1

A Framework to Identify Companies Gaps When Introducing New Standards for Safety-Critical Software

Andrea Ceccarelli[1,2] and Nuno Silva[3]

[1]Department of Mathematics and Informatics, University of Florence, Florence, Italy
[2]CINI-Consorzio Interuniversitario Nazionale per l'Informatica-University of Florence, Florence, Italy
[3]CRITICAL Software S.A., Coimbra, Portugal

1.1 Introduction

Companies working in safety-critical domains as the avionics or space have mandatorily to comply with standards, regulating the lifecycle of the system development, the techniques to be adopted and requirements to be fulfilled in different lifecycle phases. Consequently, a company that develops systems or products in compliance with a standard need skill to use the recommended techniques, often with the support of tools developed within the company or from third parties.

The variety of Information and Communications Technology (ICT) world and applicable domains nowadays imply that several standards for safety-critical systems exist, applied mandatorily and regulating the development and operation of critical systems. As examples, the DO-178B/C [1, 2], DO-254 [3] are the mandatory international standards for the avionics domain; the CENELEC EN 50126 [4], 50128 [5] and 50129 [6] are the mandatory standards for the European railway domain; the ECSS [7] is the set of standards for the space domain in Europe.

When changing domain, a company needs to apply different standards and can encounter several connected issues, such as different: (i) definitions; (ii) level of expectations; (iii) level of details of the required tasks;

(iv) maturity level of processes, techniques, tools, customers, etc.; (v) requirements for tool qualification.

A company wanting to adopt a standard, e.g., as the consequence to the decision to enter a new market, must necessarily (i) gain the skills, techniques and tools necessary to appropriately operate in compliance with the standard, (ii) have a different mindset, and (iii) acquire the necessary expertise. The question that is naturally raised is related to the effort, both in time and cost, of introduction of a standard in a company. Such an effort can be considerable, if the company never worked with similar standards or domains.

1.1.1 Contribution

We present *an easy-to-use framework and a supporting methodology to perform a rapid gap analysis on the usage of standards for safety-critical software*, being them new ones to be introduced or already applied. In other words, the framework can be applied to reason in terms of "changing standard" or in terms of "introducing a new standard". The ultimate objective is to discover with limited effort how far a company is from acquiring a level of knowledge sufficient to apply a specific standard. Our approach is based on the concept of rating the knowledge available: it starts from an understanding of the expertise of a company, and it rates the improvements, in terms of training, needed to reach an adequate level of confidence with the techniques and processes required in the standard. Our approach can be applied to a whole standard, a part of it, or to individual techniques and tools. Thus, our framework offers the possibility to depict the status of the knowledge available in the company, which may offer valuable insights on the areas that are mostly covered, and where potential improvements are possible. The approach can indicate the introduction time, which estimates the overall training time required to introduce a new standard.

In the case study, the framework and the supporting methodology are applied to investigate the verification and validation phases of the DO-178B standard in the company *CRITICAL Software S.A*.

We note that our framework cannot be dissociated from the personnel operating in the company: in fact, the personnel are actually holding the background knowledge and are in charge of acquiring new knowledge. Consequently, the identification of the personnel in the company and their role, together with an investigation of their skills, is part of our approach and connected to the outcome of the analysis.

A relevant note is that we specifically target *software companies*, *prescriptive standards for software*, and the *safety-critical domains*. Although

the framework may also be applicable to other kinds of companies, standards (e.g., goal-based standards opposed to prescriptive ones [8]), and domains, we explicitly remark that our investigations, use case and claims of validity are exclusively related to the above targets. A preliminary version of the framework and methodology appeared in [9].

The rest of this chapter is organized as follows. Section 1.2 presents the state of the art. Section 1.3 illustrates the framework and the methodology. Section 1.4 presents the structure of the dataset used, and how to populate it. Section 1.5 presents the metrics for the qualitative and quantitative evaluation of gaps. Section 1.6 presents the case study, Section 1.7 discusses relevant arguments to exercise the framework, and Section 1.8 concludes the Chapter.

1.2 State of the Art on Gap Analysis in the ICT World

Gap analysis is a renowned concept that finds application in several fields since many years; significant examples are in the fields of civil engineering [10], biology [11], economics [12] and ICT [13–18].

In ICT, gap analysis is usually defined as the study of the differences between two information systems or applications, often for the purpose of determining how to get from one state to a new state. A gap can be presented as the space between where we are and where we want to be; gap analysis is undertaken as a mean of bridging that space. We report on most relevant examples of gap analysis for safety-critical systems.

Gap analysis is part of the Software Process Improvement and Capability Determination (SPICE, now an ISO/IEC standards set [14]) to afford the process capability level evaluations of suppliers. SPICE can result useful to select the cheapest supplier amongst those with adequate qualification, or to identify gaps between the current capability of the supplier and the level required by a potential customer. Similarly, the Automotive SPICE (ASPICE, [195]) starts from SPICE but is specific to the automotive industry. Furthermore, the Capability Maturity Model Integration (CMMI, [13]) includes the Standard CMMI Appraisal Method for Process Improvement (SCAMPI, [13]) that is aimed to appraise organizations capability maturity; the SCAMPI approach can result in a capability level profile, or also in benchmarking against other organizations. However, evaluating performance lies out of its scope [20]. CMMI compliance is not a guarantee of good performance *per se*, i.e., there is high variance in performance results within a maturity level [20]. According to [21, 22], in general, these structured processes are widely applicable for large organizations, while their suitability is more arguable for smaller ones. For both large and small organizations, main concerns are the often elevated

costs, the highly complex recommendations, and the improvement projects which involve a large investment in terms of money, time, resources and long time to benefit.

There are several other examples of gap analysis in the ICT. The Integration DEFinition (IDEF, [15]) is a group of methods used to create a model of a system, analyze the model, create a model of a desired version of the system, and aid in the transition from one to the other. [16] defines an index for measuring and analyzing the divide among countries in the area of ICT infrastructure and access [17] develops a Skills Gap Analysis study to respond to immediate inquiries for information on the needs for ICT skills covering the local, regional, and global markets [18] explores the determinants of cross-country disparities in personalcomputer and Internet penetration, relating technology penetration rates with income, human capital, the youth dependency ratio, telephone density, legal quality, and banking sector development.

Other related approaches can be identified in methods for evaluating the cost of software development projects (e.g., COCOMO [23]), as well as system engineering costs (e.g., COSYSMO [24]). Additionally, there have been efforts in building frameworks to guide and support the design, assessment and certification process, for example [25, 26].

Summarizing, overall a vast literature exists on gap analysis, introduction time, and compliance to safety-critical standards. To the authors' knowledge, till today there are no publicly-available gap analysis for software safety standards that are easy-to-use, easy-to-maintain, and that allows understanding, with limited investment, the effort required to become confident with a standard. Companies can benefit from our solution to evaluate their expertise with a standard, measure how difficult it would be to introduce it, and define an appropriate plan for such a standard.

1.3 Overview of the Framework and Methodology

The framework and the related methodology are presented in this Section. They can be realized and executed with the support of a database and tools for drafting questionnaires and data analysis tools. In fact, the whole methodology was implemented and exercised using as supporting instruments a MySQL database to store data, MySQL Workbench to ease database management, the Google Docs suite to make questionnaire and reports, and a few Java classes for data extraction and elaboration, and to implement the decision tree of Section 1.5.

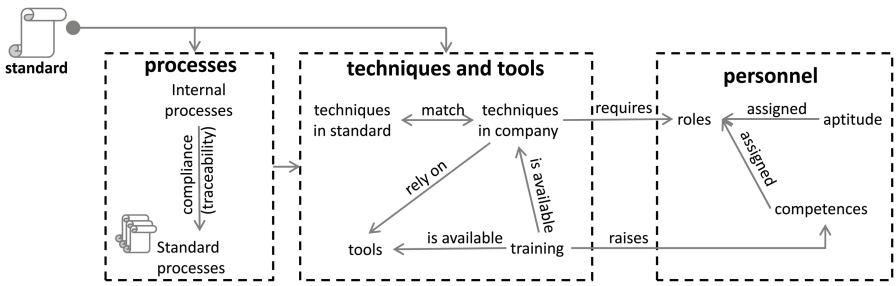

Figure 1.1 Overall view of the gap analysis framework.

In the following chapter, to include examples and to guide our case study, we refer as background knowledge to [27] that classifies the main items, techniques, and processes of aerospace software standards.

1.3.1 The Framework

We present the overall framework with the support of Figure 1.1. It is structured in three main blocks: *Processes*, *Techniques, and Tools*, and *Personnel*. The input to the first two blocks is the *standard* under analysis.

1.3.1.1 Processes

This block is devoted to the identification and matching of the processes. It contains *internal processes* and *standard processes.* Internal processes are defined and applied in a company e.g., internal quality management systems, or internal processes that are required for having certifications like ISO 9001 [28] or CMMI. *Standard processes* are instead the processes or requirements defined in standards; examples at a macro level are design, development, verification, validation, or integration processes.

For each standard, a corresponding traceability matrix must be created and populated; it checks that *internal processes* are compliant to *standard processes*. One or more internal processes should be matched to each process of each individual standard. If the matching is not complete, there may be the necessity to review internal processes; otherwise the applicability of the standard may be compromised.

Although solutions to automate these checks exist [29, 30], we believe that a visual inspection of the standard is sufficient to identify major inconsistencies. This claim is supported by the typically structured descriptions of internal processes and standard processes.

The identification and matching of such processes are inputs to the block Techniques and Tools.

1.3.1.2 Techniques and tools

Both standard processes and internal processes typically list recommended or mandatory techniques.

A whole list of techniques in the standard (*techniques in standard*) and techniques available in the company (*techniques in company*) is required. The list of the techniques in standard needs to be compiled for each standard; the list of techniques in company needs to be compiled only once, and updated when a new technique is learnt.

A traceability matrix can match techniques in company and techniques in standard, to identify the correspondence between the two or possible mismatches. For example, a technique discussed in a standard that has no correspondence among the techniques available in the company know-how. One or more *techniques in company* may be matched to each *technique in standard*. Techniques in standard and techniques in company are also matched to, respectively, standard processes and internal process.

Tools are connected to the techniques in the company, because they can support their execution (occasionally tools can support the whole process [25, 26], although this possibility is not represented in Figure 1.1). Similarly, *training* materials (e.g., slides from courses or tutorials), whenever available, are enlisted and mapped to the company tools and techniques. Noteworthy, techniques or tools not explicitly mentioned in internal processes may be available in the company and useful to support the execution of such internal processes: in this case, it is required to add such techniques or tools and create the appropriate connections to the internal processes.

It is fundamental to understand the confidence in using a technique or a tool; an option is to acquire this information through a questionnaire, as we will discuss in later sections of this chapter. Obviously, this has not to be done on individual basis to rate the single worker, but as a collective exercise between expert workers.

1.3.1.3 Personnel

The personnel are actually holding the background knowledge of the company and are in charge of acquiring new knowledge. The block *personnel* relate the company's personnel to the know-how available on the listed techniques and tools. The block contains information on the personnel as the available *roles*, the desired *aptitude skills* for each specific role, and the required *competences*. Roles are matched directly to the techniques, while competences are matched to training. *Aptitude skills* [31] are instead soft skills as behavioral skills; which have an ancillary role in the framework

but are included to present a complete characterization of personnel. More information on the roles and skills are in Section 1.4.

1.3.2 The Methodology to Exercise the Framework

The overall methodology resulting from the execution of the framework is hereby presented. The steps are the same for gap analysis of standards already in use and for the introduction of a new standard. For simplicity of the discussion, we refer here only to the last case. We assume that the standards $S_1, \ldots, S_{n-1}$ are already part of the framework, and that data on internal processes, techniques in the companies and personnel are already available. This can be done iterating the below steps for the standards $S_1 \ldots S_{n-1}$, until the dataset is up-to-date.

When a new standard S_n is introduced, the approach is the following.

Step 1. The list of standards is updated with S_n, and the corresponding traceability matrix of S_n w.r.t. internal processes is created. Table 1.1 presents a sample extract of such traceability matrix.

Step 2. The list of *techniques in standards* is updated with techniques that are mentioned in S_n; consequently, the match with *techniques in company* is updated. For example, in Table 1.2, the techniques "reviews, inspections, analysis" from the list in [27] are matched to several company techniques, as reviews, inspections, HW/SW interaction analysis (HSIA), traceability analysis. If in S_n there is a technique with no matches amongst the list of techniques in company, it is sufficient to add the same exact name to such list. As a result, a very low rating on the maturity in using such technique will be assigned in Step 4; this will be further discussed also in Section 1.4

Table 1.1 A sample extract of the traceability matrix on processes

Standard Processes (Requirements) from DO-178B	Internal Processes
SW high-level requirements comply with system requirements	Verification Process
SW high-level requirements comply with system requirements	Requirements Analysis
High-level requirements are accurate and consistent	Requirements Analysis

Table 1.2 A sample extract of the traceability matrix on techniques

Techniques in Standard	Techniques in Company
Reviews, inspections, analysis	Reviews
	Inspections
	HW/SW interaction analysis (HSIA)
	Traceability
	Static analysis

and Section 1.5. Ultimately, tools are listed and matched to the techniques in company.

Step 3. The data acquisition process gathers information on the confidence in using each technique and tool.

Step 4. Data is analyzed, and gap analysis and learning time are computed.

1.4 Dataset Structure and Population

1.4.1 Dataset Structure

With the support of the Enhanced Entity–Relationship (EER) diagram in Figure 1.2, we comment on the most relevant elements of the dataset that are

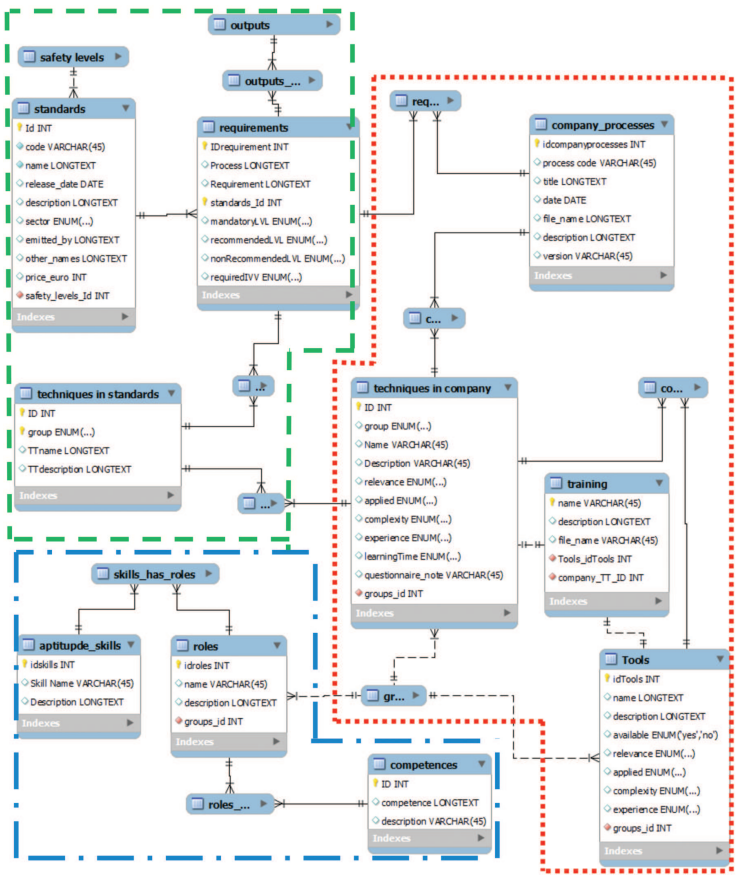

Figure 1.2 EER structure of the database.

required to exercise the framework. The diagram is organized in three areas: the first one (dashed line) contains information on the standards, the second one (dotted line) discusses internal processes and the third one (dash-dotted line) is dedicated to the definition and characterization of the personnel.

We start discussing the first area (dashed line). Table *standards* enlist the standards in use in the company including general information, for example release date, involved industrial domain, and emitting agency. Additional tables can be linked to table *standard* to annotate concepts that differs from a standard to another. As an example, the EER diagram includes the table *safety levels*, which describes the different notion of safety levels across standards. In fact, for example, safety levels are called "Software Levels" and organized in five levels in the DO-178B/C, while they are called "Automotive Software Integrity Levels" (ASILs) and organized in four levels in the ISO 26262. Other examples on safety levels can be found in [27]. Although these annotations are not deemed fundamental for the successful execution of the framework, they can simplify the execution of Step 1 and Step 2 of Section 1.3.

Table *requirements* enlist the requirements, often expressed in terms of steps of a process, described in each standard. Requirements usually suggest specific techniques: table *techniques in standards* enlist the techniques named in each standard. The *table techniques in standards* can specify if a technique is a replacement or alterative to others that are mentioned in the standard. This is useful for the mapping with the second area of Figure 1.2 (dotted line), to favor the matching of techniques in standards with those applied in a company. It is important to report recommendation level of each technique for the considered standard.

The second area includes the table *company processes*, which describes the processes available in the company. Usually, these are described in the internal documentation of a company. Table *techniques in company* enlist the techniques available. Again, such list can be extracted from the internal documentation. To perform the gap analysis, it is required to score the relevance of the technique in the daily work, its frequency of use, the complexity from the point of view of the personnel, the experience of the team in using such technique, the *learning time* (learning time indicates how much training time and hands-on-the-job time is required to gather confidence in applying a specific technique). Table *tools* contain the list of tools available in the company. For the tools table, it is required to evaluate the same attributes as above: relevance, frequency of use, complexity, team experience, learning time of the tool. Section 1.4.2 discusses how to collect such values. Finally, the table *training* enlists the training material available in the company.

The third area (dash-dotted line) is devoted to the identification of personnel. We propose the following minimum set of tables to describe the personnel, although our approach is open to improvements or adjustments in case companies offer different or enhanced characterizations of personnel.

Table *roles* enlist the different roles. Roles are related to the techniques and tools, because it is expected that people having different roles are able to apply different techniques and tools, or take responsibility over different processes. Regarding table aptitude skills, we propose from [31]: (i) behavioral skills e.g., personal integrity, interpersonal skills; (ii) underpinning knowledge i.e., knowledge on the system, required to successfully applying a technique; (iii) underpinning understanding that is general knowledge on the area of work; (iv) statutory and legislation knowledge. Table *competences*, instead, list the required competences as the number of years of experience, or the expertise in a specific topic or domain. Intuitively, table competences and aptitude skills are connected to table roles.

Relations between tables allow connecting and extracting the relevant information from the dataset. For example, the dataset can be used to verify the matching between the standards requirements and company processes. The dataset is also able to differentiate techniques that are similar but used in a different way from domain to domain; the relation of the technique to the corresponding standard is in this case fundamental.

It should be noted that terms reported in the dataset may be very general and several techniques can be matched e.g., requirements-based testing may encompass a large part of the testing activities that are performed on a system or component. The implication is that querying the dataset, different techniques applied in a company can be matched to the same technique in a standard. However this does not alter the methodology, because the different techniques available in the company are first evaluated individually, and then summarizing results are drafted, as explained in Section 1.5.

1.4.2 Population of the Dataset

We discuss hereafter how to collect the main data to populate the dataset. Some data and especially those in the first area are acquired from the documentation typically available in a company.

Regarding the second area, it is required to acquire information on relevance, frequency of use, complexity, experience, and learning time of techniques and tools. While different approaches may exist, in this chapter

we propose a questionnaire that can be distributed between expert personnel to acquire anonymous data.

In this chapter, we propose the following entries and scores to rate techniques and tools applied in the company:

- *Relevance*: high relevance = 4, medium relevance = 3, limited relevance = 2;
- *Frequency of use*: often = 4, rarely = 3, and never = 2;
- *Complexity*: complex = 4, affordable = 3, and easy = 2;
- *Experience*: high experience = 4, medium experience = 3, low or no experience = 2;
- *Learning time* (the time requested by a low-experienced worker to become able to apply a technique or tool with only periodic supervision): less than 1 month = 0.5, ∼1 month = 1, ∼2 months = 2, ∼3 months = 3, and more than 3 months = 4.

The possibility to select the option "*unknown*" is offered, meaning that the person was unable to decide on a rating. This option should be selected when the personnel feels that he is not able to comment on the technique or tool despite being an expert in the specific area. Also, the questionnaire is supposed to be filled only by personnel expert on safety-critical processes, so that they can adequately judge on the techniques and tools, even when they had limited opportunities to get confident with them. Ultimately, note that a questionnaire for techniques in standards is not necessary, because at least one corresponding technique in company is matched to each technique in a standard (see also Step 2 in Section 1.3).

Once all questionnaires are filled, for each technique and tool we select the following values to be computed and added in the dataset: *average, standard deviation*, *mode*, and the *number of unknowns* (number of answers in which the "unknown" option was selected). The mode can be selected instead of the average if the number of questionnaires is small or the results do not lead to a normal distribution. This may result useful in boundary cases, for example when a small subset of the personnel is very skilled on a tool, while the others do not know how to use it.

With respect to the third area (Figure 1.3) proposes a classification of the main personnel roles requested in critical software standards that can be used as reference to populate table *roles* in the dataset. Since our experience is from the aerospace, Figure 1.3 is specifically drafted having aerospace software standards in mind. The following blocks are here considered external

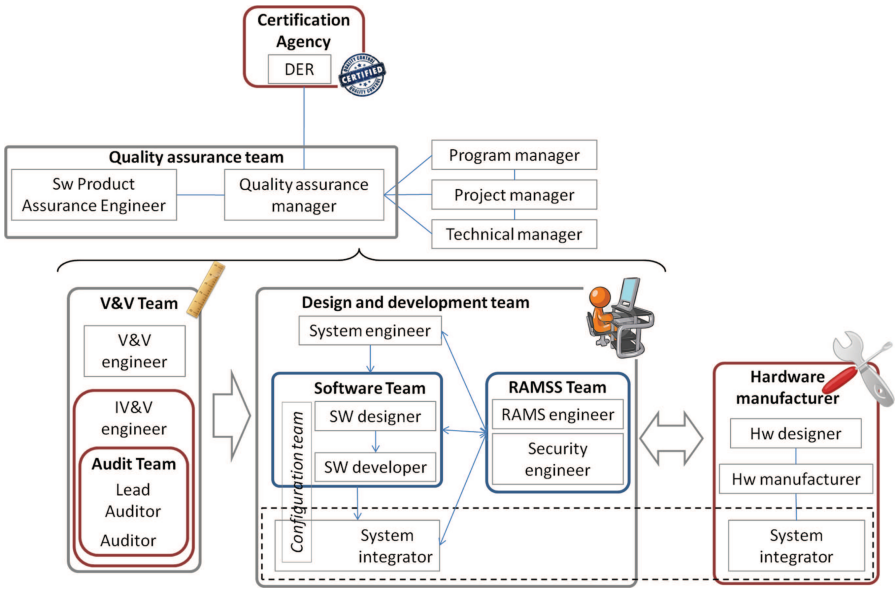

Figure 1.3 Example of roles organization in a critical software company.

to the company: Certification Agency, where the Designated Engineering Representatives, or DER, is located, Hardware Manufacturer, Independent Verification & Validation (V&V) engineer and Audit Team. This is common although it is mandatory only for the Certification Agency. System integrators are connected, because they need to interact closely for hardware–software integration.

The V&V Team and the Independent V&V Team include Test Managers and Test Engineers. The Auditor and the Lead Auditor should be included when addressing services for Independent V&V. The Design and Development Team should include also the Configuration Team, but we merge this role with Integrators, Software Designers and Software Developers. The Quality Assurance Team is in a separate group, which includes Software Product Assurance Engineers. These roles can have different aggregations on other organizations.

To verify the effectiveness of the roles subdivision, we examined the involvement of each role in the most relevant aerospace standards. We identified the personnel roles involved in the different parts of each standard, with sub-section granularity. In other words, we assigned one or more roles

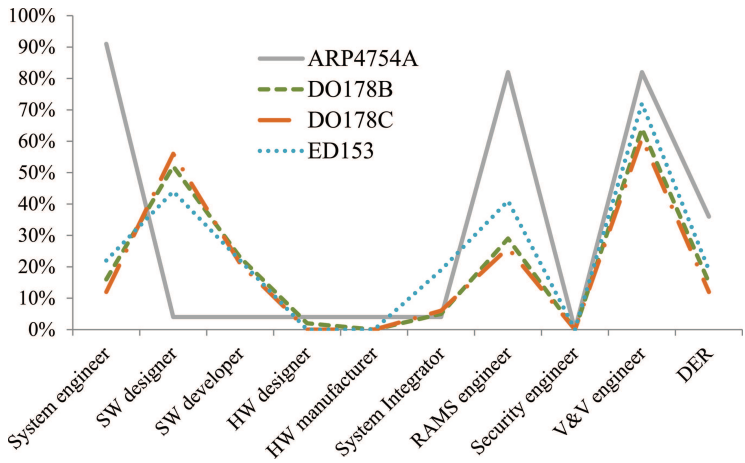

Figure 1.4 Involvement of the different roles in avionics standards.

to the requirements set contained in a subsection. We excluded introductory sections, acronyms, glossary, references sections, and Annexes.

We start our analysis from the avionics standards DO-178B, DO-178C, ED-153 [32] and ARP4754A [33]. The ED-153 applies to software that forms part of an Air Navigation System, and ARP4754A is intended for development of civil aircraft and systems, with emphasis on safety aspects. Results are depicted in Figure 1.4, showing the percentage computed approximating to the nearest integer. V&V and Independent V&V engineers are considered together due to their similar responsibilities. The managerial roles are omitted for readability as they have implicit involvement in every part of the standards. We can note that, overall, the various standards present similar percentage of involvement for the various roles. Especially, the surveyed standards have a similar percentage of V&V engineers, which ranges from 61 to 82%, and of DER. The three standards DO-178B, DO-178C and ED-153 are exclusively related to software and highly correspond for the involvement of software designers, software developers, and hardware-related roles. Instead the ARP4754A is a system-level standard and considers mainly system engineers, RAMS engineers, V&V engineers and DER, while software and hardware designers and developers have a marginal role. Security engineers are little or not considered in these standards.

We performed a similar analysis on most relevant space standards and especially software-related ones. The standards analyzed are the Galileo

software standard GSWS [34], the EUROCAE Guidelines for ANS Software Safety Assurance ED-153 [32], and ECSS standards that we deemed most relevant for safety critical software design and V&V [35–38] and product assurance [39–41]. All standards showed a similar behavior, with the exceptions of GSWS considering also System integrators, and of ED-153 giving low relevance to security engineers. [36] targets mostly system engineers, and to a lesser extent RAMS Engineers and V&V Engineers. [37] and [38] instead are almost exclusively devoted to V&V Engineers. [39] and [40] are intended for RAMS Engineers and V&V Engineers.

1.5 Metrics for Gap Analysis

Once the dataset is populated, qualitative, and quantitative approaches can support the identification of the gaps and the estimation of the introduction time.

1.5.1 Qualitative Indications

We propose a qualitative analysis for the rapid identification of potential weaknesses and get an overall grasp on the results achieved. Several approaches can be identified; we propose in this chapter an intuitive one, based on a simple binary tree that can be easily built for each technique and tool.

The first four levels of the tree correspond to the attributes *relevance*, *experience*, *frequency of use*, *complexity*. The fifth level is a comment in natural language. Starting from the root, at each node, the left or right branch is selected if the score assigned to the attribute is below a threshold or not. The leaves of the tree include conclusive judgments on the technique or tool under exam.

As example, we show in Table 1.3 the binary tree that we defined for our case study. Thresholds are set as follows:

- *relevance* of the technique (for the target standard) = 3
- *experience* = 3
- *frequency of use* = 3
- *complexity* = 3

The final leaf includes a qualitative comment, resulting from the path of the tree, which may suggest the necessity of further investigation.

Obviously, this approach can be easily extended in case of additional attributes or different rating schemes that consider multiple thresholds.

Table 1.3 The binary decision diagram

Relevance	Experience	Frequency of Usage	Complexity	Qualitative Comment
≥3	≥3	≥3	any	Relevant, applied, and experienced.
≥3	≥3	<3	≥3	Relevantand large experience, but not applied.
≥3	≥3	<3	<3	Relevant, simple, large experience, but not applied. R**equires further investigation**.
≥3	<3	≥3	≥3	Relevant and complex. *Applied with little experience*. **Requires further investigation**.
≥3	<3	≥3	<3	Relevant and *applied with little experience*. **Requires further investigation**.
≥3	<3	<3	Any	*Relevant but not applied*and not experienced.**Requires further investigation**.
<3	≥3	≥3	Any	Little relevance, large experience, and applied.
<3	≥3	<3	Any	Little relevance, and not applied.
<3	<3	≥3	Any	Little relevance, but applied *with limited experience*. **Requires further investigation**.
<3	<3	<3	Any	Little relevant, and not applied.

1.5.2 Quantitative Indication

The data acquired may contain information that is not grasped during the qualitative analysis above. We define the quantities Q_1, Q_2, Q_3, Q_4 to relate relevance, (team) experience, frequency of use (called also *applied* below for simplicity), complexity, and to identify those techniques and tools that may need particular attention. Obviously several other different quantities could be identified and applied, without introducing any limitation to the methodology.

We select Q_1, Q_2, Q_3, Q_4 to seek the appropriate balance between complexity, relevance, frequency of use and team experience. The score 0 represents a balance between the different attributes; the highest it is, the highest is the necessity of further investigating the technique or tool.

Is complexity an issue? Q_1= *complexity*^{2}*-applied* $\times$ *experience*. This quantity raises awareness of misalignment between difficulty and confidence. Q_1 is intended to heavily penalize complex techniques. A small Q_1 means that there is high confidence in the usage of a technique.

Is Experience Adequate? Q_2 = *(relevance + applied)* – (*experience* × 2). The objective of this quantity is to indicate that experience is sufficient w.r.t. the relevance and application of a technique.

Is there an overall balancing? Q_3 = *(relevance* × *complexity*) – (*applied* × *experience*). Q_3 compares the confidence in using a technique, i.e., experience and frequency of application, to the relevance and complexity of the technique. It is a summarizing quantity that relates all attributes used up to now.

Is experience justified? Q_4 = *relevance – experience*. Q_4 indicates the experience of the team w.r.t. the relevance of a technique or tool. Ideally, its target score is 0, meaning for example that a very relevant tool is applied with excellent skill; or on the opposite, that a tool recognized as almost irrelevant is also almost unknown. If Q_4 is a positive score, it indicates that a tool or technique acknowledged as relevant is not known adequately.

The case study in Section 0 reports results of these metrics for the analysis of DO-178B in a software company.

1.5.3 Driving Conclusions

The data and the results of the qualitative and quantitative analysis need to be investigated to finalize conclusions. The optimal is that for each recommended technique in the standard, one or more techniques in company are frequently applied with good experience. However, we note that often the techniques recommended in the standards can have replacement techniques, or only a subset of such techniques is actually necessary: this is further elaborated in Section 1.7.

Checks of paperwork or interviews can be a viable support to verify the gaps resulting from the above analysis. This is especially true in two cases. First, we should consider the case when different techniques can be used as alternatives to meet requirements of the standard. A gap in a technique may actually be irrelevant as far as other substitute techniques are applied with good confidence. Second, it is required to prove that the techniques are actually practiced with the skill level declared by the personnel. It is fundamental to know whether the personnel are really practicing in an effective way as declared, matching the on-paper capability of the organization with the as-practiced capability.

Whenever a gap is identified, the value of the *learning time* estimates the time required to fill the gap. The learning time indicates the effort required to train people on a technique or tool; the overall cost to cover the gap should also include the cost of tools licenses, if needed.

Finally, the *introduction time* of the standard can be estimated from all the learning times from techniques and tools where a gap is identified.

1.6 Case Study and Gap Analysis for DO-178B

The framework and methodology were applied within CRITICAL Software S.A.personnel for what concern the DO-178B standard for avionic systems. To reduce complexity, the analysis of the DO-178B we performed was limited to the sections devoted to verification and validation.

CRITICAL Software is an international information systems and software company, headquartered in Coimbra, Portugal, where our experiment took place. While CRITICAL Software works across several markets, in this work we referred to the aerospace division, which is active since 1998. In fact, it has to be noted that CRITICAL Software has relevant experience with DO-178B, applied successfully in several projects for many years. Consequently, it is evident that the objective of this case study is *not* to identify possible lacks in CRITICAL Software processes or inadequate knowledge about the required techniques, but it is to exercise the framework in a real context and verify its applicability.

Relevant data on techniques in standard were acquired from [27]. Techniques in company and tools were identified from material available at CRITICAL Software and expert involvement: this ranged from short interviews/meetings, to training material, publications, and leaflets, V&V plans for different projects, V&V reports, case studies and specific tools reports. The engineering personnel were also interviewed in order to gather the list of tools they typically use, and that may not necessarily be referred on written reports. In total, 22 Verification & Validation techniques were identified; the validation technique *testing* was further subdivided in 26 testing techniques. The number of tools identified is instead 41.

1.6.1 Matching of DO-178B Techniques and Company's Techniques

Matching between standard's and company's processes was performed by manual inspection of the standard and the company's internal processes.

For each verification and validation technique in the standard, one or more techniques were identified in CRITICAL Software processes, use cases, and V&V plans.

We summarize main results. At least one *technique in company* was assigned to each *technique in standard*. There were more than one in some cases. For example the entry "reviews, inspections, analysis" from the table technique in standard is matched to reviews, inspections, HW/SW interaction analysis (HSIA), traceability, static analysis. Similarly, the requirements-based testing amongst techniques in standard is matched to coding/unit testing, system testing, functional testing and black box testing from techniques in company.

General comments on the examples above are that i) such techniques presents significant overlaps, e.g., between functional and system testing, and ii) terms reported in the standards are often very general and several techniques can fit them e.g., requirements-based testing may encompass a large part of the testing activities that are performed on a system or component.

1.6.2 Acquire Data from Personnel

Questionnaires were filled independently by eight CRITICAL Software workers, operating as V&V, RAMS engineers or having managerial responsibilities, prevalently in the context of verification and validation and certification projects. The engineers had been selected with different experiences and expertise in order to make the questionnaires results more representative of the company level. The data were analyzed, and average, mode, standard deviation, minimum value, maximum value and number of unknowns were computed and added to the database.

1.6.3 Analyze the Data: Techniques

To favor understanding the structure of the results, an extract of the data sheets we compiled is reported in Table 1.4.

For most techniques, the standard deviation was rather limited (below 0.5) showing that despite the limited number of questionnaires, there was a good-to-high convergence of answers. Thus we preferred to use the average rather than the mode in our case study.

- *Complexity*. Less complex techniques were identified in reviews, inspections (e.g., Fagan, or walk-throughs), static analysis, traceability, code analysis, HW/SW interaction, and almost all testing techniques. Instead the most complex techniques were recognized in formal methods and modeling, with an average complexity of 3.8 (we remember from

Table 1.4 An extract of our sheet for data analysis; overall it contains 48 techniques and 41 tools. The whole data set is not reported because of its dimension and non-disclosure agreements

Technique	Relevance	Experience	Frequency of Usage	Complexity	Q_1	Q_2	Q_3	Q_4	Binary Tree	Learn. Time
Reviews	4	3.75	4	2.12	–10.48	0.50	–6.50	0.25	Relevant, applied, and experienced	Less than 1 month
Inspections	4	3.65	4	2.5	–8.25	0.75	–4.50	0.37		
Traceability analysis	4	3.5	3.87	2.25	–8.50	0.88	–4.56	0.5		
Static analysis	3.62	3.62	3.75	2.25	–8.53	0.13	–5.44	0		
⋮										
Integration testing	3.5	3.25	3.71	3.12	–2,31	0.71	–1,13	0.25	Relevant, applied, and experienced	∼1 month
Input-based testing	3.28	3.71	3.57	2.28	–8.04	–0.57	–5.76	–0.43		less than 1 month
Robustness testing	3.62	3	3.25	3.37	1.64	0.88	2.48	0.62		∼1 month

Table 1.3 that the maximum is 4). Overall, the *unknowns* were very limited, with at the highest 3 for formal methods.

- *Knowledge.* Highest scores were assigned to reviews and inspections, Fault Trees, Dependence diagrams, testing. In particular regarding testing, although several kinds of testing are enlisted, a high score was assigned to all of them.
- *Relevance* and *Frequency of use.* The smallest scores for these two quantities were assigned to model checking/formal verification. In fact, these techniques have not been considered very relevant for the company business up to now. Amongst testing, security testing was considered of little relevance and seldom applied. The reason is mostly due to the standards in use, which only sparingly require security testing.

Overall, the execution of the binary tree suggested verifying 6 techniques. The one who raised the most interesting discussion is safety analysis, which resulted relevant and complex but little applied. The reason is that a proper and unified process for safety analysis does not exist, although the companies are constantly applying techniques that are part of the safety analysis. Other two techniques that are worth noting are usability testing and use case testing: they were rated relevant, and the personnel felt expert about them, but they were seldom applied. This scarce usage of usability and use case testing is not directly imputable to the will of the engineers but it is due to the characteristics of their projects. The other three techniques were identified as relevant but not applied and with limited experience; replacement techniques are typically used in such cases.

Quantitative indicators. Q_1 answers the question "*is complexity an issue?*" Q_1 score is 10.50 for formal methods and 10.87 for model checking, resulting in the highest score for Q_1. This is in line with all the above observations. Similarly, and not surprisingly, the lowest scores are assigned to reviews (–10.48) and inspections (below –10 in both cases), confirming that they were considered techniques with low complexity.

Q_2 answers the question "*is experience adequate?*" Q_2 relates experience to relevance and application of a technique. Most of the results are contained within the interval [–1.5; +1.5], i.e., near 0. This means that there is a good balance between the relevance and application of a technique, and the experience in its usage, thus not raising any particular alarm. Few techniques are slightly outside such interval. Although no techniques are significantly exceeding the interval, the worst value is registered by safety analysis; this is justified by the reason explained previously.

Instead regarding Q_3, which answer the question "*is there an overall balancing?*" we noticed that most of the techniques are in the interval [–7; +7]. For techniques outside such interval, relevant differences were identified between the couples [relevance; complexity] and [frequency of use; experience]. Most balanced scores, close to 0, are HW/SW interaction analysis (HSIA) and functional analysis (FFPA), considered in general with average scores around 3 for all attributes.

Most troublesome (high Q_3 score) is obviously when the score is a high positive value, suggesting that there is a bad feeling with a technique acknowledged as relevant and complex. Worst values are assigned to security assessment and safety analysis. Regarding safety analysis, the previous considerations hold. A different reasoning is instead applied for security assessment. Security assessment is (correctly) perceived as relevant, and this is easily motivated with the increasing attention that security is gaining nowadays. Also, security assessment is perceived as complex, because widely-accepted methodologies or techniques for security assessment of software-based systems are still failing to root in several industrial domains. Finally, standards sparingly mention security assessment, and consequently it is rarely applied in a company.

Finally, Q_4 answers the question "*is experience justified?*" It resulted that safety analysis has the highest score, meaning that although acknowledged as relevant, the personnel interviewed expressed some doubts on their team experience. This outcome is strictly connected to the previous considerations on safety analysis. Values of Q_4 significantly below 0 were not identified; meaning that overall there is a good balance between relevance and frequency of application of techniques.

Learning time. The shortest learning time was assigned i) amongst verification techniques, to reviews, inspections, traceability, static analysis, and ii) amongst validation techniques, to coding/unit testing, regression testing, input-based testing, boundary value analysis, smoke testing, ad hoc testing. Longest learning time was assigned to formal methods and model checking.

1.6.4 Analyze the Data: Tools

Tools connected to the above techniques were evaluated, although no specific issues were identified. Some tools were identified as little relevant, not applied, or largely unknown, but this was due to the fact that the tools list included also obsolete tools.

As an example, the quantity Q_4 resulted in almost all the tools as a negative value, with a few exceptions. In all cases, the value was in the interval [–1.1; 0.7]. Note that the best value, which is –1.1, was assigned to a text editor tool: it is reasonable to believe that there is a good experience in using it, although it is not fundamental because it can be easily replaced by other products.

1.6.5 Conclusive Recommendations and Feedbacks

As expected, no issues can be identified from the analysis. In general, the outcomes which suggest smaller confidence are those related to formal methods and model checking, although other replacement techniques are accepted in DO-178B and this does not really constitutes a gap in what concerns the DO-178B application.

It is worth observing that a long learning time (above 3 months) is assigned to these techniques, meaning that it is considered not easy to acquire proficiency with them. However, this is mostly due to the fact that the company has a limited focus in such activities, thus having a limited number of people skilled in the area.

The fact that Formal Methods and Modeling are not (for the particular case study) well ranked has several reasons, and specifically: (i) engineers are not prepared for these techniques from university and prior experience, (ii) they are not yet widely accepted in industry, especially from customers, (iii) they are more complex than others, and (iv) they lack appropriate tools support.

A final remark is about the techniques in standard that are grouped as number 11 in [27], that is, *similarity, service experience, failure statistics*. The corresponding techniques in company were rated poorly, mostly showing the entry "*unknown*" in the questionnaire for all attributes. A later analysis with direct confrontation with personnel concluded that the terms used for such techniques were unclear and confused the personnel involved. In fact, the questionnaire was provided to the personnel but entries not discussed in advance. The clarification allowed to verify the absence of any gap, thus solving all issues on similarity-based approaches for the verification of critical systems, with the only action of correcting techniques names in the dataset.

Finally, it is important to note that the case study was performed in a short time frame and its results might be interesting to plan ahead, estimate and

have the company ready to tackle new domains and new certification challenges. It is relevant to mention that once these results have been presented to the company personnel, CRITICAL Software has taken actions to fill these gaps, and, in the frame of the European project FP7-2012-324334-CECRIS [42], processes, techniques and training material for safety analysis and for security assessments were developed. This outcome shows the direct impact that these types of analysis can have in prioritizing Research & Development within an organization. A more detailed discussion on this aspect, which we rate an important outcome of our work, is in Section 1.7.

1.7 Discussion about the Gap Analysis Framework

1.7.1 An Application to the Moving Process

This work represents a formalization of what is usually done by industries when tackling a new domain of expertise, but not always in a structure way and not always with all the required information to make sound decisions and appropriate plans. The results of this framework help to determine the actual level of knowledge and resources that can be reused instead of doing it in an *ad hoc* and less supported manner.

Discussing the specific *moving process* is not part of the paper but we cannot ignore it. Moving from one existing standard into a standard from another domain involves different factors. For example, the switch between space, avionics, railway or automotive domains involves at least cultural implications, domains specific adaptations, and a large learning process. We provide the basis to support this moving process, by identifying clear gaps, improvements and adaptations, and by providing an estimation of the effort of moving from one domain to the other based on what the company is already applying and the maturity associated to the application of those standards.

For the gap analysis or determination of where a certain company is before entering a specific new domain, it is essential to be able to properly and precisely model the new standards i.e., extracting the requirements, phases, techniques, outputs, etc. This is one of the main tasks of our work and it consists in studying the standards and modeling their contents. Then, it is also extremely important for a company to hold an internal knowledge base about their processes (e.g., in an internal quality management database), techniques (e.g., detailed plans) and tools (e.g., in the form of Software Development Plans and Verification and Validation Plans).

1.7.2 Time and Cost

Gap analysis processes are typically executed sparingly because of the required time, overall complexity, and cost. Consequently, we present an approach that can be executed with little time, effort and cost, provided that personnel with a strong background on safety-critical systems are available. As example, let us consider our case study. Once the framework and the methodology were ready, the whole case study including the population of the dataset was completed in a short time frame. Considering only time-consuming activities, the analysis of the DO-178B standard to fill table *techniques in standard* required 2 days, and the analysis of techniques and tools to fill table *techniques in company* and table *tools* required instead 4 days. It should be noted that these two tables will require only minor updates whenever the framework is exercised on a different standard. Two days were instead necessary to build relations between all tables. The questionnaires were filled in less than two hours each. Drafting conclusions, making interviews and presenting results required four more days.

Our analyses were carried out with a small number of supporting tools: the tools we used in our case study are a database, a spreadsheet tool, a text editor, and Java applications we developed in less than 600 lines of code. These Java applications allowed parsing the questionnaire, interfacing to the dataset, and building the binary tree. The artifacts produced by the framework, including those used in this work, are totally reusable for future analysis; this can be achieved simply maintaining the dataset.

1.7.3 Effectiveness and Reactions

Benefits of recovering a gap are usually acquired only after the introduction of the new standard is completed and the new market penetrated, or when novel services are sold thanks to the new skills acquired. This process is typically long and consequently the return on investment for covering a gap or applying a systematic quality assurance process is typically considered on the medium-long term [21].

However, since the benefits are evident, in all cases *the identification of gaps should be the trigger of recovery plans*. As an example, let us consider our case study. Although overall no problems were identified in the application of DO-178B, our analysis led the Research & Development of CRITICAL Software to focus on the topics of Safety Analysis procedures and Security Assessment techniques. In particular, two main actions were taken, partially supported by the project FP7-2012-324334-CECRIS [42].

First, research on approaches for safety analyses was started. We cite two published works that underline this research direction [43] discuss a measurable approach to fulfill the standard requirements but with an acceptable level of effort and within a reasonable timeframe [44] focus on techniques selection for safety analysis, aiming to provide to industries a ranked list of techniques that avoid specific types of issues.

Second, research on the interplay between safety and security was carried on, studying how security issues may impact safety [45] and walking towards the identification of threat assessment methodologies [46, 47]. A new security assessment process named STECA "Security Threats, Effects and Criticality Analysis" is currently under research, with the objective of making the company more competitive and more prepared to provide related services to the industry.

1.7.4 Replacement Techniques

It is important to consider that the standards, mostly based on a waterfall traditional V model, have requirements usually divided by lifecycle phases. For each of these phases there are proposed or recommended techniques, actions, and analysis, but not all of the listed techniques need to be applied in order to fulfill the standards requirements. We are aware that this situation has an impact on our framework, especially when determining gaps between standards: some gaps might be mitigated by other replacement techniques.

As future work, the most relevant foreseen improvement is to introduce the concept of *minimum set of recommended techniques.* Generally, the standards provide recommendations to several techniques but only a few are really required: several are proposed as alternatives. A company needs to be knowledgeable only with a set of such techniques. For example, formal methods are barely used, but all systems can get certified even without formal methods, and a similar reasoning can be carried out for fault injection. In this framework we are addressing this problem only in Step 4, once that all techniques have been evaluated individually. The future improvements of the framework will include solutions to automatically deal with this problem, introducing groups of techniques in the dataset, and adapting the questionnaires to rate the groups and not only the individual technique.

1.7.5 Different Approaches to Compliance

It should be remarked that compliance with standards may take place in two different ways: by sticking to what is recommended or by following tailoring rules. The framework usage applied in our case study directly fits the first

approach. The second approach is also followed in practice, especially in the case that standards requirements are unclear and open to interpretations. In several cases, certification authorities' engineers or auditors supports this approach, helping the companies to adapt and accomplish the certification evidences.

In these cases, our framework can be successfully applied only after the tailoring rules are translated into requirements and are added to the dataset. Once this operation is completed, the framework can be exercised as usual.

1.7.6 Questionnaire Assessment and Bias

We want to depict the status of the company and the feeling of workers towards specific techniques and standards in general, considering that also most of the time the workers themselves are in charge of training personnel and transfer knowledge. The personnel are expected to have a general, broad knowledge of the techniques that are executed and on their usual relevance. In other words, as far as personnel skilled in certification of safety-critical software is available, our framework will be able to rate techniques event if personnel is not familiar with them.

However, a relevant concern is the risk of a bias in the outputs due to the personnel perception of their expertise and experience. It is intuitive to expect that engineers will report higher scores for the techniques they have experience with and actually use, and that the analysis may underemphasize important techniques that the company is unfamiliar with. This reflects a simplification from an engineering perspective, as we tend to apply only one technique or a simpler tool if it is accepted for the certification or for completing the job.

These considerations require that, when interviewing the personnel, a good assessor, or an expert engineer that deeply studied the considered standard, is present. Otherwise, the process and self-image of competence, for example personnel feeling they are much more skilled than they actually are, may introduce significant bias in the results.

1.8 Conclusions

This chapter proposed an *easy-to-use* framework and a supporting methodology to perform a *rapid* gap analysis on the usage of *standards for safety-critical software*. The methodology can be applied to new standards to be introduced or to standards that are already applied in the company as long

as skilled personnel are available. The ultimate objective is to discover with reduced effort and minimal supporting tools how far a company is from having a sufficient level of knowledge to apply a specific standard. Also, the framework allows estimating the time required to cover the gaps. Our case study was executed in a short time frame, proving evidence of the intuitiveness of our solution. Results have been presented to a larger audience at the company CRITICAL Software SA, where the audience agreed that they reflect the global feeling about strengths and weaknesses, and recovery actions were taken by the Research & Development team.

References

[1] RTCA. (1992). *Software Considerations in Airborne Systems and Equipment Certification* (DO-178B/EUROCAE ED-12B).

[2] RTCA. (2011). *Software Considerations in Airborne Systems and Equipment Certification* (DO-178C/EUROCAE ED-12C).

[3] RTCA. (2000). *Design Assurance Guidance for Airborne Electronic Hardware*. (DO-254/EUROCAE ED-80).

[4] CENELEC. (2006). *Railway applications: The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) Part 1: Basic requirements and generic process* (EN 50126-1/EC: 2006-05).

[5] CENELEC. (2002). *Railway applications: Communications, signalling and processing systems – Software for railway control and protection systems*, EN 50128.

[6] CENELEC. (2004). *Railway applications: Communication, signalling and processing systems – Safety related electronic systems for signalling*, EN 50129.

[7] European Cooperation on Space Standardization (ECSS). (2014). Available at: http://www.ecss.nl/ (last accessed 11 November 2014).

[8] Penny, J., et al. (2001). *The practicalities of goal-based safety regulation." Aspects of Safety Management*. London: Springer, 35–48.

[9] Ceccarelli, A., and Silva, N. (2015). "Analysis of companies gaps in the application of standards for safety-critical software," in *RESA4CI workshop, Computer Safety, Reliability, and Security*. London: Springer International Publishing, 303–313.

[10] Karbhari, V. M., et al. (2003). Durability gap analysis for fiber-reinforced polymer composites in civil infrastructure. *J Compos. Construct*. 7.3, 238–247.

[11] Powell, G. V. N., Barborak, J., and Rodriguez, M. S. (2000). Assessing representativeness of protected natural areas in Costa Rica for conserving biodiversity: a preliminary gap analysis. *Biol. Conserv*. 93.1, 35–41.

[12] Brown, S. W., and Swartz, T. A. (1989). A gap analysis of professional service quality. *J. Market*. 53, 92–98.

[13] CMMI Product Team. (2010). "CMMI for Development". Technical Report, Software Engineering Institute, CMU, Pennsylvania.

[14] ISO/IEC 15504. (2004). *Information technology – Process assessment* 2004.

[15] Hanrahan, R. P. (1995). "The IDEF process modeling methodology," in *Software Technology Support Center*, New York, NY: IEEE 1995.

[16] Hanafizadeh, M. R., Saghaei, A., and Hanafizadeh, P. (2009). An index for cross-country analysis of ICT infrastructure and access. *Telecommun. Policy* 33, 385–405.

[17] El-Gabaly, M., and Majidi, M. (2003). *ICT Penetration and skills gap analysis*. Egypt: US AID's Mission in Egypt.

[18] Chinn, M. D., and Fairlie, R. W. (2010). ICT use in the developing world: an analysis of differences in computer and internet penetration. *Rev. Int. Econ*. 18.1, 153–167.

[19] Verband der Automobilindustrie (VDA). *Automotive SPICE – Process Assessment Model*, 1st edn, 2008.

[20] Margarido, I. L., Faria, J. P., Vidal, R. M., and Vieira, M. (2012). "Towards a framework to evaluate and improve the quality of implementation of CMMI® practices". Product-Focused Software Process Improvement. Berlin: Springer, 361–365.

[21] Pino, F. J., Pardo, C., García, F., and Piattini, M. (2010). Assessment methodology for software process improvement in small organizations. *Inf. Softw. Technol.* 52, 1044–1061.

[22] Mark, S., et al. (2007). An exploratory study of why organizations do not adopt CMMI. J. Syst. Softw. 80.6, 883–895.

[23] Kemerer, C. F. (1987). An empirical validation of software cost estimation models. Commun. *ACM* 30.5, 416–429.

[24] Valerdi, R., Boehm, B., Reifer, D. (2003). "Cosysmo: a constructive systems engineering cost model coming age," in *Proceedings of the 13th Annual International INCOSE Symposium* (pp. 70–82). New York, NY: IEEE.

[25] Ceccarelli, A., Vinerbi, L., Falai, L., and Bondavalli, A. (2011). "RACME: A Framework to Support V&V and Certification," in *5th*

Latin-American Symposium on Dependable Computing (LADC), 116, 125, 25–29.

[26] Rezabal, M. I., Elorza, L. E., Letona, X. E. (2013). "Reuse in Safety Critical Systems: Educational Use Case," in *39th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, 402, 407.

[27] Ceccarelli, A., and Silva, N. (2013). "Qualitative comparison of aerospace standards: An objective approach," in *2013 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, 331, 336. New York, NY: IEEE.

[28] ISO 9001:2008 Quality Management Systems.

[29] Deeptimahanti, D. K., and Sanyal, R. (2011). "Semi-automatic generation of UML models from natural language requirements," in *Proceedings of the 4th India Software Engineering Conference*. New York, NY: ACM.

[30] Kof, L. (2009). "Translation of textual specifications to automata by means of discourse context modelling," in *Requirements Engineering: Foundation for Software Quality*. Berlin: Springer, pp. 197–211.

[31] IET. (2007). *Competence Criteria for Safety-related system practitioners*.

[32] EUROCAE. (2009). *EUROCAE ED-153 – Guidelines for ANS Software Safety Assurance*.

[33] SAE. (2010). *ARP4754A/EUROCAE ED-79 – Guidelines for development of civil aircraft and systems-Revision* A.

[34] Galileo industries. (2004). *GAL-SPE-GLI-SYST-A/0092 – Galileo Software Standard (GSWS)*.

[35] ECSS. (2009). *ECSS-E-ST-40C – Space engineering – Software*.

[36] ECSS. (2009). *ECSS-E-ST-10C – Space engineering – System engineering general requirements*.

[37] ECSS. (2009). *ECSS-E-ST-10-02C: Space engineering – Verification*.

[38] ECSS. (2012). *ECSS-E-ST-10-03C: Space engineering – Testing*.

[39] ECSS. (2009). *ECSS-Q-ST-30C: Space product assurance – Dependability*.

[40] ECSS. (2009). ECSS-Q-ST-40C: Space product assurance – Safety.

[41] ECSS. (2009). ECSS-Q-ST-80C: Space product assurance-Sw product assurance.

[42] CECRIS. (2016). FP7-2012-324334-CECRIS: CErtification of Critical Systems. Available at: http://www.cecris-project.eu/

[43] Silva, N., and Vieira, M. (2013). "Certification of embedded systems: Quantitative analysis and irrefutable evidences," in *2013 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. New York, NY: IEEE.

[44] Silva, N. and Vieira, M. (2014). Towards making safety-critical systems safer: learning from mistakes," in 2014 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), vol., no., 162–167.

[45] Nostro, N., Bondavalli, A., Silva, N. (2014). *Adding Security Concerns to Safety Critical Certification*. ISSRE Workshops, 521–526. New York, NY: IEEE.

[46] Nostro, N., Ceccarelli, A., Bondavalli, A., and Brancati, F. (2014). Insider threat assessment: a model-based methodology. *SIGOPS Oper. Syst. Rev.* 48, 3–12.

[47] Nostro, N., Ceccarelli, A., Bondavalli, A., and Brancati, F. (2013). "A methodology and supporting techniques for the quantitative assessment of insider threats," in *Proceedings of the 2nd International Workshop on Dependability Issues in Cloud Computing – DISCCO '13*, 1–6, Braga (Portugal).