changes in the maturity levels of actions by SPA.

The criteria adopted to decide if we should implement the suggestion were:

- unanimity: all subjects, that proposed changing the action, suggested the same maturity level. - majority: the majority of subjects, that proposed changing the action, suggested the same maturity level. - agree: the suggestion was made by only one expert, but we agree. - disagree: we did not implement because we disagree with the subject.

In case of conflicts, i.e different experts suggested different maturity levels, we discussed what response action we would adopt.

VI. Discussion

The module construction process started with the investigation of literature available about RE in SCS. Accordingly, we performed SLRs (SLR1, SLR2) [2], [14], [50] to search the problem domain, learn the concepts involved as well as to explore the problems in the integration of these two areas.

With this investigation, we have gained the necessary knowledge to be able to select the information sources of the safety practices proposed in Uni-REPM SCS. The sources, see Table I, comprised well-known authors of the field (STATE-OF-THE-ART), international standards (SAFETY-STD), existing maturity models (EXISTING-MATURITY-MODS), and empirical studies (INTERVIEW-STUDY, TECH-REPORT). Hence, we chose sources from academia and industry.

During the selection of safety actions/practices, we considered the definition of requirements practice of Davis and Zowghi [96]. They classify requirements practices as the adoption of a principle, tool, notation, and/or method in order to perform a RE activity. When a practice reduces the cost of the development project or increases the quality of the resulting product, it is labeled as good requirements practice [30], [96].

In this context, we select safety practices capable of raising the likelihood that the right system will be built [96]. The practices are presented in II, III, and IV.

To define the architectural structure of our module, we have identified a set of requirements and characteristics presented in maturity models, described in Section III-A4. Accordingly, we considered such characteristics as well as the dual-view-approach of Uni-REPM: Process Area view and a Maturity Level view.

The process area view allows to visualize the hierarchy of process that consists the model and consult the practices of the same group. The maturity level view, on the other hand, classifies the practices by level, in which the actions at a level supports each other as well as the more advanced practices on the next level [35]. The safety practices were organized according to actions in sub-process areas, to separate activities that belong to the same group.

We did not perform any change in the seven (Requirements Elicitation, Requirements Analysis, Documentation and Requirements Specification, Requirements Validation, Requirements Process Management, Release Planning, Organizational Support) main process areas of Uni-REPM that were defined considering well-adopted RE processes.

From the list of safety actions/practices previously elicited, we group them in fourteen sub-process areas (Safety Planning, Supplier Management, Preliminary Safety Analysis, Failure Handling, Safety Validation and Verification, Safety Certification, General Safety Management, Safety Configuration Management, Safety Communication, Human Factors, Safety Tool support, Safety Documentation, Safety Traceability, Safety Knowledge Management).

In the Maturity Level View, we assigned a level (“Basic”, “Intermediate”, and “Advanced”) to each action considering the difficulty to implement the action, how essential it is for the RE process, and dependencies among them [35], the frequency of their appearance in different information sources as well as the ability of optimizing the safety processes considering our experience and the results of literature reviews.

Our module has 148 safety actions to be adopted in the RE process. We seek to propose a lean model, but complete considering all activities necessary to develop and maintain a safety-critical system. The module is focused on what to do instead of how to do. This approach provides flexibility for different companies to use established “in-house” procedures or processes.

For example, we say that hazards, safety requirements, accidents, risks, and other concepts must be documented. However, it is out of the module scope to prescribe which technique the company should use to elicit these information, the safety analysis method it will adopt (FTA, GSN, STAMP, etc) or how the company will document the requirements (natural language, model-based, which language etc).

Therefore, our aim is to provide clear guidance about which practices should be adopted in different maturity levels in SCS development. By providing a detailed set of practices, we address some critics that maturity models do not look deeply enough into all organizational practices [30].

It is important to note that it is not our goal with the module to provide evidence of meeting regulatory requirements/standards. First, some of them have domain-specific requirements that are not covered by our proposal. There are several works of safety evidence, traceability and certification in the literature. Our aim is to improve the development process by addressing many safety practices early in the development process, i.e. adopt practices in Requirements Engineering phase, instead of handling in in later stages of software development. Since they will be handled in the beginning, many artifacts and evidence will be produced and documented when satisfying the practices in the module. But, this is an indirect contribution.

Independently of domain, companies can and do develop successful systems without maturity models. However, litera-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

TABLE XII
IMPLEMENTED CHANGES IN MATURITY LEVELS BY SPA.

SPA	# of actions	# of suggested changes			# of implemented changes		
		Increase	Decrease	Total	Increase	Decrease	Total
Safety Knowledge Management	11	1	8	9	1	8	9
Safety Tool support	7	-	4	4	-	4	4
General Safety Management	9	4	2	6	3	2	5
Safety Planning	14	6	2	8	3	2	5
Safety Configuration Management	12	3	7	11	3	6	10
Safety Communication	12	6	3	10	5	3	9
Safety Traceability	8	-	8	8	-	8	8
Supplier Management	6	4	2	6	4	2	6
Preliminary Safety Analysis	22	19	2	21	3	2	12
Failure Handling	6	3	2	5	3	2	5
Safety Certification	9	5	2	7	2	2	4
Human Factors	6	2	2	4	1	2	3
Safety Documentation	10	5	4	9	4	4	8
Safety Validation and Verification	16	5	7	12	4	6	10
Total	148	63	55	120	43	53	98
%		42.57	37.16	81.08	68.25	96.36	81.67

ture reports [20], [97] that deadlines and budgets are routinely exceeded, resources are wasted, and there are lots of rework. For very large systems that include separate subsystems such as SCS, developed by teams who may be working in different locations, an important factor that affects product quality is the software process. The major problems with large projects are integration, project management, and communication [97]. There is usually a mix of abilities and experience in the team members and, because the development process usually takes place over a number of years, the development team is volatile. It may change completely over the lifetime of the project.

Accordingly, our goal in proposing this maturity model is to contribute to a company to evaluate its strengths and weaknesses, to develop improvement plans when compared to other organizations' standards and best practices. The maturity model we propose is more detailed, and useful because it was designed specifically for SCS and contains comprehensive assessment instrument. This model can be used to help an organization defining what to improve or implement next in order to make their processes more efficient.

The assessment of the RE processes of the companies will be performed in future studies. Thus to what extent the model can help companies to improve their development and to what extent an improved RE process can alleviate some of the typical issues associated with inadequate RE will be clarified, and might also add discrete value towards using the module even if we are not focused on "assuring compliance".

A. Static Validation

Our experience confirmed that planning, preparing and executing validations require extensive effort [94]. Six months were needed to elaborate a careful design to ensure that nothing important was missing and no error would be made; selecting and contacting experts; and analyzing the results.

The aim of the static validation was to identify possible improvements that can be done to Uni-REPM SCS. With the help and feedback from two practitioners and nine experts from academia coming from various countries with diversified expertise, significant improvement in the module were performed. With this validation, we evaluated all aspects of the module's structure (SPAs, actions and maturity levels).

The module was analyzed in terms of correctness, coverage, usefulness and applicability. The majority of the suggestions were regarding the maturity levels of actions. We addressed 98 of 120 suggestions and an revised version of the module was generated and it available in the project website www.unirepm.com. We did not perform follow-up interviews with the subjects nor different set of experts to obtain feedback about this revised version. We plan to perform this in future studies.

We acknowledge that there is some subjectivity in deciding which actions would be included in the module and the maturity level assigned. However, this is an issue of any proposed maturity model. Well-adopted maturity models such as ISO 15504, CMMI, and MPS.BR and bodies of knowledge such as PMBOK just state that their set of practices reflects current industry best practices. Moreover, they are frequently updated and new practices/actions are added or removed in each release. Thus, maturity models should be seen as constantly evolving just like all models of this type.

We believe that domain-dependent maturity models are not feasible since safety-critical systems share many practices as present in different safety-standards. Related solutions such as ISO 15504-10, +Safe-CMMI are domain-independent as well. If a practice is not applicable in a particular context, the evaluator can mark it as Inapplicable. This "feature" is built into the framework of this model.

Moreover, the improved model was fairly complete as there

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

were few suggestions about adding new actions. According to the experts' opinions, the actions in the model are applicable in real settings. Therefore, our research questions were answered and our goal to validate the module was successfully achieved.

Finally, the purpose of this model is to present all the good practices that give an organization ideas to improve. However, it is the organization's responsibility to decide whether the recommended practices are indeed beneficial and suitable and when to implement them. We observed in our interview with one of the practitioners that the actions that most attracted him/her were those of the subprocesses "Safety Tool support" and "Safety Traceability". The reasons were that the company does not yet use a support tool for requirements engineering or safety analysis; and they do almost nothing in terms of traceability. Therefore, we noticed that they have learned these practices after participating of Uni-REPM SCS' validation process.

B. Comparison with related solutions

Maturity models that explicitly address safety in RE process could not be identified. Instead, some maturity models could be detected that address RE and safety engineering as separate fields. In Tables XIII and XIV, we presented a comparison among them.

The Uni-REPM Safety module differs from existing safety maturity models such as +SAFE-CMMI-DEV [21], and ISO 15504-10 [22] in terms of purpose, scope, intended usage and number of practices.

+SAFE-CMMI-DEV, developed in 2007, is an extension to CMMI for Development (CMMI-DEV) developed for standalone use, i.e. it is not intended to be embedded in a CMMI model, but can be modified to support different safety standards. It covers two process areas that have Generic Goals, Specific Goals and Specific Practices: Safety Management and Safety Engineering [16].

The Generic Goals of *Safety Management* are Achieve Specific Goals (1 practice), Institutionalize a Managed Process (10 practices), Institutionalize a Defined Process (2 practices), Institutionalize a Quantitatively Managed Process (2 practices), and *Institutionalize an Optimizing Process* (2 practices). Specific Goals include Develop Safety Plans (4 practices), Monitor Safety Incidents (1 practice), and Manage Safety-Related Suppliers (2 practices).

The Safety Engineering process area has the same five Generic Goals (and their practices) of Safety Management. Its Specific Goals are Identify Hazards, Accidents, and Sources of Hazards (2 practices), Analyze Hazards and Perform Risk Assessments (1 practices), Define and Maintain Safety Requirements (3 practices), Design for Safety (3 practices), and Support Safety Acceptance (4 practices).

ISO/IEC 15504-10 [22], developed in 2011 as a standalone document, has be conceived to be used in conjunction with ISO/IEC 15504-5 (An exemplar Process Assessment Model) and/or ISO/IEC TR 15504-6 (An exemplar system life cycle process assessment model) process assessment models by experienced assessors with minimal support from safety domain experts [98].

The structure of ISO/IEC 15504-10 has the same process areas of +SAFE-CMMI-DEV. However, the number of practices is higher. *Safety Management* has 10 base practices, *Safety Engineering* has 11 practices. Moreover, it has one more process than +SAFE-CMMI-DEV: *Safety Qualification* [16] with 5 practices. Finally, it claims that the defined processes are consistent with five different safety standards: IEC 61508, +SAFE-CMMI-DEV, ISO 26262, IEC 60880, UK MoD Def Stan 00-56.

None of the models are intended to be used as part of a product assessment. While safety standards require the definition of safety integrity level of the system under development to set requirements for the project and the system, the maturity models provide a way to evaluate the capability of safety-related processes as well as a scheme for their improvement [98]. Hence, the maturity level achieved is not related with the safety integrity level the project has to fulfill. Accordingly, an evaluation based on any of these models is not analogous to a functional safety assessment [21].

Besides, standards do not have the feature of making possible for practitioners and a company to state "*this specific practice is not relevant for us*". Safety maturity models do, and thus, the model applicability to a real industrial context is better. Therefore, we share the view of +SAFE-CMMI-DEV and ISO 15504-10 that there is no relationship between safety integrity levels and maturity levels.

Another common feature of the maturity models is that they do not prescribe any specific technique, method or tool. Their goal is to consider the process (the "what") and not require the adoption of any specific technique or method (the "how") [98].

+SAFE-CMMI-DEV and ISO/IEC 15504-10 cover the entire project lifecycle. Hence, they do not go into detail into any particular practice area, such as RE. In introduction section, we discuss presenting references which show that requirements problems have been associated with many accidents and safety incidents. The need of integrating safety and RE teams has been well discussed by very seminal papers in SCS area, and now a consensus in academia and industry is being established that addressing safety concerns early in software development contributes to ensuring that safety problems do not propagate through subsequent phases. Furthermore, the early consideration of safety concerns in RE should be a top priority in the development of SCS since RE is essential for software quality, and the effectiveness of the software development process. Moreover, high safety levels are typically best achieved by addressing safety from the beginning; not by trying to add protection components and additional complexities after system has been developed. Accordingly, such requirements issues tend to be mitigated in companies with high process maturity levels since they do their business in a systematic, consistent and proactive approach.

Uni-REPM SCS proposes 148 safety actions while +SAFE-CMMI-DEV has 20 in which 13 actions have a correspondence with our model; and ISO/IEC 15504-10 has 26 actions being 16 present in Uni-REPM SCS. The other actions involve later stages of system development that are not the scope of our model. These demonstrate that Uni-REPM SCS has a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

TABLE XIII
COMPARISON AMONG UNI-REPM SCS, +SAFE-CMMI-DEV, AND ISO 15504-10 (PART 1).

	Uni-REPM SCS	**+SAFE-CMMI-DEV**	**ISO 15504-10**
Year	2017	2007	2011
Motivation for creating the model	In order to ensure a well-ordered safety progress, engineers should handle several features (e.g. organizational, technical, strategic). Thus, companies should improve their RE process with the purpose of overcome the difficulties they face during the construction of SCS. Requirements engineers need systematic guidance to consider safety concerns early in the development process.	The extension was developed because the Australian Defense Materiel Organization recognized that CMMI is a generically structured framework that requires amplification for specialized areas of engineering such as safety engineering. Developing safety-critical products requires specialized processes, techniques, skills, and experience within an organization.	ISO/IEC 15504 process assessment models for systems and software do not currently provide a sufficient basis for performing a process capability assessment of processes with respect to the development of complex safety-related systems. Developing safety-related systems requires specialized processes, techniques, skills and experience. Process amplifications are needed in the area of safety management, safety engineering and the safety qualification.
Full Available to download	Yes	Yes	No
Independent of any domain-specific standard	Yes	Yes	Yes
Intended usage	RE phase	System lifecycle	System lifecycle
Independent of safety integrity level	Yes	Yes	Yes
Goals of the model	It aims to reduce issues in RE during SCS development by addressing safety actions/practices that should be covered in the RE process to reduce the gap between these areas.	Its purpose is to extend CMMI to provide an explicit, specific framework for functional safety with respect to developing complex safety-critical products.	It presents these amplifications (a safety extension) as three process descriptions to provide additional life-cycle verification activities related to the methods and techniques selected relevant to safety requirements adopted and tailoring guidance for users intending to use the safety extension as part of a process assessment.
Number of levels	3 (Basic, Intermediate, Advanced)	5 (Initial, Managed, Defined, Quantitatively Managed, Optimizing)	5 (Initial, Managed, Defined, Quantitatively Managed, Optimizing)
Lowest maturity level	It contains actions of primitive but repeatable RE process. Although the actions are basic, the RE process in this level is defined and followed.	The processes are usually ad hoc and chaotic. The organization usually does not provide a stable environment to support processes. Maturity level 1 companies are characterized by a tendency to over commit, abandon their processes in a time of crisis, and be unable to repeat their successes.	The process achieves the objectives in some way and generates the expected work products.
Highest maturity level	It is the most mature RE process that cover predefined and structured procedures as well as pays adequate attention to future processes and work products.	An organization focuses on continually improving process performance through incremental and innovative process and technological improvements. The organization's quality and process performance objectives are established, continually revised to reflect changing business objectives and organizational performance, and used as criteria in managing process improvement.	The process, in addition to being executed, managed, defined and executed within quantitative limits, can be continuously improved.

good coverage of safety practices when compared with related solutions.

Therefore, the maturity model we propose is more descriptive and detailed because it was designed specifically for safety in RE and contains a comprehensive assessment instrument. Finally, we included in the module the most cited practices of Sommerville and Sawyer [32].

VII. CONCLUSIONS

Requirements engineers need systematic guidance to consider the safety concerns early in the development process of a safety-critical system.

The module proposed in this paper describes principles and practices that form the basis of safety processes maturity. Our aim is to help companies that develop SCS to improve the maturity of their processes in terms of an evolutionary path from chaotic and eventual processes towards mature and disciplined software processes.

We are planning for a dynamic validation, i.e. with practitioners in real contexts to evaluate its usefulness in practice since we have evaluated it in "theory" and the preliminary results suggest that this is a good solution so far. However, it is not possible to affirm the success of the module in industry. Hence, we are planning such validation and companies are been contacted in order to perform this. However, to convince

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

TABLE XIV
COMPARISON AMONG UNI-REPM SCS, +SAFE-CMMI-DEV, AND ISO 15504-10 (PART 2).

	Uni-REPM SCS	**+SAFE-CMMI-DEV**	**ISO 15504-10**
Evaluated Capabilities	Safety Knowledge Management; Safety Tool support; General Safety Management; Safety Planning (SP); Safety Configuration Management; Safety Communication; Safety Traceability; Supplier Management; Preliminary Safety Analysis; Failure Handling; Safety Certification; Human Factors; Safety Documentation; Safety Validation and Verification	Safety Management; Safety Engineering	Safety Management process; Safety Engineering process; Safety Qualification process
Total Number of practices	148	53 being 20 (safety-related)	26
Number of practices in common with Uni-REPM SCS	-	13	16
Strengths	It is a very detailed maturity model because it was designed specifically for safety in RE and contains a comprehensive assessment instrument.	It covers the entire project lifecycle. This extension was developed for standalone use. It is not intended to be embedded in a CMMI model.	It covers the entire project lifecycle.
Weakness	It covers only the activities of RE process.	It contains material that is fully redundant with CMMI to support its standalone use. It is too general, usually adopted by safety engineers, and do not consider the integration between safety and RE as well as the particularities of these two areas that are necessary to improve safety.	As well as +Safe, ISO 15504-10, it does not go into detail into any particular practice area from the beginning of software development process.
Safety standards considered	ISO 61508; ECSS-E-HB-40A; MIL-STD-882C; ISO/TS 15998-2; MIL-STD-882E; ISO 13849-1; ECSS-E-ST-40C; ISO 14639-1; MIL-STD-882D; ISO 13849-2; ISO 26262-6; ISO 15998; ISO/TR 14639-2	It is intended to be consistent with the Australian Defence Standard, Safety Engineering in the Procurement of Defence Systems, and is intended to be consistent with the principles of other contemporary safety standards (e.g., IEC's Safety of MachineryFunctional Safety of Safety-Related Electrical, Electronic and Programmable Electronic Control Systems; U.S. military standard, System Safety Program Requirements; the U.K. Defence Standard, Safety Management Requirements for Defence Systems, Part 1, Issues 2 and 3; and domain-specific safety standards wherever feasible).	It claims that the defined processes are consistent with the five different safety standards: IEC 61508, +SAFE, IEC 60880, UK MoD Def Stan 00-56, and ISO 26262.

practitioners to participate, we have to present a sound and refined proposal that is as good as we can make it based on state-of-the-art in academia and with indirect industrial applicability taken into consideration through researchers working with industry. This was the point of the static validation - collect early feedback and improve our proposal. Moreover, we have presented a comparison with related works (CMMI and ISO 15504-10) in order to clearly present the contributions of our work.

In the next sections, we discuss some contributions of this paper and point out some future research.

A. Contributions

Companies with high maturity levels tend to reduce requirements issues and make the system development process less challenging. Although maturity models are not "silver bullets", they may be used for several purposes providing useful benefits [35], [39], [40].

1) ***Benefits to academia****:* for researchers, the safety module offers an comprehensive summary of state of the art, by providing the identication and systematization of existing safety practices being a knowledge base. Besides, it may introduce new research opportunities on overlooked subjects or on validation of already existing topics [35].

2) ***Benefits to industry****:* for industry practitioners, the module provides a process evaluation model of safety concerns targeted at the RE process. Accordingly, it can guide requirements and safety engineers to develop SCS with high quality by providing a very practical structure with which to assess their maturity and reduce RE issues in the process.

3) ***Evaluation regarding specific concerns****:* Uni-REPM SCS addresses the problem space by identifying relevant safety actions and detailed factors that determine maturity of companies that develop SCS. Being structured in MPAs and SPAs, the module gives the potential to evaluate the maturity of whole RE process, but also specific areas in order to address the needs of several stakeholders.

4) ***Module can be used as a diagnostic tool****:* the module enables the determination of the current state ("as-is") of companies processes. The determination of company's *status quo* allows a better decision-making processs since it contributes to instruct managers regarding their current processes' and services' status [20]. Besides, such evaluation enables the elaboration of a roadmap for improving the domain position

from 'as-is' (where they are) to what they should do (the state 'to-be') [41].

*5) **Availability of assessment instrument**:* the safety module has an instrument to evaluate the maturity that is important for SCS development. It consists in a checklist allowing to provide a reminder of what to look for and reduce the chances of forgetting some safety action [8]. The instrument is fully supported by online software tool. Moreover, the subjects of the static validation agree that the safety practices/questions are easy to understand.

*6) **Tool Support**:* We designed and implement a software tool to support the usage of Uni-REPM and the safety module available at www.unirepm.com. The main features of the tool were discussed in the section IV-E. The tool has three types of users that can perform RE/Safety evaluations and all maturity levels achieved (SPA, MPA, and at project level) are calculated automatically. This tool support helps reducing time and effort in assessing the maturity.

*7) **Validation**:* The static validation we performed with the help of two practitioners and nine academic experts provided to us valuable contributions to our model and an improved version was elaborated and it is already available. The careful design contributed to collect relevant feedback, to assure that the model was of good quality and usable before its release.

In the next section, we suggest further research regarding maturity models for safety-critical systems.

B. Further research

This work has generated some research directions that should be explored in future efforts:

(1) What is the effect of applying Uni-REPM SCS when it is instantiated in different safety-critical domains?
(2) Which are the contextual factors that influence the maturity level of an organization?
(3) What is the level of acceptance of Uni-REPM SCS by practitioners?
(4) What is the module impact considering before/after assessments of companies?
(5) Can practitioners reconcile the actions in the module with what they already need to do in order to comply with mandatory standards in their application domain?
(6) Are there any conflicts among the module and the standards/existing company practices?

We intend to address such questions in future works.

ACKNOWLEDGMENT

This work was partially supported by FACEPE (Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco), and by a research grant for the ORION project (reference number 20140218) from The Knowledge Foundation in Sweden.

We also want to thank the subjects of module validation for their availability to contribute to our research.

REFERENCES

[1] N. Leveson, *Engineering a safer world: Systems thinking applied to safety*. Mit Press, 2011.
[2] J. Vilela, J. Castro, L. E. G. Martins, and T. Gorschek, "Integration between requirements engineering and safety analysis: A systematic literature review," *Journal of Systems and Software*, vol. 125, pp. 68–92, 2017.
[3] N. G. Leveson, *Safeware: system safety and computers*. ACM, 1995.
[4] R. R. Lutz, "Software engineering for safety: a roadmap," in *Proceedings of the Conference on The Future of Software Engineering*. ACM, 2000, pp. 213–226.
[5] R. Guillerm, H. Demmou, and N. Sadou, "Information model for model driven safety requirements management of complex systems," in *Complex Systems Design & Management*. Springer, 2010, pp. 99–111.
[6] A. Simpson and J. Stoker, "Will it be safe? an approach to engineering safety requirements," in *Components of System Safety*. Springer, 2002, pp. 140–164.
[7] N. G. Leveson, "An approach to designing safe embedded software," in *Embedded Software*. Springer, 2002, pp. 15–29.
[8] K. Cox, M. Niazi, and J. Verner, "Empirical study of sommerville and sawyer's requirements engineering practices," *IET software*, vol. 3, no. 5, pp. 339–355, 2009.
[9] R. B. Ahmad, M. H. N. M. Nasir, J. Iqbal, and S. M. Zahid, "High perceived-value requirements engineering practices for outsourced software projects." *JSW*, vol. 10, no. 10, pp. 1199–1215, 2015.
[10] B. Solemon, S. Sahibuddin, and A. A. A. Ghani, "Requirements engineering problems in 63 software companies in malaysia," in *Information Technology, 2008. ITSim 2008. International Symposium on*, vol. 4. IEEE, 2008, pp. 1–6.
[11] D. Firesmith, "Engineering safety-related requirements for software-intensive systems," in *Proceedings of the 28th international conference on Software engineering*. ACM, 2006, pp. 1047–1048.
[12] P. Panaroni, G. Sartori, F. Fabbrini, M. Fusani, and G. Lami, "Safety in automotive software: an overview of current practices," in *IEEE International Computer Software and Applications (COMPSAC'08)*. IEEE, 2008, pp. 1053–1058.
[13] P. J. Graydon and C. M. Holloway, "Planning the unplanned experiment: Assessing the efficacy of standards for safety critical software," 2015.
[14] L. E. G. Martins and T. Gorschek, "Requirements engineering for safety-critical systems: A systematic literature review," *Information and Software Technology*, vol. 75, pp. 71–89, 2016.
[15] G. Lami, F. Fabbrini, and M. Fusani, "An extension of ISO/IEC 15504 to address safety processes," in *System Safety, 2011 6th IET International Conference on*. IET, 2011, pp. 1–6.
[16] P. Johannessen, Ö. Halonen, and O. Örsmark, "Functional safety extensions to automotive spice according to ISO 26262," in *International Conference on Software Process Improvement and Capability Determination*. Springer, 2011, pp. 52–63.
[17] M. Johansson and R. Nevalainen, "Additional requirements for process assessment in safety-critical software and systems domain," *Journal of Software: Evolution and Process*, vol. 24, no. 5, pp. 501–510, 2012.
[18] I. O. for Standardization, "14639-1: Health informatics - capacity-based ehealth architecture roadmap - part 1: Overview of national ehealth initiatives," 2012.
[19] J. Pöppelbuß and M. Röglinger, "What makes a useful maturity model? a framework of general design principles for maturity models and its demonstration in business process management," in *ECIS*, 2011.
[20] T. L. Reis, M. A. S. Mathias, and O. J. de Oliveira, "Maturity models: identifying the state-of-the-art and the scientific gaps from a bibliometric study," *Scientometrics*, pp. 1–30, 2016.
[21] A. D. o. D. Defence Materiel Organisation, "+SAFE: A safety extension to CMMI-DEV, version 1.2," Software Engineering Institute, Technical Note CMU, Tech. Rep., 2007.
[22] I. O. for Standardization, "ISO/IEC TS 15504-10:2011 - information technology - process assessment - part 10: Safety extension," 2011.
[23] A. Saeed, R. de Lemos, and T. Anderson, "On the safety analysis of requirements specifications for safety-critical software," {*ISA*} *Transactions*, vol. 34, no. 3, pp. 283 – 295, 1995.
[24] J. Pernstål, R. Feldt, and T. Gorschek, "The lean gap: A review of lean approaches to large-scale software systems development," *Journal of Systems and Software*, vol. 86, no. 11, pp. 2797–2821, 2013.
[25] B. Sechser, "Functional safety-SPICE for professionals?" in *International Conference on Software Process Improvement and Capability Determination*. Springer, 2011, pp. 212–216.

[26] M. Glinz and S. A. Fricker, "On shared understanding in software engineering: an essay," *Computer Science-Research and Development*, vol. 30, no. 3-4, pp. 363–376, 2015.

[27] R. Shakeel, M. Shafi, and K. G. B. Jehan, "Requirement engineering trends in software industry of pakistan."

[28] M. P. E. Heimdahl, "Safety and software intensive systems: Challenges old and new," in *Future of Software Engineering*. IEEE Computer Society, 2007, pp. 137–152.

[29] T. Kontogiannis, M. Leva, and N. Balfe, "Total safety management: Principles, processes and methods," *Safety Science*, 2016.

[30] B. Solemon, S. Sahibuddin, and A. A. A. Ghani, "Requirements engineering problems and practices in software companies: An industrial survey," in *International Conference on Advanced Software Engineering and Its Applications*. Springer, 2009, pp. 70–77.

[31] M. Svahnberg, T. Gorschek, T. T. L. Nguyen, and M. Nguyen, "Uni-REPM: validated and improved," *Requirements Engineering*, vol. 18, no. 1, pp. 85–103, 2013.

[32] P. Sawyer, I. Sommerville, and S. Viller, "Requirements process improvement through the phased introduction of good practice," *Software Process: Improvement and Practice*, vol. 3, no. 1, pp. 19–34, 1997.

[33] T. Gorschek, M. Svahnberg, and K. Tejle, "Introduction and application of a lightweight requirements engineering process," in *Ninth International Workshop on Requirements Engineering: Foundation for Software Quality*, 2003.

[34] T. Gorschek, A. Gomes, A. Pettersson, and R. Torkar, "Introduction of a process maturity model for market-driven product management and requirements engineering," *Journal of software: Evolution and Process*, vol. 24, no. 1, pp. 83–113, 2012.

[35] M. Svahnberg, T. Gorschek, T. T. L. Nguyen, and M. Nguyen, "Uni-REPM: a framework for requirements engineering process assessment," *Requirements Engineering*, vol. 20, no. 1, pp. 91–118, 2015.

[36] F. Marx, F. Wortmann, and J. H. Mayer, "A maturity model for management control systems," *Business & information systems engineering*, vol. 4, no. 4, pp. 193–207, 2012.

[37] P. Fraser, J. Moultrie, and M. Gregory, "The use of maturity models/grids as a tool in assessing product development capability," in *International Engineering Management Conference*, vol. 1. IEEE, 2002, pp. 244–249.

[38] P. Williams, "A practical application of CMM to medical security capability," *Information Management & Computer Security*, vol. 16, no. 1, pp. 58–73, 2008.

[39] R. Wendler, "The maturity of maturity model research: A systematic mapping study," *Information and software technology*, vol. 54, no. 12, pp. 1317–1339, 2012.

[40] J. Becker, R. Knackstedt, and J. Pöppelbuß, "Developing maturity models for IT management," *Business & Information Systems Engineering*, vol. 1, no. 3, pp. 213–222, 2009.

[41] T. De Bruin, R. Freeze, U. Kaulkarni, and M. Rosemann, "Understanding the main phases of developing a maturity assessment model," 2005.

[42] C. G. von Wangenheim, J. C. R. Hauck, A. Zoucas, C. F. Salviano, F. McCaffery, and F. Shull, "Creating software process capability/maturity models," *IEEE software*, vol. 27, no. 4, pp. 92–94, 2010.

[43] C. G. von Wangenheim, J. C. R. Hauck, C. F. Salviano, and A. von Wangenheim, "Systematic literature review of software process capability/maturity models," in *Proceedings of International Conference on Software Process Improvement and Capabity Determination (SPICE), Pisa, Italy*, 2010.

[44] C. P. Team, "CMMI for development, version 1.2," 2006.

[45] A. Dorling, "SPICE: Software process improvement and capability determination," *Software Quality Journal*, vol. 2, no. 4, pp. 209–224, 1993.

[46] D. Hoyle, "ISO 9000: quality systems handbook," 2001.

[47] M. Fleming, "Safety culture maturity model," *Offshore Technology Report-Health and Safety Executive OTH*, 2000.

[48] A. P. Goncalves Filho, J. C. S. Andrade, and M. M. de Oliveira Marinho, "A safety culture maturity model for petrochemical companies in brazil," *Safety science*, vol. 48, no. 5, pp. 615–624, 2010.

[49] R. Wieringa, "Relevance and problem choice in design science," in *International Conference on Design Science Research in Information Systems*. Springer, 2010, pp. 61–76.

[50] J. Vilela, J. Castro, L. E. G. Martins, and T. Gorschek, "Requirements communication in safety-critical systems: A systematic literature review," submitted. For a copy: jffv@cin.ufpe.br.

[51] L. E. G. Martins and T. Gorschek, "Requirements engineering for safety-critical systems: An interview study with industry practitioners," submitted. For a copy: legmartins@unifesp.br.

[52] ——, "Requirements engineering for safety-critical systems: Overview and challenges," *IEEE Software*, 2017.

[53] ——, "Requirements engineering for safety-critical systems: Interview study with industry practitioners," Blekinge Institute of Technology, Tech. Rep., 2016.

[54] I. O. for Standardization, "61508 functional safety of electrical/electronic/programmable electronic safety-related systems," *International electrotechnical commission*, 2011.

[55] ——, "ISO 26262-6: Road vehicles, functional safety part 6: Product development at the software level," *International electrotechnical commission*, 2011.

[56] I. O. for Standardization and I. E. Commission, "ISO/IEC 25010: Systems and software engineering - systems and software quality requirements and evaluation (square) - system and software quality models," *International electrotechnical commission*, 2011.

[57] ——, "ISO/IEC 9126: Software engineering - product quality," *International electrotechnical commission*, 2004.

[58] I. O. for Standardization, "ISO 15998: Earth-moving machinery - machine-control systems (MCS) using electronic components ? performance criteria and tests for functional safety," *International electrotechnical commission*, 2008.

[59] ——, "ISO/TS 15998-2. earth-moving machinery - machine control systems (MCS) using electronic components," *International electrotechnical commission*, 2012.

[60] ——, "ISO 20474-1. earth-moving machinery - safety - part 1: General requirements," *International electrotechnical commission*, 2008.

[61] E. C. for Space Standardization, "ECSS-E-HB-40A: Space engineering - software engineering handbook," *ESA Requirements and Standards Division*, 2013.

[62] ——, "ECSS-E-ST-40C: Space engineering - software," *ESA Requirements and Standards Division*, 2009.

[63] I. O. for Standardization, "Safety of machinery ? safety-related parts of control systems ? part 1: General principles for design," 2015.

[64] ——, "Safety of machinery ? safety-related parts of control systems ? part 2: Validation," 2012.

[65] D. of Defense of United States of America, "MIL-STD-882C: Military standard - system safety program requirements," 1993.

[66] ——, "MIL-STD-882D: Military standard - standard practice for system safety," 2000.

[67] ——, "MIL-STD-882E: Military standard - system safety," 2012.

[68] I. O. for Standardization, "14639-2: Health informatics ? capacity-based ehealth architecture roadmap ? part 2: Architectural components and maturity model," 2014.

[69] S. E. Institute, "CMMI for systems engineering/software engineering (CMMI-SE/SW), version 1.2," 2001.

[70] ——, "A systems engineering capability maturity model (SE-CMMW), version 1.1," 2001.

[71] G. Schedl and W. Winkelbauer, "Practical ways of improving product safety in industry," in *Improvements In system Safety*. Springer, 2008, pp. 177–193.

[72] T.-e. Kim, S. Nazir, and K. I. Øvergård, "A STAMP-based causal analysis of the korean sewol ferry accident," *Safety Science*, vol. 83, pp. 93–101, 2016.

[73] K. Kazaras and K. Kirytopoulos, "Applying stamp in road tunnels hazard analysis," in *6th IET International Conference on System Safety*, 2011.

[74] T. Bosse and N. Mogles, "Comparing modelling approaches in aviation safety," in *Proceedings of the 4th International Air Transport and Operations Symposium (ATOS2013), Toulouse, France*. Citeseer, 2013.

[75] J. Ekberg, U. Ingelsson, H. L¨onn, M. Skoog, and J. S¨oderberg, "Collaborative development of safety-critical automotive systems: Exchange, views and metrics," in *Computer Safety, Reliability, and Security*. Springer, 2014, pp. 55–62.

[76] J. Whitehead, "Collaboration in software engineering: A roadmap," in *2007 Future of Software Engineering*. IEEE Computer Society, 2007, pp. 214–225.

[77] J. G. Hall and A. Silva, "A conceptual model for the analysis of mishaps in human-operated safety-critical systems," *Safety science*, vol. 46, no. 1, pp. 22–37, 2008.

[78] T. Grill and M. Blauhut, *Design patterns applied in a user interface design (uid) process for safety critical environments (sces)*. Springer, 2008.

[79] D. Firesmith, "Engineering safety-related requirements for software-intensive systems," in *Proceedings of the 28th international conference on Software engineering*. ACM, 2006, pp. 1047–1048.

[80] T. Kontogiannis, M. Leva, and N. Balfe, "Total safety management: Principles, processes and methods," *Safety Science*, 2016.

[81] J. Pernstål, R. Feldt, and T. Gorschek, "The lean gap: A review of lean approaches to large-scale software systems development," *Journal of Systems and Software*, vol. 86, no. 11, pp. 2797–2821, 2013.

[82] G. Lami, I. Biscoglio, and F. Falcini, "An empirical study on software testing practices in automotive," in *International Conference on Software Process Improvement and Capability Determination*. Springer, 2016, pp. 301–315.
[83] R. Likert, "A technique for the measurement of attitudes." *Archives of psychology*, 1932.
[84] K. E. Wiegers, "Software requirements: Practical techniques for gathering and managing requirement through the product development cycle," *Microsoft Corporation*, 2003.
[85] T. Gorschek, P. Garre, S. Larsson, and C. Wohlin, "A model for technology transfer in practice," *IEEE software*, vol. 23, no. 6, pp. 88–95, 2006.
[86] T. Hall, S. Beecham, and A. Rainer, "Requirements problems in twelve software companies: an empirical analysis," *IEE Proceedings-Software*, vol. 149, no. 5, pp. 153–160, 2002.
[87] N. Juristo, A. M. Moreno, and A. Silva, "Is the european industry moving toward solving requirements engineering problems?" *IEEE software*, vol. 19, no. 6, pp. 70–77, 2002.
[88] E. Kamsties, K. Hörmann, and M. Schlich, "Requirements engineering in small and medium enterprises," *Requirements engineering*, vol. 3, no. 2, pp. 84–90, 1998.
[89] I. Sommerville, "Software engineering. international computer science series," *ed: Addison Wesley*, 2004.
[90] D. Leffingwell, "Calculating your return on investment from more effective requirements management," *American Programmer*, vol. 10, no. 4, pp. 13–16, 1997.
[91] M. Lubars, C. Potts, and C. Richter, "A review of the state of the practice in requirements modeling," in *Proceedings of IEEE International Symposium on Requirements Engineering*. IEEE, 1993, pp. 2–14.
[92] U. Nikula, J. Sajaniemi, and H. Kälviäinen, *A State-of-the-practice Survey on Requirements Engineering in Small-and Medium-sized Enterprises*. Lappeenranta University of Technology Lappeenranta, Finland, 2000.
[93] G. Kotonya and I. Sommerville, *Requirements engineering: processes and techniques*. Wiley Publishing, 1998.
[94] M. Nguyen, "Empirical evaluation of a universal requirements engineering process maturity model," Master's thesis, Blekinge Institute of Technology, 2010.
[95] G. Santos, M. Kalinowski, A. R. Rocha, G. H. Travassos, K. C. Weber, and J. A. Antonioni, "MPS.BR program and MPS model: main results, benefits and beneficiaries of software process improvement in brazil," in *Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the*. IEEE, 2012, pp. 137–142.
[96] A. M. Davis and D. Zowghi, "Good requirements practices are neither necessary nor sufficient," *Requirements Engineering*, vol. 11, no. 1, pp. 1–3, 2006.
[97] I. Sommerville, *Software engineering*. New York: Addison-Wesley, 2011.
[98] G. Lami, F. Fabbrini, and M. Fusani, "ISO/IEC 15504-10: motivations for another safety standard," *Computer Safety, Reliability, and Security*, pp. 284–295, 2011.
[99] D. C. Pigosso, H. Rozenfeld, and T. C. McAloone, "Ecodesign maturity model: a management framework to support ecodesign implementation into manufacturing companies," *Journal of Cleaner Production*, vol. 59, pp. 160–173, 2013.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSE.2018.2846576, IEEE Transactions on Software Engineering

Jéssyka Vilela is a professor of embedded software engineering at Universidade Federal do Ceará (UFC). She received a B.S. degree in Computer Engineering from Universidade Federal do Vale do São Francisco (UNIVASF), Brazil in 2012. She completed her Master in Computer Science at the Universidade Federal de Pernambuco (UFPE), Brazil in 2015. Jéssyka is also currently a Ph.D student at UFPE and her research interests include Software Engineering, Requirements Engineering, Safety-Critical Systems, Embedded Systems, Context-Sensitive Systems, and Software Architecture. Contact her at jffv@cin.ufpe.br.

Jaelson Castro is a Full Professor at Universidade Federal de Pernambuco, Brazil, where he leads the Requirements Engineering Laboratory (LER). He holds a PhD in Computing from Imperial College, London, UK. His research interests include requirements engineering, adaptive systems, model-driven development and robotics. He serves on the editorial board of the Requirements Engineering Journal and the Journal of Software Engineering Research and Development and acted as Editor-in-Chief of the Journal of the Brazilian Computer Society - JBCS. Contact him at jbc@cin.ufpe.br

Luiz Martins is a professor of software engineering at Federal University of São Paulo (UNIFESP). His research interests include requirements engineering, embedded systems, safety-critical systems and model-driven software development. He has published more than 30 papers in these areas. Martins has a PhD in Electrical Engineering from State University of Campinas (UNICAMP). Contact him at legmartins@unifesp.br.

Tony Gorschek is a professor of software engineering at Blekinge Institute of Technology, Sweden. He has 10y experience working with SW intensive product development in domains ranging from automotive to telecom, working as CTO, chief architect and developer. His research interests include requirements engineering, technology and product management, lean product development, quality assurance, and innovation. Gorschek has a PhD in software engineering from BTH. He's a member of the IEEE and the ACM. Contact him at tony.gorschek@bth.se or visit www.gorschek.com.