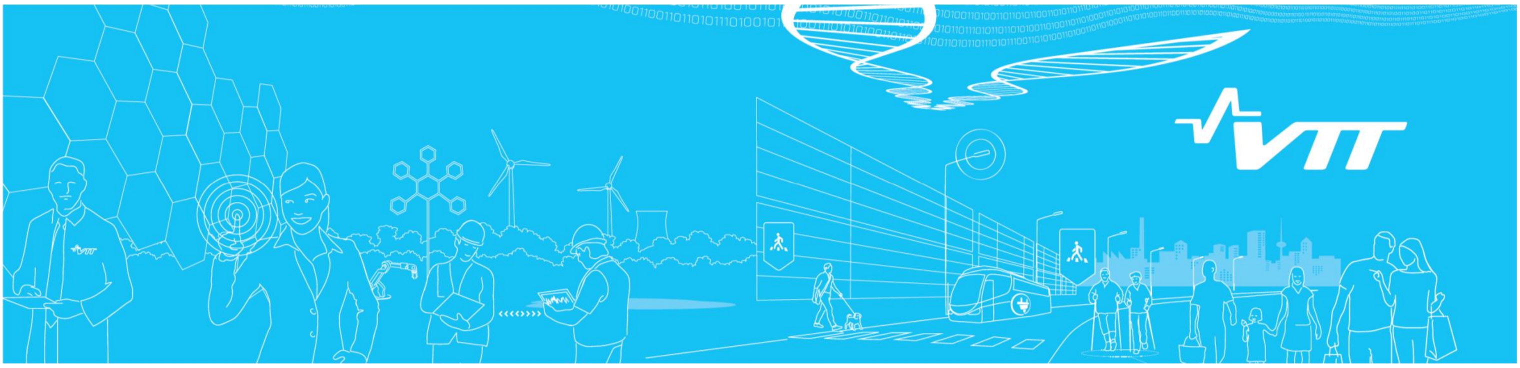

RESEARCH REPORT

VTT-R-03820-14

Comparison of software safety standards IEC 61508-3 and IEC 62138

Authors: Ossi Teikari, VTT
Risto Nevalainen, FiSMA

Confidentiality: Public

Report's title

Comparison of safety standards IEC 61508-3 and IEC 62138

Customer, contact person, address VYR	**Order reference** 4/2014SAF
Project name Coverage and rationality of the software I&C safety assurance	**Project number/Short name** 85361 CORSICA
Author(s) Ossi Teikari, Risto Nevalainen	**Pages** 30/
Keywords standard, comparison, safety, nuclear	**Report identification code** VTT-R-03820-14

Summary

The nuclear domain software safety standard IEC 62138 is currently being updated. In this work we have reviewed and compared two software safety standards, IEC 61508-3 and IEC 62138. The main purpose of the comparison was to aid the renewal process of the IEC 62138 standard by identifying aspects that should be taken into account during the renewal process. The comparison is twofold. Firstly, the standards are compared with each other in terms of general differences and differences in relevant themes in the standards. Secondly, both standards are compared against selected nuclear regulatory requirements; the Finnish YVL E.7 and the common position of seven European nuclear regulators.

As a result of comparing IEC 61508-3 against IEC 62138, the standards were found to have some significant differences in concepts and scope, but also in the main themes that they were designed to cover. Based on the comparison with the regulatory documents, potential areas of deficiencies were identified in both standards.

Confidentiality	Public

Espoo 29.08.2014

Written by	**Reviewed by**	**Accepted by**
Ossi Teikari Research Trainee	Jussi Lahtinen Research Scientist	Riikka Virkkunen Head of Research Area

VTT's contact address

VTT Technical Research Centre of Finland
P.O. Box 1000, FI-02044 VTT, Finland
Phone internat. +358 20 722 4520
Fax +358 20 722 4374

Distribution (customer and VTT)

VTT Kirjaamo, SAFIR2014 Reference group 2

The use of the name of the VTT Technical Research Centre of Finland (VTT) in advertising or publication in part of this report is only permissible with written authorisation from the VTT Technical Research Centre of Finland.

Preface

This report has been prepared under the research project Coverage and rationality of the software I&C safety assurance (CORSICA), which is part of the Finnish Research Programme on Nuclear Power Plant Safety 2011–2014 (SAFIR2014). The research project aims to improve the safety evaluation of I&C systems in nuclear industry by improving consciousness of process assessment and rationality of integrated evaluation methods. This paper presents the results of a safety standard analysis work from year 2014. In this work we have compared two software safety standards, IEC 61508-3 and IEC 62138. The main purpose of this comparison was to aid the renewal process of the IEC 62138 standard, and identify aspects that should be taken into account during the renewal process.

We wish to express our gratitude to the representatives of the organizations involved and all those who have given their valuable input in the meetings and discussions during the project.

Espoo, August 2014

Authors

Contents

1. Introduction

1.1 Background

The software safety standard IEC 62138 is currently being updated. At the time of writing, a CD1 version of IEC 62138 [5] has been prepared. Where the text mentions IEC 62138, it should be assumed that it refers to the CD1 version, unless otherwise specified.

The main intention of this study is to aid the renewal process of IEC 62138. This is done by reviewing the standard and comparing it with other selected documents that are relevant in the domain (IEC 61508-3 [8], Common Position 2013 [12], YVL E.7 [11]). Overall, the purpose of this report is to help the reader in identifying factors that IEC 62138 might be lacking or factors that might need elaboration, i.e. parts that could be added to or modified in the standard.

Firstly, to introduce the reader to the context, an overview of software related safety standards is given in section 1.2. In section 2, the focus is on explaining safety standards that are relevant specifically within instrumentation & control (I&C) systems in the nuclear domain. The comparison itself is twofold. Firstly, IEC 62138 is compared against IEC 61508-3. The structure of this comparison is introduced in section 3 and the results are presented in section 4. The second part of the comparison involves comparing IEC 62138 and IEC 61508-3 with selected regulatory documents: Common Position 2013 and the Finnish YVL E.7. The comparison and its results are presented in section 5. Finally, conclusions of the study are presented in section 6.

1.2 Overview of software-related safety standards

Software is quite a new topic in safety standards. It can be presented in many ways:

- Software safety can be a separate part in a family of safety standards or in some multi-part standard. Examples are IEC 61508 and ISO 26262.

- Software safety can be addressed in an independent safety standard. Examples are IEC 60880 [4], IEC 62318, IEC 62304 and DO-178C.

- Software safety can be explicitly included in a higher level standard at system or hardware level. An example is ISO/IEC 15288 for systems engineering lifecycle. Safety is not expected or mentioned in the standard, but it is quite obvious as the main emphasis is high quality.

- Software safety can be only implicit, in which case there is a need for context-specific interpretation. An example is the ISO9000 family of standards for quality management. Safety is not mentioned in the standard, but it could be included in the context. ISO9001 certificate is required de-facto from suppliers delivering digital safety systems including software for nuclear utilities.

In Figure 1, this multiplicity is illustrated as a dimension of abstraction levels in standards. For practical reasons, only some of the most relevant standards are included in the figure. The other dimension in Figure 1 is the degree of domain specific safety context. It is classified as generic (safety is only implicit or hidden), generic safety (without any domain context), and domain specific safety.

Examples of domains having requirements for software safety are automotive, electro medical industry, space and avionics, railways and nuclear power. Of course, many other domains also have similar concepts for software safety. As a whole, various domains have altogether several hundred standards for safety. Some of them are done in “de-jure” organisations, like ISO or IEC. Even more of them are done in domain specific “de-facto” consortiums and alliances.

Some safety standards may have EN-status, and their status is normally higher in EU member countries. Such standards may be required in EU directives or in national regulatory guides.

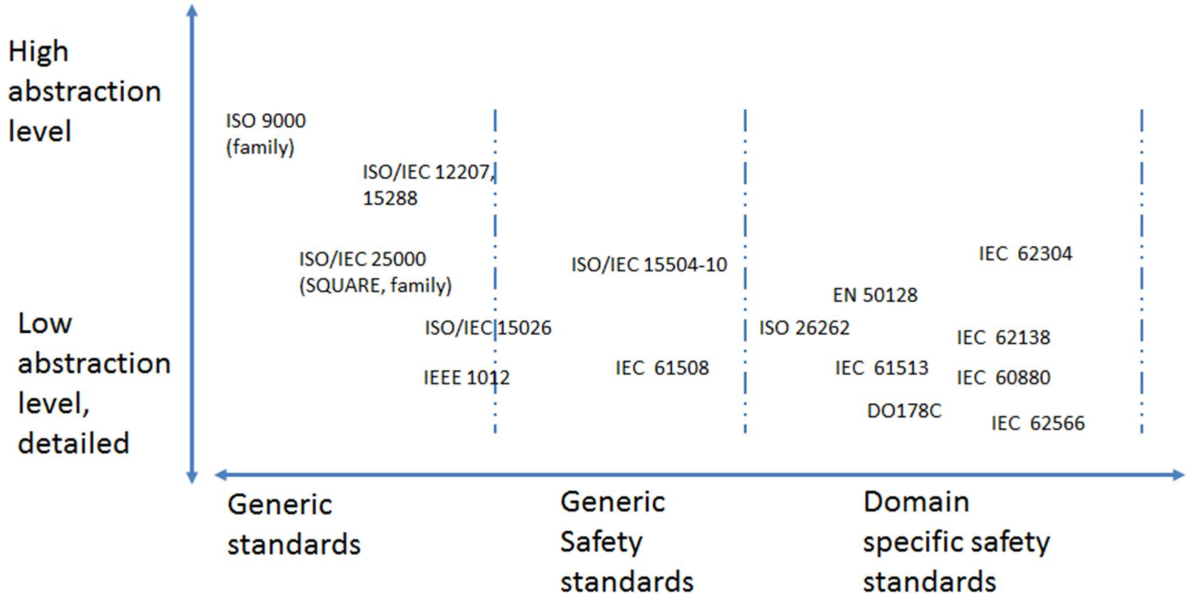

Figure 1. Some selected standards classified by abstraction level and scope

Table 1 below explains shortly the main content of each standard in Figure 1. For readability reasons, full names of the standards are not used. The potential safety content is also explained. It can be only implicit or quite detailed and normative in requirements for software engineering or software safety.

Table 1. Overview of selected standards (see Figure 1)

ID	Main content	Safety content
ISO9000 (family)	Quality management and assurance	Nothing, but can be interpreted for safety
ISO/IEC 12207:2008	Software engineering lifecycle processes	Implicit, but can be applied for safety related software development
ISO/IEC 15288:2008	System engineering lifecycle processes	Implicit, but can be applied for safety related system engineering
ISO/IEC 15026:1998	Assurance case	Implicit, is mainly generalisation of safety case concept
ISO/IEC 25010:2011	Product quality model	Safety is one sub-characteristic to be measured in quality in use
ISO/IEC 27000 (family)	Information security	Implicit, most security principles and methods are valid also for the security and safety combination

IEEE 1012 (December 2011)	V&V for critical systems and software	Implicit, can be applied for safety critical HW and SW verification and validation
IEC 61508:2010 (2nd edition)	Functional safety requirements and lifecycle (systems, hw, sw)	Safety related, defines all SIL levels 1 – 4. Defines methods and safety properties.
IEC 62304:2006	Medical device software – Software life cycle processes	Software safety classes A – C according to risk for health
EN 50128:2011	Railway application – Communications, signalling and processing systems Software for railway control and protection system	Used widely in railway companies and in certification bodies. Defines software safety IL (Integrity Level) 1 – 4.
ISO 26262:2011	Safety lifecycle for systems, hardware and software in automotive industry	Safety specific standard, defines ASIL 1 – 4 (covers SIL 1 – 3). Has 10 parts, covers system, hardware and software.
DO178 C:2011	RTCA/DO178C Software Considerations in Airborne Systems and Equipment Certification.	Software safety specific standard for avionics and space, defines 5 safety levels D (lowest) – A (highest). 178 B was the earlier version and is still in wide use.
IEC61513:2011 [2]	Nuclear power – Systems for Category A functions	Full lifecycle for electronic I&C systems, including software
IEC 60880:2006	Nuclear power – Software for Category A functions	Detailed requirements for safety-critical software in nuclear power plants
IEC 62566:2012	Nuclear power plants – Instrumentation and control important to safety using new technologies	Development of HDL-programmed integrated circuits for systems performing category A functions

2. Standards for I&C systems in the nuclear domain

2.1 Overview

There exist many different standards concerning the safety of nuclear power plants and nuclear power plant I&C systems. *International Electrotechnical Commission* (IEC), the *International Atomic Energy Agency* (IAEA), and *Institute of Electrical and Electronics Engineers* (IEEE) have all published their relevant standards. Additionally, many national regulatory bodies have individual requirements for nuclear power plant systems. This report specifically targets selected IEC I&C software safety standards IEC 61508-3 [8] and IEC 62138, which are introduced more extensively in the following sub sections.

All standards are already somewhat behind leading edge at the moment they are published. The standardisation process takes several years. Necessary changes for the next generation of standards are typically identified 3 – 4 years before final publication, and the whole development cycle for a standard is typically about 10 years.

Standards and regulations can be quite different, due to differences in their purpose and history. Some specific safety factor included in one standard may not be included in another similar standard. Therefore the report also discusses what may be missing or what is only very briefly presented in the selected standards. To fully understand the context, relevant system level safety standards also have to be discussed.

2.2 Nuclear domain IEC standards for I&C systems

IEC has published several standards specifically considering the development of safety-related I&C systems used in nuclear power plants. These standards are introduced in Table 2.

Table 2. IEC standards for nuclear I&C systems

Standard	**Domain**	**Scope**
IEC 61513	Nuclear	General requirements for I&C systems
IEC 61226	Nuclear	Classification of I&C functions
IEC 60987	Nuclear	Hardware requirements for I&C systems
IEC 60880	Nuclear	Software requirements for I&C systems performing category A functions
IEC 62138	Nuclear	Software requirements for I&C systems performing category B and C functions

The uppermost standard, IEC 61513, discusses general requirements for I&C systems in nuclear power plants. IEC 61513 has adopted a presentation format similar to the basic safety publication IEC 61508, with an overall safety life-cycle framework and a system life-cycle framework. IEC 61513 also refers to ISO as well as to IAEA GS-R-3 and IAEA GS-G-3.1 for topics related to quality assurance (QA). [2]

IEC 61226 [1] classifies I&C functions in nuclear power plants into categories A, B and C depending on the importance that the function has for safety. Systems performing functions of different categories may have different requirements for their design, implementation, and the whole lifecycle in general.

IEC 60987 [3] focuses on hardware requirements, and IEC 60880 discusses software requirements for category A systems, while IEC 62138:2004 discusses software requirements for category B or C systems. The second edition of IEC 60880 has been published in 2006, while the first edition of IEC 62138 has been published in 2004. As a system-level standard, IEC 61513 is meant to be used in association with suitable software or hardware standards. Consequently, the software standard IEC 62138 often refers to IEC 61513 for general guidelines and then proceeds to add software-specific requirements.

Most nuclear safety standards have a long history. Most of them are either second or third generation publications. Each standard has its own development cycle. For example IEC 61513:2011 is already a second edition, being initially published in 2001. It is quite normal that some major changes are done between editions to keep the content relevant and to cover new safety issues. The development cycle is quite heavy, having typically several balloting phases. As a typical result of balloting and consensus, most radical change proposals are either excluded or expressed at a high abstraction level.

IEC 62138 is currently in active development in IEC SC 45A, to be published as a second edition after passing the current CD phase and then the subsequent DIS and FDIS balloting phases. This research report is based on the CD1 version [5]. CD1 version of IEC 62138 provides requirements for the software of computer-based I&C systems of class 2 or class 3. That is a potential major change in scope, as the term “category” is not used anymore. Quite obviously, the term “class” means the same as “safety class”. As a consequence, categories A, B and C are now more hidden in the draft. Safety classes are discussed in section 4.1 of this report.

2.3 Generic IEC safety standard for I&C systems

Additionally, IEC has published IEC 61508 as a generic safety standard for safety-related electrical/electronic/programmable electronic (E/E/EP) systems. The standard is also a second edition, published in 2010. The first full version, including seven parts, was published in 1998. Predecessors of the standard family have already been published in the 1980´s. The standard is developed in an integrated way, maintaining the same timetable and balloting cycles for all parts. In this way it is kept consistent.

IEC 61508 is the main “mother standard” for many domain-specific standard families. One of the best and most ambitious examples is Automotive safety standard ISO 26262 for systems, hardware and software. Process control and railways are also good examples about tight integration between generic and domain specific safety requirements and safety management. In some other domains the closeness with IEC 61508 is not so clear, for example in electro-medical, avionics and space domains.

IEC 61508-1 [6] and IEC 61508-2 [7] are the parts of the standard that discuss general requirements for systems, whereas IEC 61508-3 [8] focuses specifically on software requirements. IEC 61508-4 [9] defines terminology used throughout the standard. Parts of the generic standard IEC 61508 that are relevant for the scope of this comparison are introduced in Table 3 along with their nuclear domain counterparts. The rows in Table 3 represent the correspondence between parts of IEC 61508 and nuclear domain IEC standards. As can be seen from the table, IEC 61513 provides the interpretation of the general requirements of IEC 61508-1, IEC 61508-2 and IEC 61508-4, for the nuclear application sector. In this framework, IEC 60880 and IEC 62138 correspond to IEC 61508-3 for the nuclear application sector.

Table 3. Correspondence between IEC standards for nuclear and generic domains

Standard	Domain	Scope	Standard	Domain	Scope
IEC 61513	Nuclear	General requirements for I&C systems	IEC 61508-1	Generic	General requirements for I&C systems
			IEC 61508-2	Generic	Requirements for E/E/EP systems
			IEC 61508-4	Generic	Definitions and abbreviations
IEC 61226	Nuclear	Classification of I&C functions	-	-	-
IEC 60987	Nuclear	Hardware requirements for I&C systems	IEC 61508-2	Generic	Requirements for E/E/EP systems
IEC 60880	Nuclear	Software requirements for I&C systems performing category A functions	IEC 61508-3	Generic	Software requirements for I&C systems
IEC 62138	Nuclear	Software requirements for I&C systems performing category B and C functions			

Just as IEC 62138 often refers to its system-level counterpart, IEC 61508-3 regularly refers to the system-level standard IEC 61508-1 for general requirements that need to be followed and then elaborates on them by introducing software-specific requirements. IEC 62138, IEC 61513 and other related standards have their own lifecycle, development project and publishing timetable, and are not so tightly interconnected and integrated as the IEC 61508 family.

3. Comparison of standards IEC 62138 and IEC 61508-3

3.1 Comparability

IEC 62138 has been designed to be the partial nuclear domain interpretation of IEC 61508-3. Therefore both the standards utilize a similar lifecycle framework in their approach on system development. Their structures are largely consistent with each other and it is relatively easy to locate corresponding sections and clauses between the standards for comparison. In many cases, requirements in both standards are also quite similar. However, in some aspects, their approaches differ quite significantly and differences both in general terminology and scope as well as individual sections and clauses can be identified.

3.2 Goals

As IEC 62138 is being updated, differences and changes to the generic safety standard IEC 61508-3 should be taken into account in the process. The goal of the comparison is to identify the most significant differences in concepts, scope and terminology, as well as individual sections and clauses between IEC 62138 new CD1 draft and IEC 61508-3 to aid the renewal process.

3.3 Method

As it was noted, both software standards IEC 61508-3 and IEC 62138 regularly refer to their system-level counterparts IEC 61508-1 and IEC 61513, respectively. Comparison of these external standards is outside the scope of this study.

The comparison itself consists of four parts:

1. General differences in concepts, scope and terminology between the standards are discussed

2. The sections of the standards are divided into 10 themes that are considered relevant in the comparison

3. All the sections are mapped between IEC 61508-3 and IEC 62138 accordingly

4. All the individual sections corresponding to a theme are then compared in more detail, and most significant differences between sections are identified

Results of the comparison are introduced in section 4.

4. Results of comparison

4.1 Differences in concepts and scope

In general, IEC 61508-3 gives more detailed recommendations on which practical means and measures should be used in each of the phases of the software lifecycle. These techniques are listed in the standard's annexes A and B. The standard then uses probabilistic *safety integrity levels* (SIL) to determine which of these particular techniques listed in the annexes should be utilized or supported in a given situation. For instance, the use of formal proof in software verification is not recommended for SIL1, recommended for SIL2 and SIL3, and highly recommended for SIL4. A highly safety-critical system should typically reach a high SIL, and more demanding measures should be used to develop such systems.. However, the standard also recognizes that software projects deeply depend on project-specific circumstances. Consequently, none of the suggested features are firmly required. Instead, they are classified as "recommended" or "highly recommended", depending on the SIL in question.

For each of the lifecycle phases IEC 61508 also introduces a list of relevant properties. For example, properties for the software safety requirements specification phase include:

1. Completeness with respect to the safety needs to be addressed by software
2. Correctness with respect to the safety needs to be addressed by software
3. Freedom from intrinsic specification faults, including freedom from ambiguity
4. Understandability of safety requirements
5. Freedom from adverse interference of non-safety functions with the safety needs to be addressed by software
6. Capability of providing a basis for verification and validation [10]

The rest of the properties are defined in IEC 61508-7. In annex C, IEC 61508-3 ranks each of the techniques listed in annexes A and B based on how effective they are in achieving the properties listed in IEC 61508-7. Ranking is done for each of the lifecycle phases accordingly. To achieve this, IEC 61508-3 uses the concept of rigour, which is ranked on an informal scale R1 – R3, R3 being the highest level. Levels R1 – R3 are defined in Table 4.

Table 4. Rigour levels in IEC 61508 [10]

R1	without objective acceptance criteria, or with limited objective acceptance criteria. E.g., black-box testing based on judgement, field trials.
R2	with objective acceptance criteria that can give a high level of confidence that the required property is achieved (exceptions to be identified & justified); e.g., test or analysis techniques with coverage metrics, coverage of checklists.
R3	with objective, systematic reasoning that the required property is achieved. E.g. formal proof, demonstrated adherence to architectural constraints that guarantee the property.

IEC 61508-3 suggests that the linking between individual techniques and properties using rigour levels can be used for comparison purposes when trying to compare a set of techniques for a given case, or as evidence when trying to justify the selection of a certain technique.

On the other hand, IEC 62138 doesn't emphasize specific techniques and measures or use rigour levels to link individual techniques to properties. Nor does it include the concept of safety integrity levels. Instead, it considers two different classes of systems - safety-classes 2 and 3. Contrary to SIL, safety-class is a deterministic measure. Class 2 systems are more critical to safety, and generally they have equal or more extensive requirements than class 3 systems. Deterministic measures have traditionally been used in the nuclear domain over probabilistic ones. Also, since the behaviour of software is deterministic, determining the probability for a software failure would be particularly difficult.

Overall, IEC 61508-3 gives more detailed recommendations on which techniques should be used in each of the phases of the software lifecycle, what are the most important properties of a given phase, and which techniques are most rigorous in satisfying these properties. On the other hand, IEC 62138 focuses on listing objectives and goals to be achieved in a given phase of the lifecycle. It doesn't contain annexes with extensive tables of techniques to be used. Since IEC 62138 targets only nuclear-domain systems of classes 2 and 3, it also has a more specialized scope.

4.2 Comparison of terminology

IEC 61508 refers to *safety-related systems* in general while IEC 62138 refers to *systems important to safety* (i.e. important to nuclear safety) [5]. Nuclear domain uses also the term "safety system". It is mainly used in safety class 1 context, and is therefore not relevant in IEC 62138. The generic term "systems important to safety" is suitable as a high level concept for all safety classes in nuclear domain. In IEC 61508, the term "safety-related system" is widely used, because the standard covers systems and software at all SIL 1 – 4 in all domains. The high level term "safety-related" is therefore a suitable high level concept in IEC 61508.

IEC 61508-3 divides supporting software tools into *on-line support tools* (tools that can directly influence the safety-related system during its runtime) and *off-line support tools* (tools that support a phase of the software development lifecycle and that cannot directly influence the safety-related system during its run time). It then divides off-line support tools into classes T1, T2 and T3 in the following manner:

- T1: generates no outputs which can contribute to the executable code (e.g. text editors).
- T2: supports the tests or verification of the design or executable code, cannot directly create errors in the executable software (e.g. static code analysis tools).
- T3: generates outputs which can contribute to the executable code of the safety related system (e.g. compilers). [9]

In IEC 62138, the term *software tool* corresponds to the term off-line support tool in IEC 61508, but the terminology in IEC 62138 doesn't include classes T1 – T3. However, the standard makes a distinction between software tools "which only might lead to overlooking already existing faults" and "software tools which might introduce faults in software or in system design". These roughly correspond to classes T2 and T3 in IEC 61508.

4.3 Selection of themes

When contents of the standards were being reviewed, 10 individual themes were identified as the basis for the comparison of individual standard sections. Themes were chosen using expert judgement, based on their relevance regarding the comparison. Also, only themes that were clearly recognizable in both standards were chosen. Identified themes are:

1. Configuration management
2. Lifecycle
3. Requirements
4. Validation
5. Design and implementation
6. Selection of pre-developed software
7. Selection of tools
8. Integration
9. Modification
10. Verification

Each theme corresponds to one or multiple sections in both standards.

4.4 Correspondence of sections

Correspondence between theme sections is illustrated in Table 5.

Table 5. Correspondence of theme sections (adapted from [5])

Theme	IEC 61508-3	IEC 62138
1. Configuration management	6.2.3 Software configuration management	5.1.3 Configuration Management
2. Lifecycle	7.1 General	4.4 Software and System Safety Lifecycles 5.1.1 Software Safety Lifecycle – Software Quality Assurance
3. Requirements	7.2 Software safety requirements specification	5.3 Software requirement specification
4. Validation	7.3 Validation plan for software aspects of system safety 7.7 Software aspects of system safety validation	5.7 Software aspects of system validation
5. Design and implementation	7.4 Software design and development	5.4 Software design 5.5 Implementation of new software
6. Selection of pre-developed software	7.4.2 General requirements	5.2 Selection of pre-developed software
7. Selection of tools	7.4.4 Requirements for support tools, including programming languages	5.1.4 Selection and use of software tools

		5.1.5 Selection of languages
8. Integration	7.5 Programmable electronics integration (hardware and software)	5.6 Software aspects of system integration
9. Modification	7.6 Software operation and modification procedures 7.8 Software modification	5.10 Software modification
10. Verification	7.9 Software verification	5.1.2 Verification

Sections that were not selected as theme sections and sections that were only found from one of the standards are listed in Table 6.

Table 6. Rest of the sections (adapted from [5])

IEC 61508-3	**IEC 62138**
8 Functional safety assessment	In the nuclear sector, this assessment is connected to the licensing process and depends on the safety bodies and national regulations.
Annex A (normative) Guide to the selection of techniques and measures	Requirements focus on the objectives to be achieved rather than on the techniques and measures.
Annex B (informative) Detailed tables	Requirements focus on the objectives to be achieved rather than on the techniques and measures.
Addresses defences against common cause failure due to software, in particular in **appendix C**, but has no dedicated clause on this topic.	**5.11 Defences against common cause failure due to software**
Annex D (normative) Safety manual for compliant items – additional requirements for software elements	**5.2.2 Documentation for Safety**
Annex F (informative) Techniques for achieving non-interference between software elements on a single computer	-
Annex G (informative) Guidance for tailoring lifecycles associated with data driven systems	-
Outside the scope of IEC 61508-3 as it is addressed in **IEC 61508-1**	**5.8 Installation of software on site**
Outside the scope of IEC 61508-3 as it is addressed in **IEC 61508-1**	**5.9 Anomaly reports**
Addressed **in IEC 61508-1 (annex A)**	**Annex A (informative) Typical list of**

	software documentation

Additionally, IEC 61508-3 discusses its relationship with IEC 61508-2 in annex E. IEC 62138 discusses its correspondence with the system level standard IEC 61513 in annex B, and in annex C, it also includes a brief comparison between IEC 62138 and IEC 61508.

4.5 Detailed comparison of theme sections

Individual sections corresponding to selected themes in IEC 61508-3 and IEC 62138 were compared against each other. Table 7 shows the main differences that were identified between the standards.

Table 7. Comparison of theme sections

Theme	**Included in IEC 61508-3, but not in IEC 62138**	**Included in IEC 62138, but not in IEC 61508-3**
1. Configuration management	A detailed list of example documents that shall be under configuration management (6.2.3). More detailed instructions of how software configuration management should be implemented (6.2.3). For example part e) states that “configuration management should ensure that appropriate methods are implemented to load software into the target system”, since loading software (e.g. firmware) to some targets might require special measures. Overall, IEC 62138 discusses the meaning and objectives of configuration management, while IEC 61508-3 discusses more practical means to maintain an appropriate configuration management.	-
2. Lifecycle	The concept of V-model (7.1 Figure 6). The use of any software lifecycle model is permitted (7.1.2.2), as long as it satisfies all the requirements on clause 7. In this case, the V-model may also be tailored to the needs of the project. Also, if the lifecycle model satisfies all the requirements, it is allowed to be customized for the needs of the particular project, e.g. for a smaller software project	The software safety lifecycle is closely integrated to the system safety lifecycle which is presented in IEC 61513. The lifecycle model is sequential, and resembles the waterfall model. The software safety lifecycle has different paths for the implementation of application software and new operational system software. (4.4, Figures 2-4)

	it is allowed to merge some of the lifecycle phases listed in Table 1 (7.1.2.4). Overall, IEC 61508-3 offers more room for customization of the software lifecycle model.	
3. Requirements	Identifying non-safety related functions and independence requirements between functions is required (7.2.2.9, 7.2.2.10). The importance of close co-operation between the hardware and software developers is clearly highlighted, since software design specifics might have impact on hardware architecture (7.2.2.2). Requirements for safety-related software are derived directly from system requirements (7.2.2.2) and made available to the software developer. Thus, the software developer is required to evaluate software requirements in detail to ensure that they are adequately specified (7.2.2.5).	More details about the required content for the requirement specification. IEC 61508-3 references IEC 61508-2 and IEC 61508-7 for similar specifics (7.2.2.2).
4. Validation	More detailed requirements for the contents of the validation plan, such as: a) details of when the validation shall take place; b) details of those who shall carry out the validation; c) identification of the relevant modes of the EUC operation. (7.3.2.2) Cases where discrepancies occur in validation are more clearly taken into consideration. In such cases, validation may be continued or a change request can be issued, resulting in returning to an earlier part of the development lifecycle. (7.2.2.6)	Audition of the results of software validation by persons not directly involved in the validation process is required (5.7.7, 5.7.8). At least one person who didn't engage in design and implementation of the software, shall participate in the creation of the validation plan (5.7.10).
5. Design and implementation	Requirements for the selection of software design method (7.4.2.2). Techniques and measures that should be used in architecture	Focuses more on the objectives of software design, and particularly on the objectives and contents of the Software Design

	and detailed design (7.4.3, 7.4.5). More emphasis on software modularity (7.4.5.3, 7.4.5.4).	Specification document (5.4). Separate requirements for implementation in application-oriented languages and general-purpose languages (5.5.3, 5.5.4).
6. Selection of pre-developed software	Specific requirements for systems that consist of pre-existing functionality that is configured by data to meet specific application requirements (7.4.2.14).	Divides pre-developed software into software components that would be integrated into other components (e.g. RTOS) and complete operational system software. Requirements for complete operational system software are more extensive than the ones for software components (5.2.3.1.1).
7. Selection of tools	Need to consider availability of selected tools over the whole lifetime of the system (7.4.4.2). Competencies of users regarding to selected tools should be considered (7.4.4.2). Precise requirements for the documentation of the results of tool validation (7.4.4.7). Development of safety-related software shall be done according to suitable coding standards (7.4.4.12). Justification of languages that don’t satisfy requirements (7.4.4.11). Each new version of off-line support tool shall be qualified (7.4.4.18).	Tools that might influence the correctness of software have to be identified and recorded in the Quality Assurance Plan (5.1.4.5). In class 2 systems, requires tracing of the use of tools that might introduce faults in software design or that might lead to already existing faults being overlooked (5.1.4.8). Application-oriented languages should be preferred over general-purpose ones, and machine-level languages should be used only if their use is justified (5.1.5.2, 5.1.5.3).
8. Integration	In the case of a failure found in the integration testing, documenting the cause for the failure is clearly required. Additionally, if any resulting modifications to the software are made, such modifications shall be subject to an impact case analysis which determines all other software elements impacted, and whether any necessary re-verification must be done. (7.5.2.6, 7.5.2.8)	-

9. Modification	An authorized software modification request is required (7.8.2.2). It states: a) the hazards which may be affected; b) the proposed modification; c) the reasons for modification. Hazard risk analysis may be required (7.8.2.3).	Regression software integration / validation (5.10.3, 5.10.4).
10. Verification	Verification of software shall be planned concurrently with the development, for each phase of the software lifecycle (7.9.2.1). More detailed list of factors that the software verification planning shall address (7.9.2.2). More detailed rules for the results of verification. For example, IEC 62138 states that results must have the required contents and comply with any resolution agreed (5.1.2.2), while IEC 61508-3 lists concrete elements that the results should consider, such as code testability and readability (7.9.2.6). More detailed requirements for the contents of verification of different areas. For example, factors that should be considered when verifying software safety requirements, architecture, design, code, data, timing performance and so on, are listed (7.9.2.8 – 7.9.2.14).	Persons who participated in the activity that is being verified shall not participate in the verification (5.1.2.3).

5. Comparing the suitability of the standards for regulatory needs

5.1 Introduction

The standards were also evaluated against their usefulness to cover and satisfy regulatory needs in the nuclear domain. Since this research is done in the Finnish research program SAFIR2014, the regulator in the context of this report is the Finnish Radiation and Nuclear Safety Authority, STUK.

Regulatory needs in software safety for nuclear domain that are relevant in this study are described mainly in two guides:

- STUK YVL Guide E.7: Electrical and I&C equipment of a nuclear facility (later called **E.7**) [11].
- Common Position 2013: Licensing of safety critical software for nuclear reactors: Common position of seven European nuclear regulators and authorised technical support organisations (later called **CP2013**) [12].

CP2013 was developed by a task force established by Western European Nuclear Regulators' Association (WENRA). Seven nuclear regulatory authorities in Western European countries participated in the task force.

The suitability and coverage problem can be summarised in two questions:

- How well the standards IEC 62138 CD1:2014 and IEC 61508-3:2010 cover requirements and other topics for software safety, as they are described in E.7 and CP2013?
- How well the full compliance with either one of the standards could satisfy and cover regulatory needs? In other words, are they good enough to be recommended as primary references to deliver and operate safety class 2 or 3 software in nuclear power plants?

As discussed in sub sections 4.1 and 4.2, there are significant differences in the fundamental concepts between IEC 62138 and IEC 61508-3. Arguably the most notable difference is the lack of the SIL concept in IEC 62138.

Almost all method tables in IEC 61508-3 Annexes A – C can be assigned into two groups: SIL 1 and 2, most methods and techniques being then “Recommended”. In SIL 3 and 4 methods and techniques are almost always “Highly Recommended”. To make comparison easier, IEC 62138 is interpreted to be near SIL 3 grading in IEC 61508-3. This interpretation is needed in some items in E.7 and in CP2013, to evaluate to coverage and the overall goodness of selected standards.

IEC 61508-3 is only one part in a much bigger IEC 61508 standard family. This suitability and coverage study is limited to only part 3 of the family. Many important topics in selected nuclear regulatory guides can be covered in some other part of IEC 61508. Especially relevant are parts 1 and 2 in topics in system and hardware safety. Of course, also IEC 62138 is a part of a much bigger family of nuclear safety standards.

As most standards, IEC 62138 and IEC 61508-3 also have both normative and informative parts. The body text is normative in both standards. The lists of normative references are also normative. Some Annexes are informative and some normative in both standards. In evaluating the overall goodness of standards the difference between normative vs informative parts and requirements was not considered very significant.

The results of suitability and coverage analysis are presented in section 5.2. In sections 5.2.1 and 5.2.2, both E.7 and CP2013 are explained and their potential similarities and overlaps with compared standards are described. The suitability and quality of compared standards are further discussed in section 5.3.

The coverage and coverage of compared standards against E.7 and CP2013 requirements was done by classifying the mutual mapping with a 4-point scale:

- F, Fully satisfied. It means that there is good and clear mapping between any selected regulatory requirement and the standard. In a typical case it was also at same abstraction level. Full compliance with the standard means also that there is clear evidence to verify the selected regulatory requirement.

- L, Largely satisfied. It means that there is good mapping between any selected regulatory requirement and the standard. There can be some difference in the language, terms and in the abstraction level. Some interpretation may be needed to check the correspondence. Compliance with standard means, that the selected requirement is satisfied but there is some risk in interpretation.

- P, Partially satisfied. It means that there is identified mapping between any selected regulatory requirement and the standard. There is clear difference either in the language, terms or in the abstraction level. Compliance with standard does not guarantee that the regulatory requirement can be verified without further actions.

- N, Not satisfied. The link and mapping between any regulatory requirement and the standard is missing. Maybe it is caused by difference in terminology and abstraction level, maybe some other reason. Compliance with standard does not mean anything in relation to the regulatory requirement.

- NA, Not Applicable. This is an additional rating value. Either the regulatory guide or the standard may have some requirement, which is not at all relevant for the other. If both compared standards are rated as NA, comparison and validation becomes meaningless and such item can be removed.

5.2 Coverage analysis of compared standards with selected nuclear regulatory guides

5.2.1 STUK YVL E.7 Electrical and I&C equipment of a nuclear facility

YVL E.7 has 379 requirements, presented clearly with an ID number, title and description. Around 50 requirements are directly related to software. When also system and hardware are taken into account, about 140 requirements are covered. Some E.7 requirements are rather generic and wide, and can be classified in several ways.

Because IEC 62138 covers only safety class 2 and 3 requirements for software, additional filtering is needed. Most E.7 requirements cover all safety classes. Some requirements focus only in safety class 1, and they can be excluded.[1] As a result, 36 requirements are still relevant in E.7 when compared with IEC 62138. That is about 10 % from the whole E.7 content, and 70 % from all software safety related requirements. Still some additional filtering and elimination was needed, because some software requirements may not be relevant for either standard (they are not applicable). Finally, 33 requirements remained for this analysis.

[1] STUK has different classification for safety than IAEA and what is used in IEC 62138. This is taken into account in this analysis.

Main topics in E.7 including direct requirements for software in safety class 2 and 3 are (because E.7 is only in Finnish, also titles in Finnish are presented):

- 5.9 Operational experience (käyttökokemukset)
- 5.10 Type acceptance (tyyppihyväksyntä)
- 6.1 Special requirements for devices including software (ohjelmistopohjaisten laitteiden erityisvaatimukset)
- 6.2 Qualification of software platforms and application software (perusjärjestelmän ja sovelluksen ohjelmiston kelpoistaminen)
- 6.3 Software engineering methods and processes (ohjelmistojen suunnittelumenetelmät ja –prosessit)
- 6.4 Software tools (ohjelmistotyökalut)
- 6.5 Cybersecurity (kyberturvallisuus ja tiedonsiirrollinen erotus)
- 6.6 Pre-developed software (olemassa oleva ohjelmisto)
- 6.7 Software testing (ohjelmiston testaus)

Results of the coverage and coverage analysis of compared standards against selected YVL E.7 requirements are presented in Table 8.

Table 8. Analysis of the results of YVL E.7 vs. compared standards

Rating (see explanations in section 5.1)	Coverage of IEC 62138:2014 CD1, # of E.7 items	Coverage of IEC 61508-3:2010, # of E.7 items
F, Fully satisfied	13	7
L, Largely satisfied	7	7
P, Partially satisfied	5	7
N, Not satisfied	7	8
NA, Not Applicable	1	4
Total # of analysed requirements in YVL E.7	33	33

The coverage of IEC 62138 is better than in IEC 61508-3, as can be seen in Table 8. This is not a surprise, because IEC 62138 is specifically a nuclear domain standard. More than 60 % of all relevant requirements can be verified by demonstrating compliance with IEC 62138.

The relatively weak coverage of IEC 61508-3 with E.7 can be explained mainly by different terminology and lack of specific nuclear domain context.

5.2.2 Common Position 2013

Similar analysis was also done for the requirements in CP2013. Requirements in CP2013 are identified by a detailed subchapter numbering up to 5-digit level. They are classified in two main groups: generic licensing issues (requirement ID starts with 1) and life cycle phase licensing issues (requirement ID starts with 2). Additionally, each chapter has rationale, issues involved, common position and recommended practices as standardized third level subchapters.

Common Position 2013 is a very detailed guide. It has about 340 software safety related requirements in Common Position subchapters. That is much more than in E.7. Recommended practices chapters have around similar amount of recommendations and practical hints. In this report only the requirements of Common Position are included in order to check suitability and coverage of the compared standards.

After filtering and analysis 71 requirements remained. They are in the following chapters:

- 1.1 Safety demonstration
- 1.2 System Classes, Function Categories and Graded Requirements for Software
- 1.4 Pre-existing software
- 1.7 Software Quality Assurance Programme and Plan
- 1.11. Graded Requirements for Safety Related Systems (New and Pre-existing Software)
- 1.12 Software Design Diversity
- 1.13 Software Reliability
- 1.14 Use of Operating Experience
- 2.3 Software Requirements, Architecture and Design
    - 2.3.3.1 Software Functional and Non-Functional Requirements
    - 2.3.3.2 Software Architecture and Design
- 2.4 Software Implementation (coding, subroutines, data structures and addressing, defensive programming, language and compiler, operating system, support software)
- 2.5 Verification (tools, planning, coverage, traceability, documentation, independent verification)
- 2.6 Validation and Commissioning
- 2.7 Change control and configuration management (modification, maintenance, CM)

Results of the coverage and coverage analysis of compared standards against selected CP2013 requirements are presented in Table 9.

Table 9. Analysis results of CP2013 vs. compared standards

Rating (see explanations in section 5.1)	Coverage of IEC 62138:2014 CD1, # of CP2013 items	Coverage of IEC 61508-3:2010, # of CP2013 items
F, Fully satisfied	14	30
L, Largely satisfied	23	10

P, Partially satisfied	10	18
N, Not satisfied	21	10
NA, Not Applicable	3	3
Total # of analysed requirements in CP2013	71	71

In this analysis, IEC 61508-3 has clearly better coverage than IEC 62138:2014 CD1. It is more detailed, as also CP2013. In this analysis, IEC 62138 can be seen as a quite narrow standard without all required details.

5.3 Potential deficiencies in the compared standards

Detailed analysis of YVL E.7 and Common Position vs compared standards lead to a long list of potential deficiencies. In some cases, there can be major difference in some issue between the compared standards. In some other cases, both standards may be weak in relation to regulatory requirements. List of potential deficiencies is in table 10. Each item is also discussed at general level. It is also explained shortly why there may be a gap compared with regulatory requirements.

Many topics and issues below can be seen as “obvious” or “normal professional practice”. Anyway, in such a case they are treated as potential deficiencies in the standards. If an issue is not mentioned in the standard, it may be left out also in the compliance statement or certificate. In some cases the issue could be identified by suitable interpretation of the standard, but that should be avoided if possible. In such cases there is a risk that the issue is not fully recognized or is left out in purpose. All issues explicitly mentioned in the regulatory guides should be handled somehow.

Some identified topics in table 10 may be intentionally outside of the current scope of compared standards. IEC 62138 has strict focus in safety of class 2 and 3 software, and therefore for example security is not included in the current CD1 version. Many analyses may not be in the scope, because they may be in other standards or are based on operational experience and data.

Table 10. List of potential deficiencies the compared standard(s)

Topic, requirement example in E.7 and/or in CP2013.	**General rationale**	**Discussion, findings**
Operational experience. E.7 requirement # 563, 564. CP2013 chapter 1.4.3.10.	Operational experience is needed as input for various reliability calculations.	Operational experience is well presented in IEC 62138, mainly in pre-developed software. It is missing in IEC 61508-3, maybe because focus is in development, not in analysis or operation. **Note**: This topic may be covered in some other standards or parts.
Self-diagnostic.	Self-diagnostic is important to achieve fault tolerance	Surveillance or diagnostic is required for pre-developed software

E.7 # 609 The term used in CP2013 is *Autocontrol*, see Table 2 in Part 1.	and high integrity (also in operation phase).	in both standards.
Traceability. E.7 # 612. Table 2 in CP2013 part 1 for pre-developed software, also CP2013 2.5.3.4	Bi-directional traceability is used in test coverage calculations, change management etc. Traceability should be detailed and end-to-end, including also code level. Traceability should be a property in architecture documents, not only in configuration management.	Traceability as a requirement is identified in current generic standards. In safety standards it seems to be more implicit, for example in requirements for tools and change control. In IEC 61508-3 this requirement is well identified in Annex A –C method tables.
Qualification (planning). E.7, # 614. Most parts of CP2013.	Qualification is based on evidence, and evidence is most cost-effective to collect during development.	Both standards have lot of attention in verification. Qualification is mentioned mainly as part of tool selection process.
Safety manual (and other similar documentation), safety plan. CP2013 2.3.3.1.3.	Safety manual is a good way to document what needs to be said about achievement and demonstration of safety.	List of recommended documents is in IEC 62138 Annex A. It does not include safety manual or safety plan. Safety manual is defined at detailed level in IEC 61508-3 Annex D.
Safety assessment (also type testing, certification). E.7 # 616. CP2013 uses term safety demonstration, and is described in chapter 1.1.	Safety assessment is a basic approach to check achievement of safety. Quite similar topics are product evaluation, safety case etc.	Safety assessment is quite well defined (at least between the lines) in both compared standards. It is very clearly defined in IEC 61508-3, see chapter 7.1 Table 1, and chapter 8Table A.10.
Quality assurance. E.7 # 622. CP2013 chapter 1.7.	QA is an independent activity at project and/or organizational level to check adherence with policy and standards. It is also a mechanism for learning from previous and current activities.	Quality assurance is very centric in IEC 62138. QA plan is recommended. QA is less transparent in IEC 61508-3. Maybe some other wording is used instead.
Cybersecurity, information security.	Security controls prevent intentional harm.	Security is not in the current scope of IEC 62138.

E.7 # 631. CP2013 chapter 1.8.	Security and safety are closely related. Better security leads to better safety.	It is almost missing and implicit in IEC 61508-3. It is mentioned only in safety manual, Annex D.
Non-executable code. E.7 # 633.	Additional and non-executable code may cause unintended risks and failures also in safety related code.	Both standards have requirements for executable code and version control. Non-executable code is not clearly defined in compared standards.
Pre-developed software, suitability analysis. E.7 # 641. Chapter 1.4 in CP2013.	External components, libraries, system software etc. shall be analysed to avoid potential risks. Also qualification or certification is required in several domains and higher safety classes / integrity levels.	Suitability analysis is well presented in IEC 62138. See chapter 6.2.2. Suitability analysis is only implicit or mainly absent in IEC 61508-3. It is required in the selection of tools.
Static and dynamic tests. E.7 # 648. CP2013 1.15.3.9. (and many other subchapters). Terms static and dynamic analysis are used.	Common term in several standards is verification. It covers both static and dynamic tests and analyses. Both approaches are needed for quality and cost reasons in all kinds of software.	(Tool based) static analysis is well defined in IEC 62138. Also dynamic testing is well required. IEC 61508-3 is more detailed in verification in general. Static and dynamic analyses are required especially in tables in Annex A, B and C.
Test coverage. E.7 # 650. CP2013 1.7.3.9. , 2.5.3.2. etc.	Goal value for test coverage is needed to specify required test cases. Standard ISO/IEC 29119-4 specifies a large number of test coverage criteria.	IEC 61508-3 Annex B, table B.2 has some test coverage items, but not all. This topic is missing or implicit in IEC 62138.
Development process. E.7 # 618. CP2013 1.4.3.5 and Part 2.	Development process is necessary in software, because it can assessed better than the software product itself. Process quality enables product quality.	Requirement for development process (capability) is only weakly and implicitly required in both standards. Quite obviously, development process is assumed to exist in safety lifecycle.
Diversity. CP2013 1.12.3.1. – 5, 8, 12, 13.	Diversity is required to achieve higher reliability and fault tolerance in safety systems. Diversity is needed also to	Diversity is not considered to be software specific and is not mentioned in IEC 62138. Diversity is weakly identified in IEC 61508-3, but may be more explicit

	avoid common cause failures.	in other parts.
Mixed criticality. CP 2.3.3.2.3.	It is a trend to use multicore, FPGA etc. for several purposes.	Mixed criticality is typically not considered in nuclear domain, at least not in SC 1. There are no requirements about that in either selected standards.
Coding directives. CP2013 2.4.3.2.	Without agreements and policy about code standards, it leads to variation and more difficult change control.	IEC 62138 does not mention coding standards. IEC 61508 has it as one property, see table 7.1 point 10, and table B.1 and C.11.
Dynamic memory / storage allocation. CP2013 2.4.3.3.7, 2.4.3.8.	Dynamic memory allocation is considered as risk in high reliability and fault tolerant systems. In multicore and FPGA platforms each application should have its own strictly isolated memory.	Dynamic memory allocation is not explicitly mentioned in compared standards. This requirement is a good example of more detailed requirement in CP2013 than the compared standards.
Demonstration of safety (see also safety assessment above). CP2013 chapter 1.1, also 2.5.3.1.2	Typical solution is safety case, currently also assurance case (according to standard ISO/IEC 15026).	This is better in IEC 62138. Topic is not mentioned in IEC 61508-3, is implicit in requirements for functional safety assessment.
Software (project) management. E.7 # 618. CP2013 chapter 1.6, for example safety culture and personnel competence.	Too narrow focus on software engineering and technical issues leaves several risks and issues outside considerations. Management is needed for example in quality and risk management and in process improvement.	Both standards are weak in management topics. Standards include mainly planning and control aspects of technical management, but it could mean a lot more. IEC 62138 has normative reference to ISO9001 and ISO90003. Management is mainly in IEC 61508-1 of the standard family.

6. Conclusions

This report presents the results of a study that was performed under the research project Coverage and rationality of the software I&C safety assurance (CORSICA), to support the update process of nuclear domain I&C software safety standard IEC 62138. In the study, software safety standards IEC 62138 and IEC 61508-3 were compared with each other, and relevant differences in concepts, scope, terminology and selected major themes were identified. Additionally, both of the standards were compared against two selected regulatory guides (the Finnish YVL E.7, and Common Position 2013).

One of the most important general differences between IEC 62138 and IEC 61508-3 is the absence of safety integrity levels in IEC 62138. Instead, since it is a nuclear domain standard, IEC 62138 categorises systems into deterministic classes. In the CD1 version of IEC 62138, the expression “System performing functions of category B or C” was replaced with the expression “System of class 2 or class 3”. This was made due to clarification purposes. However, for further clarification, using the term “System of safety class 2 or 3” should be considered, since the term “class” is quite obviously meant to correspond to “safety class”.

The results of the comparison between the theme sections of IEC 62138 and IEC 61508-3 are summarised in Table 7. Most notable differences include the overall more detailed requirements and recommendations that IEC 61508-3 provides regarding techniques and measures to be used in configuration management, validation, design & implementation and verification. One matter that has proved to be surprisingly difficult in practise is making sure the tools used in a project are available during the whole lifecycle of the system. This is clearly required in IEC 61508-3, but is not mentioned in IEC 62138.

Unlike IEC 61508-3, IEC 62138 doesn’t mention the concept of V-model. Moreover, IEC 61508-3 offers more freedom in selecting a lifecycle model suitable for the situation in question. It also permits modification of the V-model according to the needs of a particular project.

It should also be noted that, as explained in section 2.3, IEC 61508 is a series of standards consisting of 7 parts, whereas IEC 62138 is an independent standard intended to be used in collaboration with other nuclear domain IEC standards such as IEC 61513. Consequently, while the parts in IEC 61508 are being updated in a similar frequency, IEC 62138 is not as closely bound to other nuclear domain IEC standards. Thus, the parts of IEC 61508 form a more consistent configuration e.g. regarding updated content and terminology.

The results derived from the comparison of IEC 62138 and IEC 61508-3 against the regulatory documents are summarised in Table 10. According to the results, IEC 62138 has deficiencies e.g. in the following areas:

- Traceability
- Safety manual
- Test coverage
- Development process capability
- Coding directives
- Management topics

IEC 62138 covers the requirements in YVL E.7 quite well, but does worse when comparing against CP2013. However, one has to bear in mind that some of the areas IEC 62138 seems to be lacking in are maintained that way in purpose. IEC 62138 is a software-specific standard, and in many occasions it refers to the system-level standard IEC 61513, which provides more requirements.

References

1. International Electrotechnical Commission, IEC 61226, Nuclear power plants – Instrumentation and control systems important to safety – Classification of instrumentation and control functions, IEC: 2005.
2. International Electrotechnical Commission, IEC 61513, Nuclear power plants – Instrumentation and control for systems important to safety – General requirements for systems, IEC: 2001.
3. International Electrotechnical Commission, IEC 60987, Nuclear power plants - Instrumentation and control important to safety - Hardware design requirements for computer-based systems, IEC: 2007.
4. International Electrotechnical Commission, IEC 60880, Nuclear power plants – Instrumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions, IEC: 2006.
5. International Electrotechnical Commission, IEC 62138, Nuclear power plants - Instrumentation and control important for safety - Software aspects for computer-based systems performing category B or C functions, Ed2/CD1, IEC: 2014.
6. International Electrotechnical Commission, IEC 61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems – Part 1: General requirements. Second edition, IEC 2010.
7. International Electrotechnical Commission, IEC 61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems – Part 2: Requirements for electrical/electronic/programmable electronic safety-related systems. Second edition, IEC 2010.
8. International Electrotechnical Commission, IEC 61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems – Part 3: Software requirements. Second edition, IEC 2010.
9. International Electrotechnical Commission, IEC 61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems – Part 4: Definitions and abbreviations. Second edition, IEC 2010.
10. International Electrotechnical Commission, IEC 61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems – Part 7: Definitions and abbreviations. Second edition, IEC 2010.
11. STUK, Radiation and Nuclear Safety Authority, Guide YVL E.7 Electrical and I&C equipment of a nuclear facility, 2013.
12. European Commission's Advisory Experts Group, Nuclear Regulators Working Group, Licensing of safety critical software for nuclear reactors - Common Position of seven European nuclear regulators and authorized technical support organizations, Revision 2013 (2013).