UPPSALA
UNIVERSITET

UPTEC E 17010
Examensarbete 30 hp
Juni 2017

The effect of the update of the European standard EN 50128

- The management of the safety of the software applications for railway applications

Åsa Nordström

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

The effect of the update of the European standard EN 50128

Åsa Nordström

The European standard EN 50128 "Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems" is one of the European standards for European Railway systems. It is intended for software aspects, specifying procedures and technical requirements for the development of programmable Electronic systems, which are used in railway control and protection applications. Since 2017-04-25 the original version EN 50128:2001 has been replaced by the updated new version EN 50128:2011. The update is quite extensive and will effect many parts of the existing Railway systems. The aim of this study is to investigate the effect of the EN 50128 update.

The work for this study includes literature study, document research and interview with the relevant supplies and experts. Qualitative and quantitative methods have been used in the study to reach the possible best results.

The effects due to the EN 50128 update have been extensively investigated. The following issues have been addressed:

- How can the update of the standard EN 50128 be done smoothly by the companies?

- How much money have they spent to update their process to follow the 2011 version?

- What parts of the process have been the most extensive and expensive to change due to the standard update?

The results of the work are useful for an organized and professional assessor to help and support the companies dealing with this complex software, in order for them to be prepared for the upcoming standard update as well as possible. If the affected companies have been proactive in their own development of their methods/techniques, the 2011 version of the standard will not be a major work to follow for their process. A standard is a guideline and a support in the way to a safer system.

Handledare: Per Ocklind
Ämnesgranskare: Ping Wu
Examinator: Mikael Bergkvist
ISSN: 1654-7616, UPTEC E 17010

Acknowledgment

This master thesis work is the final part of the Master Programme in Electrical Engineering, with the specialization of electrical system control. It has been conducted in the Swedish Transport Administration (TRV) in association with the ERTMS project, during spring 2017. It has not been an easy ride, but I have had good support along the way. Without civil order, I would like to thank the persons below for being a great support throughout the process.

Per Ocklind - My supervisor at ERTMS TRV. He has been an excellent support throughout the time of the study.

The supplier, who has been a great support in this study. They have had answers to all my questions, always within a short time frame. Without them, there would not have been any result to this study.

Sam Berggren and Staffan Fors - Employees at TRV and members of the Swedish national committee, SEK TK 9. Their knowledge in the subject has been a great resource in this subject.

Andrea Palermo and other members of SGA18 - The participation in this Survey Group has been a great opportunity for me to experience the work behind the standards. I have also due to this been blessed with contacts in all of Europe.

Thomas Borglin and Svensk Elstandard (SEK) - They have supported me with information and documents needed in order to make the quality of this study as good as possible. The course and lectures entered at Svensk Elstandard contributed with a lot of interesting and inspiring information in the subject of this study.

Ping Wu - My subject reviewer, Associate Professor in the division of Signals and Systems at Uppsala University. He supported and guided me whenever I needed throughout the study.

All colleagues at ERTMS TRV, my partner and everybody else that have supported me in this master thesis work.

Solna, Sweden, may 2017

Åsa Nordström

Sammanfattning

Den europeiska standarden EN 50128 ”Järnvägsapplikationer - Kommunikations-, signal- och bearbetningssystem - Programvara för järnvägskontroll och skyddssystem” är en av de europeiska standarderna för europeiska järnvägssystem. Den är avsedd för programvaruaspekter, specificering av förfaranden och tekniska krav för utveckling av programmerbara elektroniska system, vilka används i järnvägskontroll och skyddsprogram. Sedan 2017-04-25 har originalversionen EN 50128:2001 ersatts av den uppdaterade nya versionen EN 50128:2011. Uppdateringen är ganska omfattande och kommer att påverka många delar av de befintliga järnvägssystemen. Syftet med denna studie är att undersöka effekterna av EN 50128-uppdateringen.

Arbetet för denna studie innefattats av litteraturstudie, dokumentforskning och intervju med relevanta leverantörer och experter. Kvalitativa och kvantitativa metoder har använts i studien, för att nå de bästa möjliga resultaten.

Effekterna på grund av EN 50128-uppdateringen har undersökts i stor utsträckning. Följande frågor har tagits upp:

- Hur kan uppdateringen av standarden EN 50128 göras smidigt av företagen?
- Hur mycket pengar har de spenderat för att uppdatera sin process för att följa 2011-versionen?
- Vilka delar av processen har varit den mest omfattande och dyra att ändra på grund av standarduppdateringen?

Från undersökningen avslöjas det att det kan ta cirka 500 timmar för en leverantör att utföra uppdateringen för en produkt och cirka 160 timmar av de 500 timmar som behövs för att klassificera verktygen. Denna del av den uppdaterade versionen upplevdes vara viktig, tidskrävande och omfattande. Alla verktyg i processen är uppdelade i tre grupper, T1, T2 och T3, beroende på verktygets säkerhetseffekt. En beskrivning av syftet, säkerhetspåverkan, mildringar etc. måste läggas till varje verktyg. När det gäller detta omfattande arbete såväl som för andra delar av denna standard, anser leverantörerna att standarden har brist på exempel och intuitiva förklaringar. Den bör förlängas med en ytterligare beskrivande del för att ge en bättre guide och stöd för de företag som följer detta standard. Resultaten av denna studie påpekar också på att den behöver en guide (eller något liknande) från beställaren, hur standarden ska användas. Dessutom bör de kostnader som överenskommits i upphandlingarna mellan beställaren och leverantörerna dokumenteras mer ingående. På grund av brist på detaljerade dokumentationer är resultatet av studien om kostnader ganska approximativ.

Den här studien visar dessutom att det är viktigt för berörda parter att aktivt delta i standardiseringsgrupperna. Nuvarande arbetsgrupper saknar deltagare från slutanvändare, vilket kan leda till en standard som är mindre användarvänlig än den är avsedd att vara. Längs vägen för denna studie har ytterligare frågor uppmärksammats, till exempel *hur mycket TRV ska vara involverat i leverantörernas arbete? Hur djupt förhållandet är okej för en beställare och en leverantör att ha? Räcker det med en revidering vartannat år? Är SIL-klassificeringen säker nog?*

Executive Summary

The standard EN 50128 ”Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems” is one of the European railway safety standards. It focuses on software aspects specifying procedures and technical requirements for the development of programmable electronic systems which are used in railway control and protection applications.

The original version of this standard is EN 50128:2001. Due to the fast development of software, the 2001 version has been updated to the new version EN 50128:2011, which by 2017-04-25 has replaced the 2001 version. The new version includes almost double the amount of requirements and lifecycle documentations compared to the previous. As this update is extensive for the users of this standard, the ERTMS project at Trafikverket has requested this study.

The study has been carried out through literature and document research, and interviews with relevant suppliers and experts. Printed and unprinted sources have been studied, as well as trustworthy Internet sites. These interviews were both structured and unstructured with different extension in depth. Some of the issues are following on the focus:

- How can the update of the standard EN 50128 be done smoothly by the companies?
- How much money have they spent to update their process to follow the 2011 version?
- What parts of the process have been the most extensive and expensive to change due to the update?

From the study, it reveals that it may take approximately 500 hours for a supplier to execute the update for one product, and approximately 160 hours out of the 500 hours are needed for the classification of the tools. This part of the updated version was experienced to be important, time consuming and extensive. All of the tools in the process are divided into three groups, T1, T2 and T3, depending on their safety impact. A description of the purpose, safety impact, mitigations etc. needs to be added to each tool. Regarding this extensive work, as well as for other parts of this standard, the suppliers think that the standard lacks examples and intuitive explanations and it should be extended with an additional, descriptive part, to provide a better guide and support for the companies following this standard. The result of this study also point out that it needs a guide (or something similar) from the orderer, of how the standard should be used. In addition, the costs agreed of in the procurements between the orderer and the suppliers should be documented more in detail.

Moreover the present study reveals that it is important for affected parties to participate actively within the standardization working groups. The present working groups lacks participants from developers, which may result in a standard that is less user friendly than it is intended to be.

Along the way of this study, additional questions have been raised. *How much should Trafikverket be involved in the work of the supplier? How deep relation is okay for an orderer and a supplier to have? Is it enough to have a revision every other year? Is the SIL classication safe enough?*

Keywords: EN 50128, Safety Standards, Safety related Software, EN 50657, tool classification, SIL, CENELEC, TC 9X.

Contents

List of Figures

Glossary

Term	Definition
ASR	Assessor
ATC	Automatic Train Control
CEN	European Committee for Standardization
CENELEC	European Committee for Electrotechnical Standardization
COTS	Commercial off-the-shelf
DES	Designer
EOS	ERTMS Onboard System
ERTMS	European Rail Traffic Management System
ETCS	European Train Control System
ETD	End-of-Train device
ETSI	European Telecommunications Standards Institute
HR	Highly Recommended
IEC	International Electrotechnical Committee
IMP	Implementer
INT	Integrator
ISO	International Organization for Standardization
ITS	Informations Teknisk Standardisering
ITU	International Telecommunications Union
M	Mandatory
MASCOT	Modular Approach to Software Construction, Operation and Test
NC	National Committee
NR	Not Recommended
PFD	Probability of Failure on Demand
PM	Project Manager
QA	Quality-Assurance
R	Recommended
RAMS	Reliability, Availability, Maintainability & Safety
RBC	Radio Block Center
ReX	Return of experience
RQM	Requirement Manager
RRF	Risk Reduction Factor
RTRT	Rational Test Real Time
SEK	Svensk Elstandard
SG	Survey Group
SIL	Safety Integrity Level
SIS	Swedish Standards Institute
SEEA	Software Error Effect Analysis
SGA18	Survey Group A18
SSF	Sveriges Standardiserings Förbund
STM	Specific Transmission Module
TC	Technical Committee
THD	Tolerable Hazard Rate
TST	Tester
V& V	Verification and Validation
VAL	Validation Engineer
VER	Verification Engineer
WG	Working Group

Clarification of roles

Supplier	Is in this study the software developer that is working with products being affected by the standard.
Experts	This are the persons that are working in international working groups, developing and maintaining standards. Included in this designation are also persons that have been in contact with this standard or its area of software development.
User	The meaning of user in this study, has the same meaning as the supplier. It is a company/person that is following the standard in their daily work.
Developer	For this standard, the developer and the user are the same.
Orderer	An orderer is in this study a company/person affected by the standard, but who is not following the standard in their process. The orderer is buying a product that is developed following this standard.

1 Introduction

Building a railway system must follow railway safety standards. In Europe these are the European EN501xx family developed by CENELEC (European Committee for Electro-technical Standardization), consisting of EN 50126, EN 50128, EN 50129 and EN 50155, as shown in Figure 1.1.

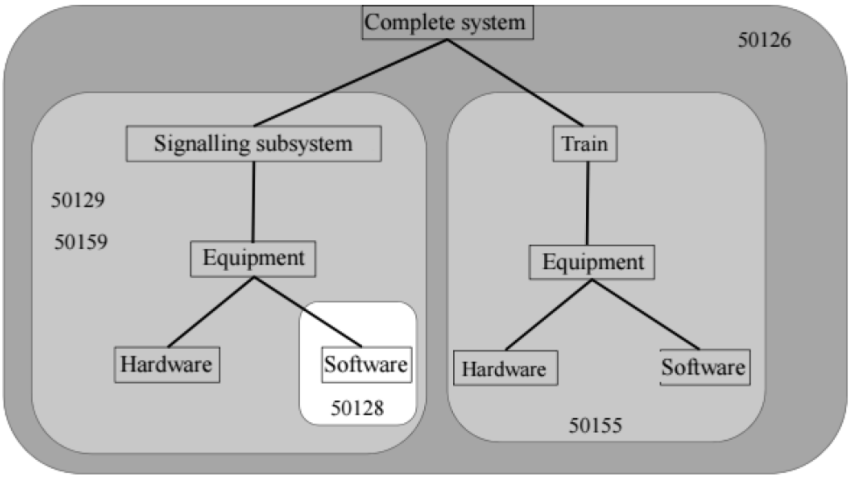


Figure 1.1: [1] Overview of the European EN501xx family

The standard EN 50128 Railway applications - Communication, signalling and processing systems is intended for software aspects, and specifies procedures and technical requirements for the development of programmable electronic systems which are used in railway control and protection applications [1]. EN 50128 is applied on all safety life cycles of electrical/electronic/programmable electronics systems which are used in safety of the system. The first version of this standard, EN 50128:2001, was released in 2001. Since the development of software used for safety applications in railway has been moving fast forward, the standard for the safety related software for railway systems has been updated in order to take into account new issues and problems that are not considered in the previous versions. The 2001 version has, since 2017-04-25, been replaced with the updated EN 50128:2011 (the 2001 version is still valid in Germany). This change in version could be quite extensive, depending on how proactive the affected companies have been, and will affect most of the parts of the whole process, from development and tests to organization and maintenance. The new version is a result of a maintenance work of the European committee CENELEC TC9XA, where the different national committees were requested to speak their opinion about the previous version EN 50128:2001. From these opinions and a several of meetings, the new version was formed and released in 2011.

1.1 Background

This subsection is intended to give the reader background information, and be a good background to the rest of this report.

1.1.1 The Swedish Transport Administration

"Everyone arrives smoothly, the green and safe way" [34]

This is the vision of The Swedish Transport Administration, which, in Swedish, is called "Trafikverket". A common Swedish abbreviation for Trafikverket is TRV, which is used below in this report. TRV is a government agency in Sweden, controlled by the Parliament and the Government of Sweden [36]. The headquarter is located in Borlänge, with regional offices from Malmö in the south, to Luleå in the north. TRV was established in 2010-04-01, when the operations of Banverket (Swedish Rail Administration), Vägverket (Swedish Road Administration) and SIKA (The State Institute for Communication Analysis) were merged into one common authority [36]. Furthermore, TRV have also took over part of the operations of Luftfartsverket (Civil Aviation Administration) and Sjöfartsverket (Swedish Maritime Administration).

TRV has around 6500 employees and is responsible of the longterm planning of the transport system in Sweden, including road traffic, rail traffic, shipping and aviation. The maintenance of the railways and roads is also included in the responsibility of TRV. The assignments for TRV is defined as [36]:

- TRV shall be responsible for the long-term infrastructure planning of road traffic, rail traffic, shipping and aviation, as well as for the construction and operation of state roads and railways;
- TRV shall work for a basic accessibility in interregional public transport;
- Based on a Civil and Environmental construction perspective, TRV shall create conditions for a socio-economically efficient, internationally competitive and long-term sustainable transport system;
- TRV shall work towards fulfilling the national transport political goals.

The organization of TRV is divided into three main categories, as shown in Figure 1.2. In the first category, the board, the General director and others, including the internal audit are located. The General director of TRV is Lena Erixon. In the second category, general functions such as economy, planning, communication, HR, IT etc. are located. In the third category, the business areas are represented, including Large Project, which, among others, includes the ERTMS project.

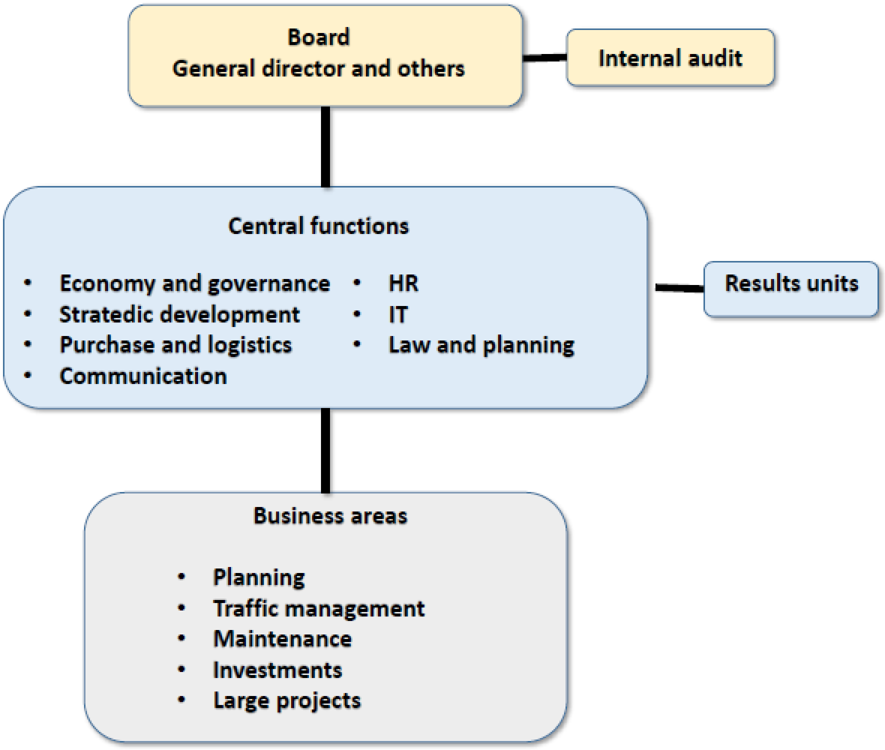


Figure 1.2: The structure of the organization of The Swedish Transport Administration.

TRV is a large authority responsible for the longterm planning of the infrastructure of the road traffic, railway traffic etc, in Sweden. They do not create any products within the company, which makes it important to have control over all of the different processes being used by their suppliers. A concern that has been raised from researchers, is the fact that TRV is commissioned to let private companies handle many services. This is even the case for complex technical services, such as the products affected by the version update of EN 50128.

A comparison can be made to IKEA. IKEA has the whole Supplier-chain within the company, which makes it much easier to control all parts of the process and not waste resources. As a result of the fact that TRV does not have any suppliers within the organization and is dealing with safety related complex technical services. It is important to guarantee that the whole chain is fully controlled.

1.1.2 ERTMS

The existing signalling system in Sweden is dated and in a need of an update. ERTMS (European Rail Traffic Management System) is a EU common, specified signalling system for railways. TRV is responsible for the introduction in Sweden and [35] ERTMS is a part of 'Large Projects' in their organization (as shown in Figure 3.1). It is financed with 30 billion SEK and is planned to be finished in 2035.

The purpose of implementing ERTMS is multi-fold. Firstly, ERTMS is a EU directive. I.e. all new railway tracks are required to be equipped with ERTMS, with the goal of finally have the same signalling system throughout the EU. The traveling over borders will be easier, in comparison with the existing signalling system. Secondly, the new signalling system is required in order to handle the new high speed tracks, that are implemented in Sweden. In the old (existing) signalling system ATC (Automatic Train Control), optical lightning is used for communication with the driver. With ERTMS (level 2 and 3) the optical signalling is changed to cabin signalling. The driver gets the messages to the monitor inside the vehicle. GSM-R, a GSM-based radio system using frequencies specifically reserved for railway applications, is used for communication. The real time signalling to the cabin will make the system safer, and without the optical signals, the system will be cheaper to maintain [34].

There are three different levels that defines ERTMS, level 1, 2 and 3. Level 2 and 3 involves continuous supervision of train movement with continuous communication, provided by GSM-R. The main differences between level 2 and 3 is that for level 3, there is no need for lineside signals or train detection systems on the trackside other than Eurobalises [41]. For level 2, lineside signals are optional and the train detection is performed by the trackside equipment. Level 1 has a continuous supervision of train movement and a non-continuous communication between vehicle and trackside. Lineside signals are used in level 1 [41].

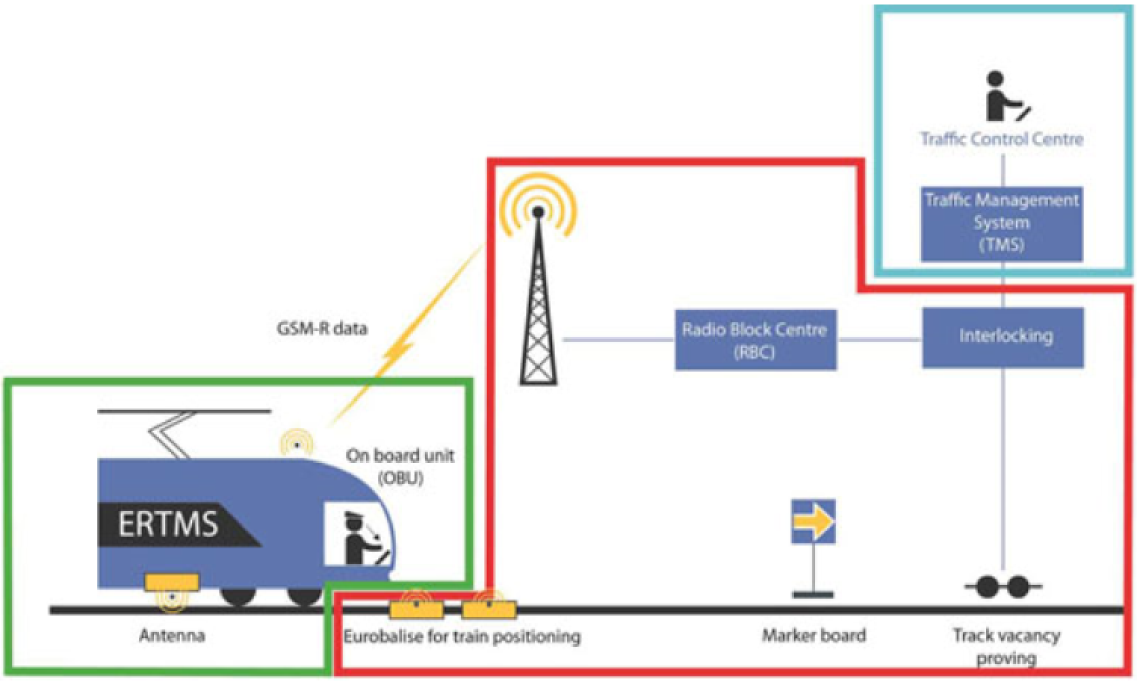


Figure 1.3: ERTMS, divided into three categories, ERTMS onboard (green), ERTMS trackside (red) and traffic management system (blue).

In Sweden level two (L2) and regional is implemented. Regional is a simplified version of Level 3 (L3), intended to be used on low density lines. Train movements are monitored continually by the radio block center (RBC) using this track-side-derived information. The movement authority is transmitted to the vehicle continuously via GSM-R together with speed information and route data [38].

ERTMS can be divided in three categories (Figure 1.3).

- ERTMS onboard - The onboard system consists of the signalling equipment placed on the vehicle.
  - BTM (Balise Transmission Module) activates the Eurobalise located on the railway track, and then receives the message from the Eurobalise. It then transmits the message to the onboard system.
  - The information panel on the drivers seat gives the driver all required information, such as position, speed, delays etc. The information panel is built up by the EVC (European Vital Computer) and can communicate with the EVC through the DMI (Driver Machine Interface). The main focus of the on-board computer EVC is to process: [11]
    - information received from the wayside equipment;
    - data introduced by the driver;
    - data coming from on-board sensors.
  - STM (Specific Transmission Module) is needed in order for the vehicle to communicate with the old/existing ATC balises and thus be able to traffic non-ERTMS equipped lines.
  - ETD (End-of-train device) is placed on the last carriage. It helps the driver to know the position of the last carriage, which is very useful when driving very long trains. Furthermore, it is also intended to inform the driver if the last carriage is lost.
- The ERTMS trackside system in the red frame in Figure 1.3, consists of the following parts:
  - Eurobalise is a small device placed on the railway track. The purpose is to transmit information to the vehicle and further to the Traffic Management System, regarding the position and other required information. The Eurobalise transmit information through the BTM, placed under the train. In order to keep track of the direction of movement of the carriage, the Eurobalises are placed in pairs. They are market '1' and '2', this way the trains knows whether the direction is nominal (1→2) or reversed direction (2→1).
  - Stop Marker boards are installed along the railway track at each end of block section. They are informing the driver of the exact location to stop the train in case it is requested. The marker board is blue with a yellow arrow, with a white border, and has a reflecting panel, in order to be visible from a far distance [11].
  - RBC (Radio block center) is the management of the movements authorities for all trains within a specific area. It is also storage, management and transmission of selected trackside data [13]. The train is continuously sending data to it, in order to report the exact position and direction [40].
- The traffic management system controls the whole system consisting of a traffic control center, a disturbance control center, a traffic control center etc.

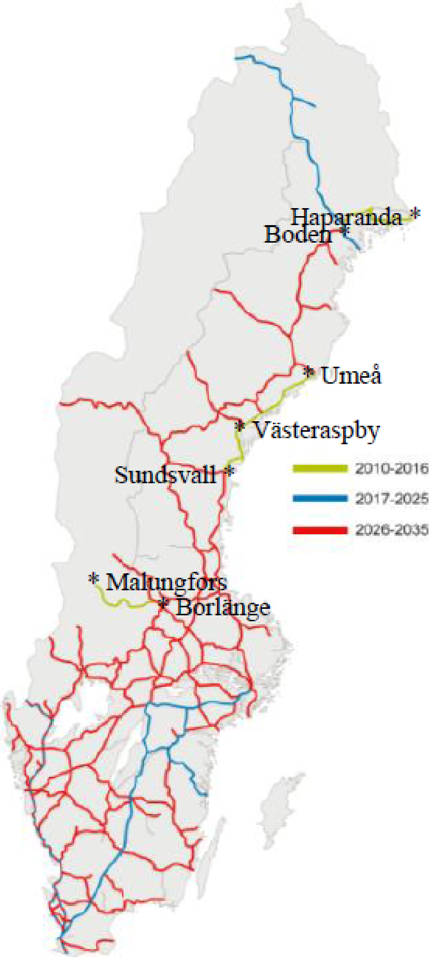


Figure 1.4: [34] Planned and executed introduction of ERTMS on the Swedish railway.

Up to now, four railway tracks have been implemented with the new signalling system:

- Botniabanan, 190 km, Level 2, Umeå-Västeraspby
- Västerdalsbanan, 140 km, Regional, Borlänge-Malungsfors
- Ådalsbanan, 130 km, Level 2, part route: Sundsvall-Västeraspby
- Haparandabanan, 160 km, Level 2, Boden-Haparanda

1.1.3 Humans and change

"Imagine life without change. It would be static... boring... and dull. When people feel stuck and frustrated, it is often their fear of change that is causing the problem"[33]

Without changes, no development happens [5]. Humans are not comfortable with changing processes. Different cultures are reacting to the changes differently, but overall it is a difficult process. Humans takes the same way to work, are eating the same food and are watching the same shows on TV. When a change happens in an organization, it affects the persons working there as well [5].

When a change of a standard is done, the daily work of many persons might be changed. The 2001 version of EN 50128 was followed for over 10 years, so when an extensive updated version is introduced to the process, this could be difficult for many persons in the organization, who are used to do things in a specific way. Tõive Kivikas [35] mentions in his blog that, he never uses the word 'change'. He instead uses 'development' or 'improvement', since these words indicate that the process is positive. You do not perform a change in order for something to become worse, you do it to improve the process.

Researchers in the area of humans and change are talking about different kinds of fears of change. A person that is afraid of a change will probably enter more than one of these stages, in a change process. Different kinds of fears of change can be:[33]

1. Fear of the Unknown - It can be scary to not know the outcome of a change. Many persons are staying at the same job their whole life, not because they are loving it, because of the fact that it is to scary to try something else.
2. Fear of Failure - You will never get everything in the way you expect it, right away. Therefore, the process of change has to start as soon as possible, in order to not be rushed in the end.
3. Fear of Commitment - This fear is why people do not set firm goals or accomplish what they set out to do. It is also this fear that might stop orderer of a product to get involved in the process of the supplier. The solution to this is that persons are not getting involved in anything, 'always stay on the safe side'. This lack of participation could lead to increased costs, no voice in important questions etc.
4. Fear of Disapproval - As a change are made, there might be a group of people disapproving to the changes. These are always a group of people that will disapprove changes, regardless what the changes are.
5. Fear of Success - This fear is most common on a personal level. "If you are successful, are people going to like you?"

When a standard is produced, the main issue is the theoretical aspect, as all needed requirements have to be included. The second part is the meaning of the requirements. This is many times forgotten. The meaning might be seen as obvious, and is therefore not included to the majority of the requirements. Last but not least, operation, *how and when should the requirement or method-/technique be implemented to the process?* There should be a balance between theory, meaning and operation in order for a standard to be followed as it should. This widely used triangle is explained in figure 1.5. Tõive Kivikas [35] is also pointing out the importance of motivations. In order to motivate others, the same arguments that you did get motivated by, should be used. The faster a change is done, the better for the affected.

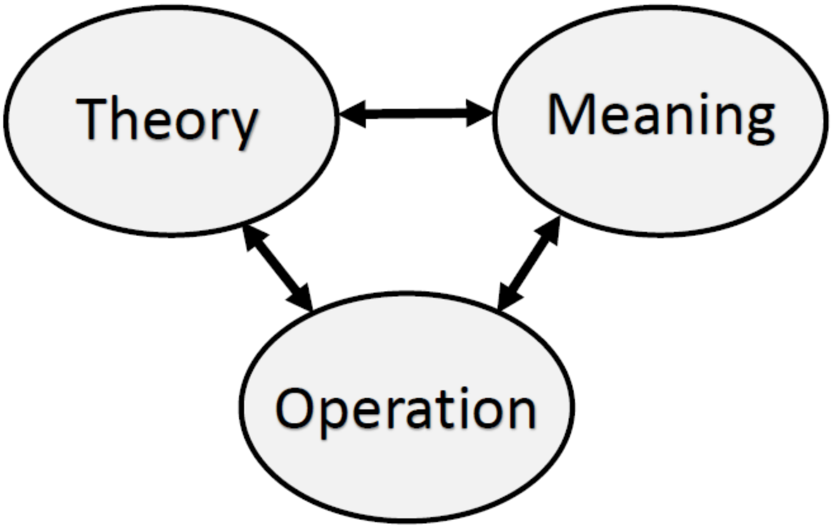


Figure 1.5: Meaning, theory and operation should all be clear in order for a standard to be used as it should.

When a standard is produced, the operation of it is obviously thought about during the standardization process. The question is, therefore, *why is there no plan for operation included in EN 50128?*

The purpose of the 2011 version of EN 50128 is clear, it is to better fit the modern and complex software, but *what is the meaning of each and everyone of the requirements? how and when are they supposed to be implemented?* The triangular relation between Theory, Operation and Meaning described in Figure 1.5 should be kept in mind as a standard is produced. A change process is, as mentioned earlier, scary to the majority of the humans. Many persons also connect a change to a lost of something. This makes it important to motivate why, when and how a change, or better expressed as an improvement, should be done.

It is important to prepare for a changing process and understand the advantages of the change in relation to stay in the current situation. It is important to get the right amount of information and not rush through the changes. Additionally, to be surrounded with positive energy is important. It is easier to be motivated to perform a change that is invented by the company, in order to become more successful. An updated version of a standard is a forced change process, which makes it even more important to make sure that the meaning and the operation is included in the standard.

1.1.4 Updating of the European standard EN 50128

For companies that are using this standard, the change in their working methods because of an updated standard can, as said above, be an extensive work, both regarding time and costs. Some parts are less extensive than others and less hard to change than others. The standard is also intended to be followed by different products, in different ways, which has resulted in an interpretable content. As the standard does have many parts that are interpretable, 'following the standard' might result in different approaches, for the same type of product. Many parts in this standard can also be removed from the process, even if that is not the purpose of the standard, if the cause is good enough. This means that either a person with good argument skills, or an Assessor with not enough knowledge, or an Assessor with a 'friendly' relationship with the personnel of the process to be assessed, could skip important parts of the standard. *Is this the meaning of the standard? Or is this a mistake in this standard?* The interpretable part of this standard, will make the result of this study more interesting, since the work with the standard can be different in different processes, even within the same company.

Is there any simple, correct and time saving way of dealing with the change in standard for a product/process? No, the question can not be answered with one concrete answer for all processes/products. The process/product size, how far in time the process/product has been going on and the total affect on the safety it will have, are all examples of details to look into, in order to decide when and how to change the standard for the specific process/product.

1.1.5 Problem description

Due to the update from EN 50128:2001 to EN 50128:2011, TRV has requested this study of how this change will affect them, in terms of costs, time and safety. Since TRV is not directly affected by this update of EN 50128, the study will be focused on the suppliers of TRV, and how the update will affect them and their products and thus their deliveries to TRV. Furthermore, information reached during interviews with experts having experience in the area is also added.

1.2 Purpose

The purpose of this study is to investigate the results of the update of the European standard EN 50128, from EN 50128:2001 to EN 50128:2011. *How has this EN 50128 update affected the companies using this standard?* As this extensive update in standard is done, it is important and interesting to see how much time and costs the update has caused the companies. It is also interesting to document the experience of this update. The companies might not find the update as extensive as it seem to be, or the other way around. The most affected parts of the process, from the companies point of view will be documented. *Is the standard easy to understand and is it following the modern and complex track as it is intended to do?* It is interesting to see how far the standard can be argued not to be used, and approved by the Assessor. This means that the standard is 'followed without being followed'. If the standard can be interpreted in to many ways, it might not be safe enough.

1.3 Issues

In order for the problem to be as clear as possible, some part problems are created. These part-problems will make it easier to reach the best possible result:

- Where is the standard used and who are mostly affected by the change of this European Standard?
- What are the possible ways of interpreting the 2011 version? In other words, could the standard be interpreted in different ways?
- How long time will it take for the affected companies to implement it and how much will it cost?

1.4 Conditions

This master thesis comprises 30 credits, which is about 20 weeks of full time studies. The course name is 'Degree Project in Electrical Engineering' and the base for the studies is at Uppsala Universitet. The study is performed by the writer, at TRV located in Solna, Sweden. The study has also been taking place at the location of the suppliers and during the SGA18 meetings (See section 3.2.1.1).

The work is based on the documents:

Document	Edition	Title
EN 50128	2001	Railway applications – Communications, signalling and processing systems – Software for railway control and protection systems
EN 50128	2011	Railway applications – Communications, signalling and processing systems – Software for railway control and protection systems

1.5 Outline

This report starts with an introduction in order for the reader to get a picture of the subject and get information of the report structure, issues, conditions and purpose of this master thesis work. In chapter two the working methods and techniques are presented. Ethical considerations and source criticism are also presented in this chapter. Since this master thesis is not intended to be an audit of the work with a product for a company, but is following a working process of a sometimes sensitive subject, the ethical considerations and the source criticism is important to take into account. Chapter three is the theory part. In this chapter all background informations connected to the subject work is presented, including information about TRV, standards e.g. CENELEC, EN 50128, Safety Integrity Level (SIL) and other relevant information required for the reader to fully understand the result, which will be presented in chapter four.

The base of the result is connected to the experience and documentations from the suppliers work with this update in standard for safety related software. The result will also include detailed information regarding particularly affected parts by the update. In the Appendices, tables are included covering the result.

In chapter five, an analysis of the result is done and in chapter six a discussion of the results is presented. In the discussion the working methods and the techniques used on this subject are discussed. The report will end in chapter seven, where a conclusion is presented together with future research suggestions and an evaluation of the study.

2 Methodology

This section deals with a description of the methods and techniques used in this master thesis work, reasons why these methods and techniques were selected and how these methods and techniques were used.

2.1 The working process

The method used for this thesis project is a combination of Qualitative research and Quantitative research.

- Qualitative research - In this method, phrases like 'why' and 'how' are widely used [30]. These phrases are intended to be more open than the phrases used in the quantitative research. This method is used in order for the person being interviewed to be more open, to be able to share memories and experiences. Questions used in this research technique is intended to open up for additional information from the person being interviewed. The answers should include memories and thoughts. The time of the interviews using this method is hard to estimate in beforehand, it is therefore important that a comfortable atmosphere is applied to the interviews.

- Quantitative research - In this method, phrases like 'what', 'where' and 'when' is widely used [30]. Survey researches with a closed amount of alternative of answers are a common form of data collection. The questions asked are applied in order to be answered with 'yes', 'no', a number etc. There is no space for thoughts and memories in these specific questions. Qualitative research is widely used in statistics, economics, mathematics etc. intended to present the results with graphs etc.

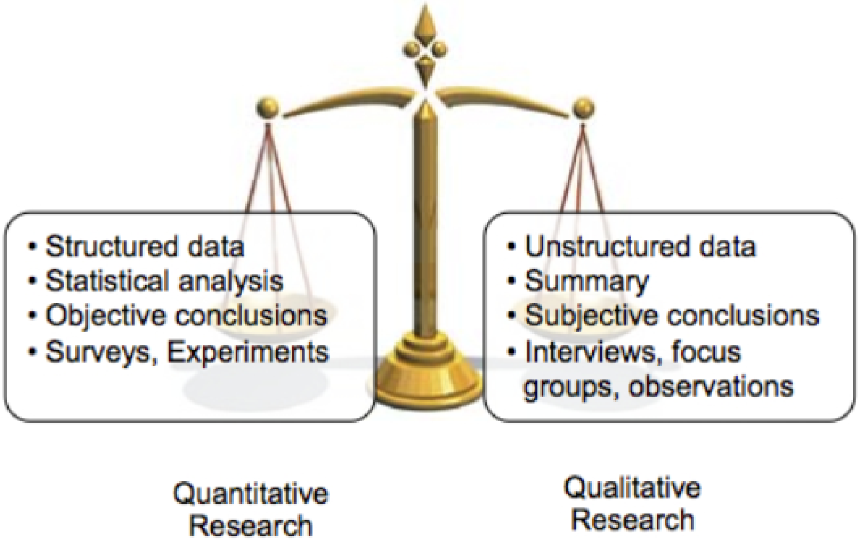


Figure 2.1: [29] Qualitative Research vs Quantitative Research.

As qualitative research is used in the phase where analyses are produced, the researchers needs to use their intellectual and creative capability [3]. I.e. the qualitative part of the work does mainly come to its purpose when the result should be analyzed, as opposed to the quantitative part of the process, which mainly results in concrete mathematical results. Because of this, both of these methods are important to this study, since experiences and thoughts in the analysis is as important as the tabulated data. The tabulated data/information presented in the Appendix B (section 8.2) are both created from the qualitative and the quantitative methods, while the tables where the methods are presented, are created from closed structured questions and the connected requirements are created from open experience based questions, see Appendix A, section 8.1.

This combination of research methods is called method triangulation or mixed methods. By this combination of methods, it is easier to get all interesting information in a specific area. With the mixed method, the technique of using one method in order to get ideas and questions for the other method is applied [2]. Since the subject of this master thesis has been clear, but where and what kind of results to find has been unclear, the mixed method has been a perfect approach, in order to create clear and specific question from an unclear base. A lot of questions has therefore been sent to the supplier after the official qualitative interviews.

The result and the analyses has been produced around the collected data, the experiences, reflections and memories explained in the interviews and the reflections and conclusions from the writer. The answers on questions asked and comments from the interviews has been used as "coding" for the result and analyses. This coding-technique makes it easier to start and to cover all interesting parts of the process. It will also give the reader a better insight to the background of the result.

Snowball selection is used to find the correct persons with the specific information needed. In this selection method, a small non-random sample of people who have the relevant knowledge in the subject are chosen. This selection of people is used in order to find more respondents [4]. This is an excellent way of finding the persons with the required knowledge in a specific area, especially when the researcher is new to the subject.

The mixed method and the in depth interviews (explained in section 2.1.2) are combined with the snowball selection method in order to find the respondents with the specific knowledge. This method was the most efficient to apply on this subject, because of the complexity, which led to a more limited amount of people in the area. Persons with the specific knowledge needed for this master thesis, would be very hard to find if this snowball selection technique was not applied.

As the work proceeds and the collected data, experiences and theories are collected, there will be some questions coming up. These questions might come as a performance anxiety, but should not be thrown away. These questions should instead be thought about, and might even lead the result and analyses to be more suitable and interesting for the subject. These question could for example be:

- What is it that is important in the empiricism?[3]
- What is it that is interesting in connection to the area that is set for the master thesis work?
- Has there been coming up any new interesting topics that should be interesting to include in the work?
- Is the problem still relevant formulated after a lot of data and information is collected?[3]

These questions will especially come up in a work when the outcome is unsure, as for this master thesis work. The study has to start somewhere and might not end as expected. These questions is important to take into account, changes in the plan might make a better result and analysis. This has been the case for this master thesis work. I.e. new interesting topics has been added and focuses has been changed along the way. This has made the outcome of this master thesis work more interesting and relevant.

2.1.1 Literature study

Search engines has been used in order to find interesting information, literature and persons to contact in the subject. The search for literature on the Internet began with searches of phrases or sentences which were relevant based on study research questions or labor issues. Most of the searches included the name of this European standard, 'EN 50128'. As relevant documents and literature was found from these searches, the authors of interesting information was contacted, in order to collect more information from these persons. Generally, this resulted in additional interesting and valuable information from these writers.

The literature study was performed in order to get an as wide and deep knowledge in the subject as possible. Since not much is published or performed in this subject, the literature has mainly been studied in order to build a knowledge in the subject, and with this deep knowledge be able to easier create interesting interviews and understand the answers from these interviews. If the answer is clearly understood, interesting supplementary questions can be asked to the subject. The literature is mainly books, and documents written on a neutral basis, not written intended to sell a product or in other ways share information that is in the writers interest to angle the information in the advantages of the writer.

2.1.2 Interviews

The suppliers and other stakeholders in contact with this European standard have been interviewed. Most of the interviews were held ”face-to-face”, not over the phone. This was due to the fact that in many of the questions asked, experiences and thoughts were expected, which could be harder to share over phone [4]. Other aspects of why the interviews where held ”face-to face” is that in these situations, the facial expressions can be seen and interpreted. Furthermore, a personal contact is established easier in person than over the phone.

The interviews were mainly structured in order to cover every part of this standard. Where it felt relevant, unplanned and planned additional questions were asked in order to get more details in the areas where the result was more complex. A good atmosphere is preferable for an in depth interview. There should be time and space in the interviews, assigned for the supplier to explain deeply when they have interesting things to add. Feelings and thoughts about the process were both planned as questions, but also improvised where it felt appropriate. Therefore, an in depth interview was the best technique for this purpose, since it was hard to predict what kind of information the suppliers intended to share. The thought behind in depth interviewing is to create a situation for a quite free conversation around questions/subjects that the interviewer has decided in forehand. It is important to remember that everybody does not remember everything as it truly was [4], so for a study that needs the facts to be true, documents are better to study in order to get only true facts. In this study, the experiences plays an important part of the process, so even if the memories are not truly correct, it is how the person being interviewed experienced it. This interviewing technique uses open questions, as opposed to survey research, which uses the technique of closed questions [3]. Closed questions have a closed selection of answers, which does not allow the person being interviewed to share additional information, memories or experiences. Survey research is therefore more sufficient for projects that are focused on a large population, in the purpose of find a specific thought connected to, e.g. a specific age group. Since this study is based on a limited amount of stakeholders and it was hard to predict the outcome of the result, an open research technique for interviews was the best choice for this project.

The interviews had an open end, as the intention was to add new questions depending on the answers from the person being interviewed. These new questions were raised both during the actual initial interview, but also afterwards when the original answers were analyzed. This was an efficient way of working with this extensive study, since without the opportunity of asking additional questions, the result would lack in information.

2.1.3 Data collection

The data collection was basically based on literature study and document research as well as interviews with relevant suppliers and stakeholders. Since there are not many studies done in this subject, the work is basically made with primary data. By primary data, it means that the data is collected by the author and not by somebody else. The opposite to primary data is secondary data. The primary data is collected over time from interviews and document research. For this kind of studies, when not much information is written in the area, all data needed is impossible to collect at one time. During analyses of the data, more data might be necessary to collect.

The data that is collected is connected to the requirements and the meaning of this European standard. Since it was hard to know what the result from the interviews would be in beforehand, more additional questions had to be asked later, as the result from the interviews was connected to the requirements of this standard. Additional information from experienced persons had to be added along the way, when questions were coming up. This technique for data collection can be compared with the snowball selection method. The data collection starts at one point and is as the project is proceeded, added with new questions and sources for the data collection. By using this technique, no interesting information goes missing.

If only interviews were used as the source for information, the result would only be based on memories and personal experiences. Because of that, it is important that the data collection is based on a mixture of document studies and interviews, as the questions can be asked based on what is documented. It is also important that experiences and memories are collected from as many different parts as possible, in order to make the study as neutral as possible.

In order to get as much information of the standardization process as possible, courses in the subject were taken. These courses were intended to expand the knowledge on the subject in order to, in a better way understand, analyze ans assess the data and results. In the international working group SGA18, experiences have been shared and contacts connected, intended for additional questions and future work in this subject. This international working group is a collection of experts and users of this standard (see section 3.2.1.1). Meetings have been set up for the purpose of creating an audit on the 2011 version of this standard, similar to this study in the present master thesis project. The work in SGA18 has also been focusing on how other, new versions of related standards could be/have been affecting EN 50128:2011. Two physical meetings will be attended during the time of this master thesis work, the first in Florence, Italy and the second in Copenhagen, Denmark. The work of SGA18 will include two more meetings and then be finished in autumn 2017.

2.2 Ethical considerations

When a study is performed, it is important that involved persons and companies are clear with what the intension of the study is. Before an interview with a stakeholder is started, information and intension of the study are therefore shared. The information requirements are followed by the author in this master thesis work, which states that, all involved in the study have to be fully informed of the purpose of the study. The involved parties also have to be informed of the structure of the study [4].

Since this master thesis is not intended to be an audit of a specific product or company, names of products and companies are chosen to be kept hidden. The products presented is named with a number and the companies are called suppliers, since this master thesis is written for TRV who is affected by this standard via the products of their suppliers. Comments is presented and discussed, but the sources are not disclosed.

2.3 Source criticism

The study is based on interviews and document studies. Background information regarding the subject has been studied in books, Internet sites and in older studies that have been made on similar subjects. The purpose has been to be as neutral as possible. This has been challenging in some parts, since many of the sites and documents written on the subject has been produced by companies. As the companies has the attention of selling a product, the information has been studied carefully. Therefore, the background information has mostly been studied from neutral writers. Sites like Wikipedia has also been carefully examined, as the writers of this information are not always found to be neutral. A Bryman describes four important questions in the book 'Samhällsvetenskapliga metoder', to take into account as these Internet sites are examined. These questions has been taking into account when the sites have been used as material for this study. These evaluation criteria questions are described in the book as:[4]

- Authenticity - Is the material real and from a unambiguously background?
- Credibility - Is the material free from distortions and inaccuracies?

- Representativeness - Is the material typical to the subject? If not, do you know in what extent the material is not typical to the subject?
- Meaningfulness - Is the material clear and easy to understand?

Internet sites are avoided as sources as long as possible, due to the fact that new sites come out and old sites are removed. If Internet sites that are used as sources are removed as the study is read, this does lower the credibility to the author(s) of the study. The date visiting the Internet site needds to be given in its reference to in the reference list, since the site can be changed after the site has been examined [4].

The result is based on information from companies that are affected by this change in standard. These companies could possibly keep information that could affect the result negatively. The result is based on the assumptions that this is not the case, and that all questions asked are truly answered.

3 Theory

In this section, information is presented, that is considered to be important in order for the reader to easier understand the content of the result and the analysis.

3.1 Standards

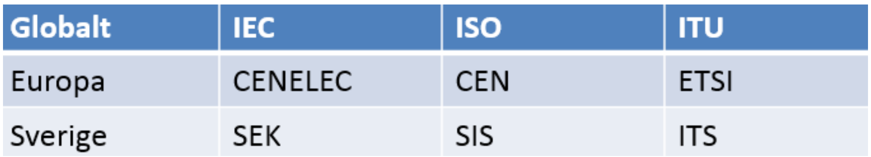
A standard is a norm that applies to all aspects of a matter. It is documented knowledge from prominent players in the industry, business and society [20]. The documented knowledge is mostly recurrent problems, and the standard a documented set of rules and requirements. A standard is also great when you want to create order or to determine the requirements that products and services can be measured against [22]. There are standards about everything from medicine to furnitures as well as quality management systems for companies. Standards aims improved safety, increased trade, reduced costs, secure interrupt, and environmental and consumer protection [20]. They are in many cases voluntary to use, but in some cases the government regulations are cited to standards.

The developer of a standard has the intension to make affected products/processes safer, but the standard itself can not make the process/product safer. By following a specific standard for the process/product, other involved parts/stakeholders of the process can understand that the process or the product produced. If all actors in an industry are using the same standard, the process/products are understood in terms of safety etc. by all actors in that area. This way the standard can affect the safety in a positive way. As a standard is used on a software, the software has the responsible of the safety, not the standard. The standard is used in order for the software to e.g. be developed and tested, in a controlled way.

There are different types of standards [22]:

- Basic Standard - a standard that have great coverage and contain general ground rules for a specific area;
- Terminology Standard - a standard that handles terms, usually with the term definitions, and sometimes by explanatory text, illustrations, examples, etc;
- Test Standard - a standard related to test methods, sometimes with additional and appropriate ground rules for testing, e.g. selection, use of statistical methods, the sequence of the test etc;
- Product Standard - a standard that specifies requirements to meet for a product or group of products, to ensure that they fulfill their purpose;
- Process Standard - a standard that specifies requirements to meet for a process to ensure that the process meets its purpose;
- Service Standard - a standard that specifies requirements to fulfill a service, to ensure that the service meets its purpose;
- Interface Standard - a standard that specifies requirements for the interoperability of products and systems via their interface;
- Standard for data to provide - a standard that contains a list of characteristics, that is, values or other data, to be specified for the product, process or service.

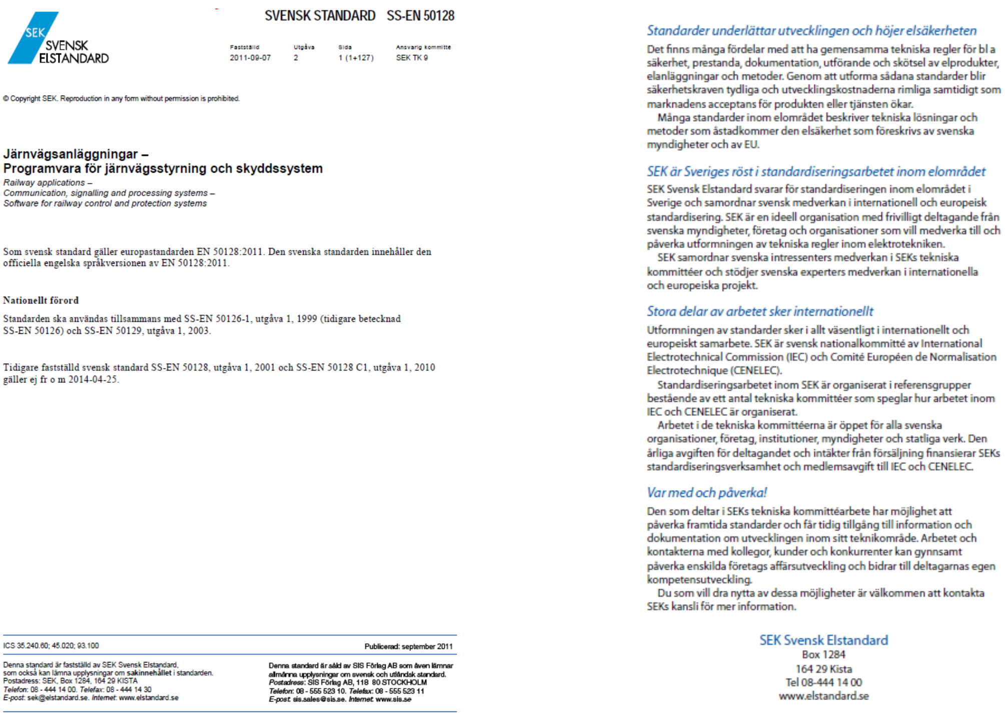
Standardization is the creation of joint conventions or ways to work for it to work smoothly with other parts. Sweden participates in international standardization through the three, by the state recognized standard bodies, SIS (Swedish Standards Institute), SEK (Svensk Elstandard) and ITS (Svenska Informations- och Telekommunikationsstandardiseringen). SIS is a member of the international standardization organizations CEN (European Committee for Standardization) and ISO (International Organization for Standardization), SEK is a member of CENELEC (the European Committee for Electrotechnical Standardization) and IEC (International Electrotechnical Commission), and ITS is a member of ETSI (European Telecommunications Standards Institute) and ITU (International Telecommunication Union) (figure 3.1). Usually a standardization project is financed by the companies, organizations, authorities and other stakeholders that are contributing and in order to create/improve a certain standard [21].



Globalt	IEC	ISO	ITU
Europa	CENELEC	CEN	ETSI
Sverige	SEK	SIS	ITS

Figure 3.1: [42] Global, Europe and Swedish standardization system.

Depending on where the standard is created and by who, it is named differently. European standards fixed by some of the European standards organizations CEN, CENELEC and ETSI [24] (figure 3.1) is named with EN and numbers between 50 001 and 59 999, e.g. EN 50128. An international standard fixed by any of the global standards organizations ISO, IEC or ITU (figure 3.1) is named with IEC or ISO. IEC comes with a number over 60 000. If the international standard then becomes a European standard, the number is kept, but IEC is changed to EN. EN gets, when connected to a Swedish standard, SS as a prefix. Similarly happens in other countries. An example of this is SS-EN 50128 [23].



SVENSK STANDARD SS-EN 50128

Fastställd	Utgåva	Sida	Ansvarig kommitté
2011-09-07	2	1 (1+127)	SEK TK 9

© Copyright SEK. Reproduction in any form without permission is prohibited.

Järnvägsanläggningar –
Programvara för järnvägsstyrning och skyddssystem
Railway applications –
Communication, signalling and processing systems –
Software for railway control and protection systems

Som svensk standard gäller europastandarden EN 50128:2011. Den svenska standarden innehåller den officiella engelska språkversionen av EN 50128:2011.

Nationellt förord

Standarden ska användas tillsammans med SS-EN 50126-1, utgåva 1, 1999 (tidigare betecknad SS-EN 50126) och SS-EN 50129, utgåva 1, 2003.

Tidigare fastställd svensk standard SS-EN 50128, utgåva 1, 2001 och SS-EN 50128 C1, utgåva 1, 2010 gäller ej fr o m 2014-04-25.

ICS 35.240.60; 45.020; 93.100 Publicerad: september 2011

Denna standard är fastställd av SEK Svensk Elstandard,
som också kan lämna upplysningar om **sakinnehållet** i standarden.
Postadress: SEK, Box 1284, 164 29 KISTA
Telefon: 08 - 444 14 00. *Telefax*: 08 - 444 14 30
E-post: sek@elstandard.se. *Internet*: www.elstandard.se

Denna standard är såld av SIS Förlag AB som även lämnar allmänna upplysningar om svensk och utländsk standard.
Postadress: SIS Förlag AB, 118 80 STOCKHOLM
Telefon: 08 - 555 523 10. *Telefax*: 08 - 555 523 11
E-post: sis.sales@sis.se. *Internet*: www.sis.se

Standarder underlättar utvecklingen och höjer elsäkerheten

Det finns många fördelar med att ha gemensamma tekniska regler för bl a säkerhet, prestanda, dokumentation, utförande och skötsel av elprodukter, elanläggningar och metoder. Genom att utforma sådana standarder blir säkerhetskraven tydliga och utvecklingskostnaderna rimliga samtidigt som marknadens acceptans för produkten eller tjänsten ökar.

Många standarder inom elområdet beskriver tekniska lösningar och metoder som åstadkommer den elsäkerhet som föreskrivs av svenska myndigheter och av EU.

SEK är Sveriges röst i standardiseringsarbetet inom elområdet

SEK Svensk Elstandard svarar för standardiseringen inom elområdet i Sverige och samordnar svensk medverkan i internationell och europeisk standardisering. SEK är en ideell organisation med frivilligt deltagande från svenska myndigheter, företag och organisationer som vill medverka till och påverka utformningen av tekniska regler inom elektrotekniken.

SEK samordnar svenska intressenters medverkan i SEKs tekniska kommittéer och stödjer svenska experters medverkan i internationella och europeiska projekt.

Stora delar av arbetet sker internationellt

Utformningen av standarder sker i allt väsentligt i internationellt och europeiskt samarbete. SEK är svensk nationalkommitté av International Electrotechnical Commission (IEC) och Comité Européen de Normalisation Electrotechnique (CENELEC).

Standardiseringsarbetet inom SEK är organiserat i referensgrupper bestående av ett antal tekniska kommittéer som speglar hur arbetet inom IEC och CENELEC är organiserat.

Arbetet i de tekniska kommittéerna är öppet för alla svenska organisationer, företag, institutioner, myndigheter och statliga verk. Den årliga avgiften för deltagandet och intäkter från försäljning finansierar SEKs standardiseringsverksamhet och medlemsavgift till IEC och CENELEC.

Var med och påverka!

Den som deltar i SEKs tekniska kommittéarbete har möjlighet att påverka framtida standarder och får tidig tillgång till information och dokumentation om utvecklingen inom sitt teknikområde. Arbetet och kontakterna med kollegor, kunder och konkurrenter kan gynnsamt påverka enskilda företags affärsutveckling och bidrar till deltagarnas egen kompetensutveckling.

Du som vill dra nytta av dessa möjligheter är välkommen att kontakta SEKs kansli för mer information.

SEK Svensk Elstandard
Box 1284
164 29 Kista
Tel 08-444 14 00
www.elstandard.se

Figure 3.2: [15] The Swedish cover pages for the European standard EN 50128:2011.

SS stands for Swedish standard and is a standard adopted by one of Sweden's three national standards bodies SIS, SEK, or ITS [24]. A standard marked with SS does not mean that it is in Swedish. It is very rare that a standard is translated to Swedish, but a cover page has to be included in Swedish, see figure 3.2 If a translation is done, it is a possibility that translation mistakes are done. If possible, translations are avoided.

When a standard is about to be updated, it first has to be used for some time in order for the experts to see the result of the work with the specific standard. The process of maintenance and update the standard is then an extensive and time consuming work, a process that takes about five years (depending on the area and extent on the standard). During this time of maintenance, the standard will get old, which means that, as an updated version of a standard is released, the standard will already be dated (depending on the area). This has both positive and negative sides.

Methods/techniques in the standard can be said to be tested, as they have been used in the processes already. On the other hand, a standard that is experienced to be dated can be hard to motivate to the users.

Different profiles are involved in the standards working groups. Most of them are authorities, service providers and system integrators. Technology suppliers, academy persons, consultants and vendors are presented, but the end-users representatives are missing, which is seen as a big problem.

3.2 CENELEC

CENELEC - *European Committee for Electrotechnical Standardization* is responsible for the European standardization in the area of electrical engineering. CENELEC, ETSI and CEN (figure 3.1) are together covering the technical standardization in Europe. The contents of the standards are created by the international working groups. CENELEC is taking care of the printing, advertising and sales of the standard.

They were founded in 1973 as a merger from CENELCOM and CENEL [26], and are based in Brussels, Belgium, at the CEN-CENELEC Management Center (CCMC) [25]. The committee was founded in order to facilitate trade between countries. By creating European standards that all parties could follow, the products/processes were understood by them all. As a country accepted the European standard, they had to give up their national standard in the same area. A country can only follow a national or an international standard. Most of the European countries are at the moment members of CENELEC. CENELEC also has cooperation agreements with Canada, China, Japan. South Korea, Russia and informal agreement with the USA [25].

3.2.1 CENELEC TC 9X

Standardization of electrical and electronic systems, equipment and associated software for use in all railway applications, whether on vehicles or fixed installations, including urban transport [7].

This is the scope for TC 9X (Technical Committee 9X). TC 9X was set up as a new committee in 1989, in the area of electrical and electronic applications for railways. TC 9X works in close liaison with Technical Committee 9 (TC 9) of the International Electrotechnical Commission (IEC), Electrical equipment and systems for railways.

TC 9X is divided in three sub-Committees [7]:

- TC 9X SC 9XA - Standardization of railway Communication, signalling and processing systems, taking into account the relevant safety requirements;
- TC 9X SC 9XB - Standardization of electrical, electronic and electromechanical material on board rolling stock, including associated software;
- TC 9X SC 9XC - Standardization of - AC and DC supply lines, both overhead and third rail type, - ancillary circuits, - machinery and equipment of specific feature for traction in fixed plants, - installations and safety requirements in fixed plants.

TC 9XA are taking account the relevant safety requirements for the signalling systems. EN 50128 is under the responsibility of TC 9XA. The responsibility includes development, maintenance, changes, audits etc.

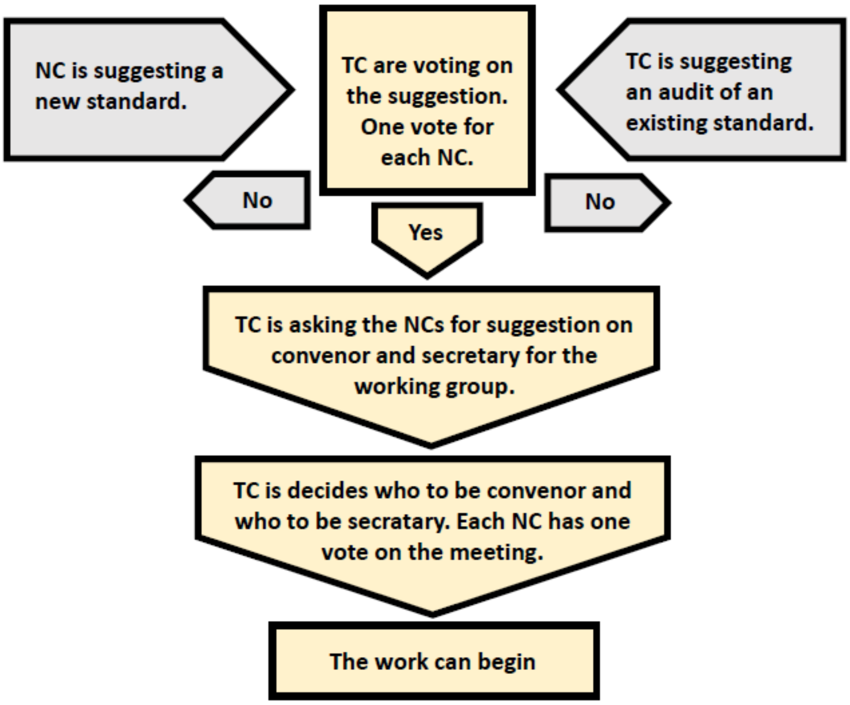

NC is suggesting a new standard.
TC are voting on the suggestion. One vote for each NC.
TC is suggesting an audit of an existing standard.
No
Yes
No
TC is asking the NCs for suggestion on convenor and secretary for the working group.
TC is decides who to be convenor and who to be secratary. Each NC has one vote on the meeting.
The work can begin

Figure 3.3: The standardization process, both for creating new standards, performing audits or maintaining an existing standards.

The work done by these committees are recorded in the database of CENELEC (section 3.2). The committee TC 9X are having a plenary meeting twice a year, in order to follow up on existing projects and plan new ones. In connection to these meetings, the sub-committees are having meetings for the same purpose, but also to follow up the decisions made at the TC 9X meetings.

As a new standard are suggested by a NC (National Committee), or a maintenance/audit work on an existing standard are suggested by TC, voting in TC will take place. Each NC has one vote. If the voting results with 'yes', TC will ask the National committees to suggest a convener and a secretary for the working group. TC is then making the decision through voting. Each NC has one vote. The work of the working group can then begin. The steps from suggestion to work, are illustrated in figure 3.3.

3.2.1.1 CENELEC SG A18

CENELEC SG A18 (Survey group A18) are a survey group created in order to perform an audit of EN 50128:2011. This audit are planned to take approximately six month and will result in a technical report in autumn 2017. The main subjects in this survey group is:

- Experience with EN 50128:2011 over the past 5 years;
- Relationship between EN 50128:2011 and the emerging EN 50126-1, EN 50126-2 and EN 50129;
- Impact and input of the emerging rolling stock standard EN 50657 to EN 50128:2011.

The Survey group will have four meetings and there is around 25 members of this group, from all of Europe.

3.2.2 EN 5012x

In the CENELEC EN5012x family (Figure 1.1), four standards are included, these are:

- EN 50126 - Railway Application - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS);

  - EN 50126-1 - Basic requirements and generic process;
  - EN 50126-2 - Guide to the application of EN 50126-1 for safety;
  - EN 50126-3 - Guide to the application of EN 50126-1 for rolling stock RAM;

- EN 50128 (section 3.2.2.1) - Railway Application - Communications, Signalling and Processing Systems - Software for railway Control and Protection systems;

- EN 50129 - Railway Application - Communications, Signalling and Processing Systems - Safety related electronic system for Signalling;
  - EN 50129-1 - Cross-Acceptance;
  - EN 50129-2 - Safety Assurance;

- EN 50121 - Railway applications - Electromagnetic compatibility
  - EN 50121-1 - General;
  - EN 50121-2 - Emission of the whole railway system to the outside world;
  - EN 50121-3 - Rolling stock - Train and complete vehicle;
  - EN 50121-4 - Emission and immunity of the signalling and telecommunications apparatus;
  - EN 50121-5 - Emission and immunity of fixed power supply installations and apparatus;

The cooperation of EN 50126, EN 50128 and EN 50129 are presented in Figure 1.1.

3.2.2.1 EN 50128

The European standard EN 50128 - *Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems*, specifies procedures and technical requirements for the development of programmable electronic systems, which are used in railway control and protection applications. It is applied on all safety lifecycles of electrical/electronic/programmable electronics systems, that are used in safety of the railway system.

The first version of EN 50128 was approved by CENELEC TC 9X on 2000-11-01, and released in mars 2001.

EUROPEAN STANDARD
NORME EUROPÉENNE
EUROPÄISCHE NORM

EN 50128

June 2011

ICS 35.240.60; 45.020; 93.100

Supersedes EN 50128:2001

English version

Railway applications -
Communication, signalling and processing systems -
Software for railway control and protection systems

Applications ferroviaires -
Systèmes de signalisation, de
télécommunication et de traitement -
Logiciels pour systèmes de commande et
de protection ferroviaire

Bahnanwendungen -
Telekommunikationstechnik,
Signaltechnik und
Datenverarbeitungssysteme -
Software für Eisenbahnsteuerungs- und
Überwachungssysteme

This European Standard was approved by CENELEC on 2011-04-25. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

CENELEC
European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Management Centre: Avenue Marnix 17, B - 1000 Brussels

© 2011 CENELEC - All rights of exploitation in any form and by any means reserved worldwide for CENELEC members.

Ref. No. EN 50128:2011 E

Figure 3.4: The cover page of the European standard EN 50128:2011.

The 2001 version was decided to expire 2014-04-25, three years after the release of the 2011 version. In the end of 2003, the German NC suggested that the 2001 version should be in use another three years [6]. The reason for this was the reconstruction of EN 50126. It would be to expensive and extensive for the affected companies, if updated versions of both EN 50128 and EN 50126 should be implemented at the same time. The postponement from the German committee are attached in Appendix E, section 8.5. The outcome was that the 2001 version of EN 50128 would be replaced by the 2011 version by 2017-04-25. The process of developing EN 50126 was stopped in 2014, since over 5600 comments came in about the standard [6]. The work with EN 50126 started again, but the planned integration with EN 50128 and EN 50129 was changed. The updated version of EN 50126 has still not been released, but the maintenance work is in the final phase. Today, a draft of EN 50126 is available. See figure 1.1 for an overview of the EN 5012x family.

The 2011 version of EN 50128 includes a double number of requirements, particularly 360 instead of 180, since the complexity and advantages of the software has increased. A standard that is handling software for railway control and protection systems needs to adapt to the development of the software. Methods/techniques required in the 2011 version are modernized after dated methods/techniques have been removed. The updated version might not be more suitable because of the increased amount of requirements, but is more consequent and includes a clearer setup with more intuitive explanations. A company that is in the forefront of software development and implementation is probably developing their way of working with methods/techniques, personnel etc., in order to have the safest system possible. If this is the case, a standard is used as a guideline and a support for the companies. These companies will not have to spend a lot of resources on an updating process.

In Figure 3.5 and 3.6, the structure of the two versions of EN 50128 is illustrated. The 2011 version includes a lot more restrictions, but it also has a clearer structure. During the maintenance work, the structure of the 2001 were criticized in many aspects. The structure was therefore the main focus in the maintenance work. Therefore, the group of requirements were reorganized and the main clauses were reduced in number. Some of the main clauses in the 2011 version are similar to the previous main clauses, and some are completely rearranged.

Clause in EN 50128:2001	Clause in EN 50128:2011
4. Objectives and conformance 5. Software safety integrity levels	4. Objectives, conformance and software safety integrity levels
6. Personnel and responsibilities 7. Lifecycle issues and documentation	5. Software management and organization
10. Software design and implementation 11. Software verification and testing 12. Software/Hardware integration 13. Software validation 14. Software assessment 15. Software quality assurance Parts of: 6, 8	6. Software assurance
8. Software requirement specification 9. Software architecture 10. Software design and implementation 11. Software verification and testing 12. Software/Hardware integration 13. Software validation Parts of: 7, 15	7. Generic software development
17. Software configured by application data	8. Development of application data or algorithms: systems configured by application data of algorithms
16. Software maintenance Part of: 13	9. Software deployment and maintenance

Appendix is extended in the 2011 version. Appendix A includes the tables of techniques/methods, in both of the versions. The content of Appendix B in the 2001 version are changed to Appendix D in the 2011 version. It includes 'Bibliographical detailing the techniques employed'. Appendix C is new and includes the 'Document control summary'. The 'Document control summary' was included in requirement 7.2.10 in the 2001 version, and was named 'Document Cross-Reference Table'. The content of Appendix B in the 2011 version is new to this standard and includes 'Key software roles and responsibilities'.

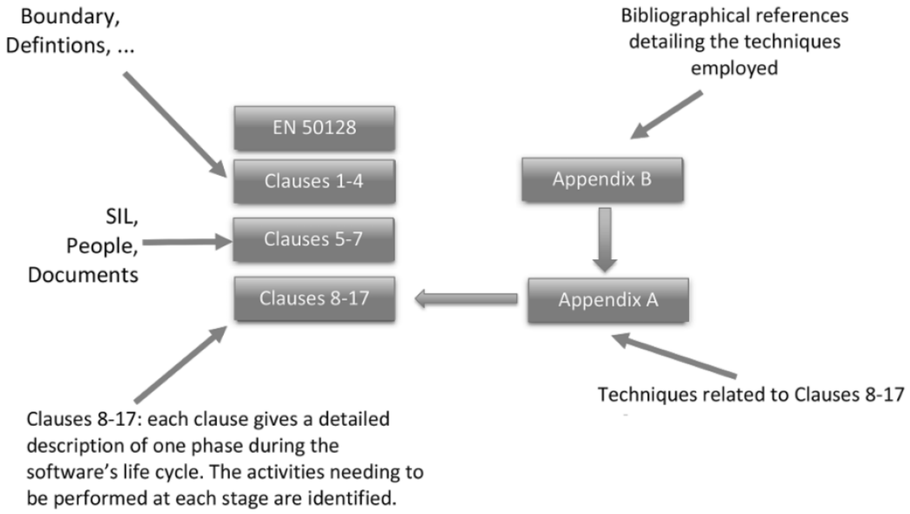

Boundary,
Defintions, ...
Bibliographical references detailing the techniques employed
EN 50128
Clauses 1-4
Appendix B
SIL,
People,
Documents
Clauses 5-7
Clauses 8-17
Appendix A
Techniques related to Clauses 8-17
Clauses 8-17: each clause gives a detailed description of one phase during the software's life cycle. The activities needing to be performed at each stage are identified.

Figure 3.5: [1] The structure of EN 50128:2001.

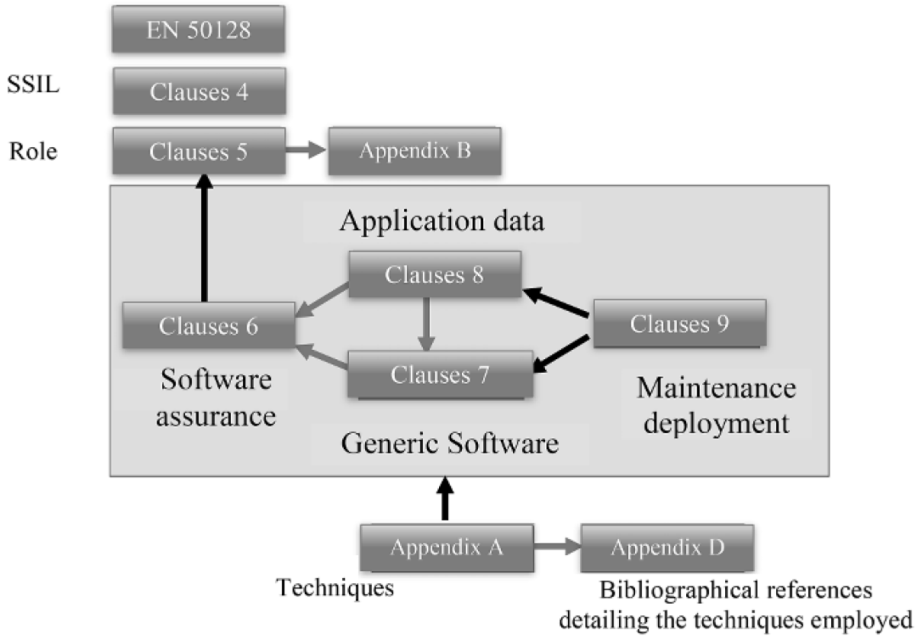


Figure 3.6: [1] The structure of EN 50128:2011.

The update can be experienced to be extensive, but many of the new requirements are clarifications of old unclear requirements. This is the case for the lifecycle documentations. In the 2001 version, they were presented as ”document groups” (see figure 4.2 and 4.3). The content of these ”document groups” has been clarified in table A.1 (see Appendix A, section 8.1), which made the lifecycle documentation clearer. The order of the phases are changed in the 2011 version of this standard, the lifecycle documentations are the same, but the structure of the V-model are slightly changed (see figure 3.7 and 3.8). The lifecycle documents with additional responsibilities for each document are presented in the 'Document Control Summary' in figure 8.1 and figure 8.2.

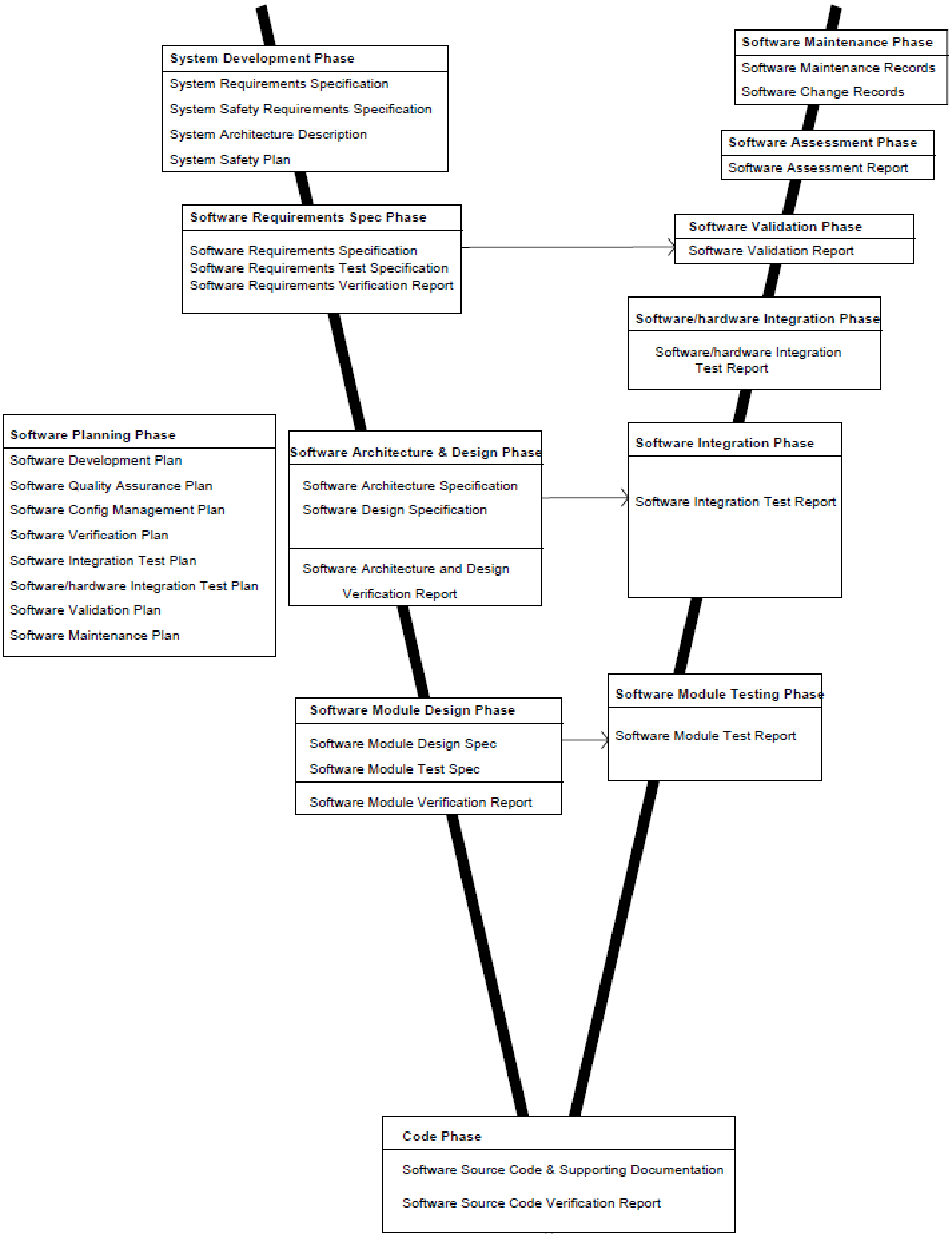

System Development Phase
System Requirements Specification
System Safety Requirements Specification
System Architecture Description
System Safety Plan
Software Requirements Spec Phase
Software Requirements Specification
Software Requirements Test Specification
Software Requirements Verification Report
Software Planning Phase
Software Development Plan
Software Quality Assurance Plan
Software Config Management Plan
Software Verification Plan
Software Integration Test Plan
Software/hardware Integration Test Plan
Software Validation Plan
Software Maintenance Plan
Software Architecture & Design Phase
Software Architecture Specification
Software Design Specification
Software Architecture and Design Verification Report
Software Module Design Phase
Software Module Design Spec
Software Module Test Spec
Software Module Verification Report
Code Phase
Software Source Code & Supporting Documentation
Software Source Code Verification Report
Software Module Testing Phase
Software Module Test Report
Software Integration Phase
Software Integration Test Report
Software/hardware Integration Phase
Software/hardware Integration Test Report
Software Validation Phase
Software Validation Report
Software Assessment Phase
Software Assessment Report
Software Maintenance Phase
Software Maintenance Records
Software Change Records

Figure 3.7: (EN 50128:2001) The illustrative Development Lifecycle - The V-model.

The V-model is an illustration of how the lifecycle documentations should be implemented in the different phases. It should be followed from left to right. The left side of the V-model are remained the same for the two versions of the standard. The "middle-phase" 'Code phase' is changed to 'Software Component Implementation Phase'. One of the lifecycle documentations is remained and one are removed to the next phase in the model for the 2011 version. The following two phases (from the bottom to the top on the right side) are remained the same. The next phase, 'Software/hardware Integration Phase' in the 2001 version are combined into the 'Software Integration phase' in the 2011 version. The 'Software Assessment Phase' is moved outside the V-model in the 2011 version. The remaining phases are the same for the two versions of the standard.

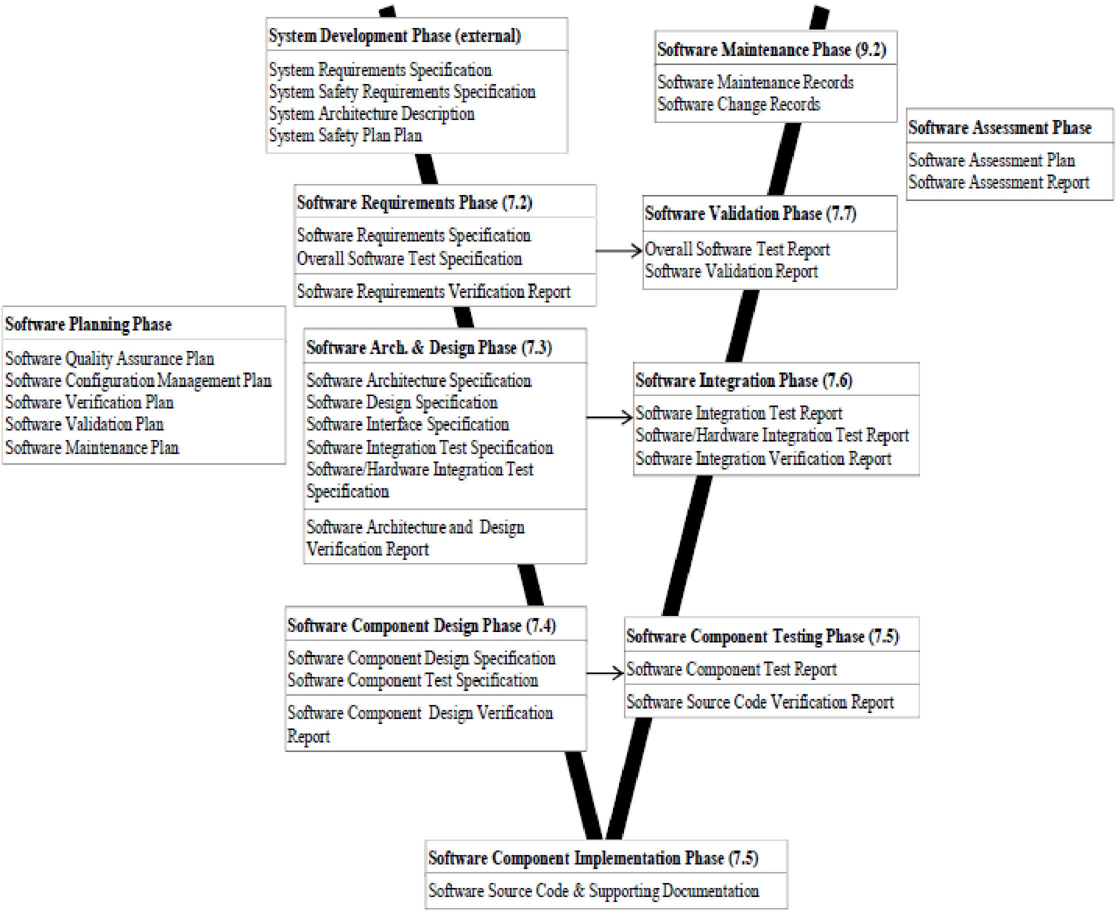

System Development Phase (external)
System Requirements Specification
System Safety Requirements Specification
System Architecture Description
System Safety Plan Plan
Software Maintenance Phase (9.2)
Software Maintenance Records
Software Change Records
Software Assessment Phase
Software Assessment Plan
Software Assessment Report
Software Requirements Phase (7.2)
Software Requirements Specification
Overall Software Test Specification
Software Requirements Verification Report
Software Validation Phase (7.7)
Overall Software Test Report
Software Validation Report
Software Planning Phase
Software Quality Assurance Plan
Software Configuration Management Plan
Software Verification Plan
Software Validation Plan
Software Maintenance Plan
Software Arch. & Design Phase (7.3)
Software Architecture Specification
Software Design Specification
Software Interface Specification
Software Integration Test Specification
Software/Hardware Integration Test Specification
Software Architecture and Design Verification Report
Software Integration Phase (7.6)
Software Integration Test Report
Software/Hardware Integration Test Report
Software Integration Verification Report
Software Component Design Phase (7.4)
Software Component Design Specification
Software Component Test Specification
Software Component Design Verification Report
Software Component Testing Phase (7.5)
Software Component Test Report
Software Source Code Verification Report
Software Component Implementation Phase (7.5)
Software Source Code & Supporting Documentation

Figure 3.8: (EN 50128:2011) The illustrative Development Lifecycle - The V-model.

Between June 2011 (release of 2011 version) and 2017-04-25 there has been a time of coexistence of the two versions. All new projects started during this time should follow the 2011 version. Since some processes were following the 2001 version and some the 2011 version during this time of coexistence, the operator had to take their responsibility of safety as follows: [1]

- Every operator has its responsibility for development and improvement of the safety of the railways.
- Prevention of serious injuries should be prioritized during the process.
- Safety management System should always be well documented in responsibilities and how the safety management System should be continuously improved.

Projects that started before June 2011 and were planned to finish before 2017-04-25, should follow the 2001 version throughout the project. Additionally, as a project would continue after 2017-04-25, the 2011 version of this standard had to be introduced to the project.

If a change or a maintenance had to be performed during the time of coexistence, 'minor' and 'major' could be handled differently. This was the case for projects started after 2011. If the change was classified as 'minor', it could be handled according to the 2001 version. If the change was classified as 'major', it had to be handled according to the 2011 version. The classification of the maintenance work has to be approved by the Assessor. If new functionalities are added, the environment changed, the SIL class changed etc. the maintenance is classified as 'major'. Otherwise, the maintenance is classified as 'minor'.

3.3 SIL

SIL is a way of measure the performance of a Safety Instrumented Function (SIF). A safety Instrumented Function is a combination of sensor(s), logic solver and final element(s) that detect a hazard and bring the process to a safe state [28]. SIL describes the quality and safety level of the system. This measure reference is created in order to sort the applications in grade of effecting the safety of the system. This is done in order to avoid an accident as far as possible.

Accident – An accident is an unexpected event or series of events causing death, injury, loss of a system or service, or damage to the environment [1].

SIL stands for safety integrity level and is defined on a scale from 0 to 4. SIL 4 products has the largest impact on the safety and 0 has the lowest safety impact. In the 2001 version of EN 50128, SIL 0 did not have any safety impact and therefore no requirements. This is changed in the 2011 version. SIL 0 has in the 2011 version, the lowest level of safety impact and can be seen as SIL 1 light. Requirements are added to SIL 0, as a result of the changed meaning. This change has made SIL 0 widely misjudged, the name are remained the same, but the meaning are changed. 32 documentations should now be produced for the SIL 0 products, compare to non in the 2001 version. The different levels of SIL can be described as [1]:

- SIL 4: catastrophic impact;
- SIL 3: impact on the community;
- SIL 2: major protection of the installation and of production, or risk of injury to employees;
- SIL 1: minor protection of the installation and of production;
- SIL 0: SIL 1 light, lowest level of safety impact;

The five SIL can be seen as three SIL, where SIL 0 is the 'lowest level', SIL 1 and 2 are the 'medium-level' and SIL 3 and 4 are the 'high-level'. With 'high-level' it means that the applications in this category has a high impact on the safety of the system.

In the 2011 version of EN 50128, the requirements related to the SIL is harder, especially for level 0, 2 and 4, but also in the roles of who/whom to perform the evaluations of the SIL. Requirements regarding the roles and responsibilities are stricter in the 2011 version, which is important. *The chance of human mistakes might be minimized with the right personnel, with the right experience, on the right place?!* The roles included in each level of SIL are illustrated in figure 3.9. Even though there is a lot more requirements for SIL in the 2011 version, the requirements are clearer explained and thereby easier to understand. *Can it be made even clearer?*

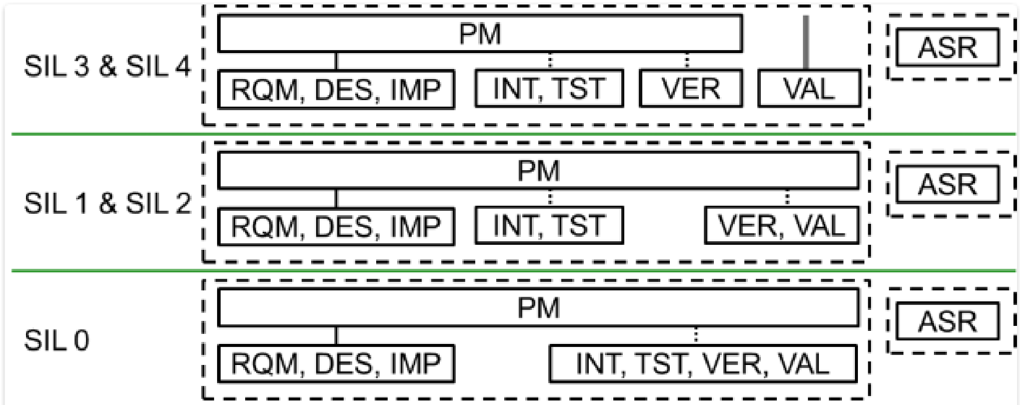
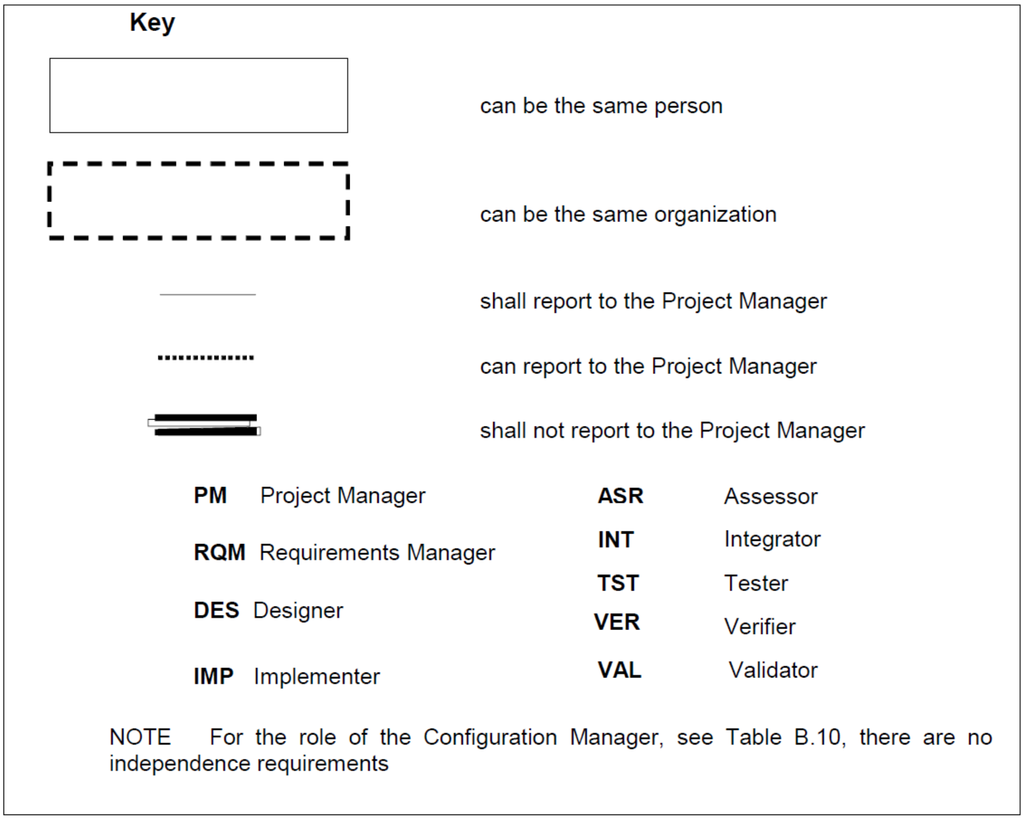


Figure 3.9: [19] (EN 50128:2011) The different levels of SIL are presented with required roles. A translation of the abbreviations are presented in figure 3.10.



Key

□	can be the same person
⬚	can be the same organization
—	shall report to the Project Manager
⋯	can report to the Project Manager
═	shall not report to the Project Manager

PM	Project Manager	**ASR**	Assessor
RQM	Requirements Manager	**INT**	Integrator
DES	Designer	**TST**	Tester
		VER	Verifier
IMP	Implementer	**VAL**	Validator

NOTE For the role of the Configuration Manager, see Table B.10, there are no independence requirements

Figure 3.10: [19] (EN 50128:2011) A description of the different roles and boxes presented in figure 3.9.

3.3.1 Perform a SIL study

SIL can not be measured. A SIL study has to be performed in order to evaluate the SIL of an application. There is a few different way of evaluating the SIL of an application. For the software application related to the standard EN 50128, this SIL study helps determine the:[28]

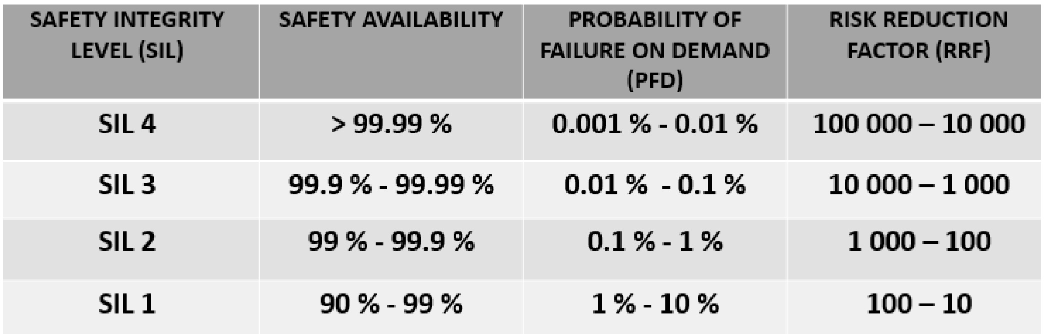
- type of device;
- hardware architecture;
- voting logic;
- proof test interval;

required to meet the target Risk Reduction Factor (RRF).

The SIL study is performed through an amount of steps:[28]

- Step 1: Break down the SIF into its components and architecture;
- Step 2: Calculate the PFD of each component. In order to perform this part, documented, historical failure rate data are required. There are many different formulas to use for calculating the PFD, depending on the product/process. PHD equals 1 minus Safety Availability, see figure 3.11 [12];
- Step 3: Combine all component PFD:s to determine the SIF PFD, which is simply done by adding all of the PFD:s of the different component of the product;
- Step 4: Translate the total PFD to RRF and compare this to the expected SIL.

If this RRF did not measure up to the expected SIL, the 'weakest link' has to be found among the components of the product. By 'weakest link' means the component that has the largest distribution of failure. In order to make the component enter the relevant SIL, the 'weakest link' has to be corrected. The different SIL with associated PFD:s and RRF:s is presented in figure 3.11.



SAFETY INTEGRITY LEVEL (SIL)	SAFETY AVAILABILITY	PROBABILITY OF FAILURE ON DEMAND (PFD)	RISK REDUCTION FACTOR (RRF)
SIL 4	> 99.99 %	0.001 % - 0.01 %	100 000 – 10 000
SIL 3	99.9 % - 99.99 %	0.01 % - 0.1 %	10 000 – 1 000
SIL 2	99 % - 99.9 %	0.1 % - 1 %	1 000 – 100
SIL 1	90 % - 99 %	1 % - 10 %	100 – 10

Figure 3.11: The interval of PFD and RRF for each SIL.

4 Results

In this section the result of the study is presented, including comments from experts and suppliers. The section is divided into subsections, where the most significant topics, as a result of the updated standard are presented. The significant topics, are parts of this standard that was recurrently pointed out in the interviews, as extensive parts of the change process. Relations are made to the tables in Appendix A, section 8.1, where tables with results and comments are presented. All of the new requirements are also listed in Appendix B (section 8.2), each clause/sub-clause is presented with additional comments to each requirement. A description of the changes and in some cases a background summary of the changes done, is also added in each sub-clause.

The products used in this study are presented as product 1 and product 2 are the company developing/produces these products are named 'supplier'. Comments in the result are taken from the supplier and from experts. Comments from experts are collected from the working group SGA18, lecturer and from interviews. It is important to take into account that this study is not intended to be an audit of the work of the supplier, but a study of the affect of the update from version 2001 to 2011 of EN 50128.

4.1 The change process

"The 2011 version of this standard is absolutely not crystal clear." (Supplier)

Even though the 2011 version of this standard is much clearer than the 2001 version, it is definitely not experienced to be crystal clear. In order to be clearer it still needs more intuitive explanations and examples.

"We have checked through the standard and made sure that everything is covered. Something unexpected can be seen as updates etc. are made." (Supplier)

The procedure to make sure that requirements for Product 1 and 2 follow the updated standard, has been to go through the 2011 version from the top to the bottom, step by step. Each part, especially for the methods/techniques and responsibilities has been compared to the 2011 version of this standard, to make sure that everything is covered. If a part was not covered before the update, that part was updated or justified in order to fit the requirements of this standard. The work of going through the process, part by part and performing the changes required, took approximately 500 hours for each of the products to proceed. Which would result in a cost divided between TRV and other orderer of approximately:

$$500 \times 2000 = 1\,000\,000\,SEK \tag{1}$$

for each product. These costs are calculated in beforehand and does not have to be close to this sum calculated afterwards. If 500 hours is exactly the amount of hours that the supplier expected for this change, and this sum for developing their Generic Product (GP) is divided between their involved customers, approximately 10, this will result in a total cost for TRV for this change process of approximately:

$$\frac{1\,000\,000}{10} = 100\,000\,SEK \tag{2}$$

for each product. This cost is only a short term cost, only costs for changing version of the standard, not costs for working with the 2011 version of this standard. A large part of these 500 hours was spent on the classification of the tools, see section 4.1.2.

"The Assessor has been updating the working methods/techniques and documents along the way, to the update of this standard. The Assessor has been aware of the requirements in the 2011 version of this standard, as these updates has been done. Due to this, the process of changing the version of this standard in the process was not experienced to be extensive." (Supplier)

Many parts of the 2011 version of this standard had slowly been introduced by the assessor. The assessor had the 2011 version of this standard in the back of her/his head as the work proceeded with the 2001 version of this standard. The assessor is working as a link between the working groups and the supplier, both in order to support the supplier, but also to review the process executed by the supplier. As a link between the working groups and the supplier, new drafts and changes due to the standard used are always in the hands of the supplier as soon as something is released. This way many parts were already in the process when the complete update of the standard has been released. Because of the slow introduction of new parts of the process, the update did not feel as extensive as it would have felt, if everything was introduced at the same time. As a result of this, the exact amount of resources spent on the updating process is hard to define, since the updates made in beforehand are not included in the total resources spent on this standard update.

"Waited as long as possible with the change in version of this standard, as a result of the uncertainly about what this update would imply for us." (Supplier)

Product 1 is completely developed following the updated standard. When it comes to Product 2 it is not completely developed following the standard, due to the uncertainly about what this update would imply for the product/company. The supplier is also saving these potential extensive updates as long as possible in order to not waist resources, even though they admit that waiting is not the correct way to go. However, a positive result of waiting with product 2, has been that a lot of the "pre-work" have been done during the development of product 1. I.e. when the supplier initiated the process to make sure that also Product 2 follows the updated standard, they could take advantage of the "pre-work" done for Product 1, e.g. templates in form of tables etc. This makes the work easier for the staff working with product 2 to update the standard for this product.

Since the products recently has been introduced to the 2011 version of the standard, there has not been any updating issues detected or any problems with maintenance work detected at this stage of the process, but the supplier indicates that it might happen further into the process.

"This European standard lacks in examples." (Supplier)

This European standard is considered to lack examples. Due to this shortage some parts is taking longer time to perform than they would taken if an example for each new part was included in this standard. This is especially the case for the new requirements regarding the tool classification (see section 4.1.2). If examples of how the descriptions in the different classes should be formed, this process of classification of the tools would be much more time efficient and thereby less expensive.

"We used IEC 61508 as a guide in order to find examples of how to execute new parts of the process, that are required by EN 50128:2011. We also investigated how the car industry handled these new parts of the process." (Supplier)

Other standards that affected similar subjects was investigated in order to find examples execution of requested new parts of the standard. Again resources would be saved if more and clear examples of how to execute the new parts of this standard was presented in this standard.

"Some parts in this standard feels clearer in the 2011 version of this standard, than in the 2001 version of this standard. This can be due to the fact that we recently read the 2011 version of this standard many times, in order to really understand it." (Supplier)

The structure of the 2011 version of this standard is intended to be clearer than its predecessor. Many of the new requirements are therefore added only as a clarification of old, woolly requirements. Parts of this standard that felt unnecessary has been deleted, and parts that has been in the process of the supplier for a long time, has been added.

4.1.0.1
The main groups of requirements are reduced in number (Figure 4.2 and 4.3), and reorganized in order to have a clearer structure. Appendix B, including a description of the roles and associated responsibilities are added in order for EN 50128 to be clearer.

”Is component the correct word for this purpose? Is it defined in the correct way in EN 50128:2011?” (Experts)

Whether or not ’component’ are correctly defined are discussed in SG A18. ’Module’ (2001 version) was changed to ’component’ (2011 version). This was done in order to decrease the amount of confusion and misunderstandings, *But, has this been the outcome? Or, does it has to be clearer defined?* It seems like the users of the standard knows the meaning of ’component’, *but are they all having the same view of the meaning?* There could be misunderstandings, if the definitions of the key words are not defined clear enough.

> 4.1.0.2
>
> 3.1.4 component:
>
> a constituent part of software which has well-defined interfaces and behavior with respect to the software architecture and design and fulfills the following criteria:
>
> - it is designed according to “Components” (see Table A.20);
> - it covers a specific subset of software requirements;
> - it is clearly identified and has an independent version inside the configuration management system or is a part of a collection of components (e.g. subsystems) which have an independent version.

4.1.1 SIL

The supplier indicated the confusion of whether a product/process is SIL 3 or SIL 4, and the same for SIL 1 and SIL 2. One purpose of the 2011 version of this standard was to make SIL clearer and easier to use. However, there is still a lot of confusion around SIL, perhaps more, due to the new requirements regarding SIL 0.

”What is the significant differences of the different SIL?” (Supplier)

In this standard, the requirements are exactly the same for SIL 3 and SIL 4, and exactly the same for SIL 1 and SIL 2. By looking at the standard, there is nothing that differs. Is it then any purpose of having five different SIL, if there is only three different levels of requirements? There is one significant difference between SIL 1 and SIL 2, SIL 1 products/processes can only harm the hardware and does not have any impact on humans, but SIL 2 products/processes affects the safety of the humans as well. The supplier does not have any SIL 3 product, since the only difference between SIL 3 and SIL 4 is the allowed level of error (not included in this standard), see figure 3.11. Therefor the supplier are only using SIL 4, not SIL 3.

> 4.1.1.1
>
> SIL 4: catastrophic impact (PFD 0.001 % - 0.01 %);
>
> SIL 3: impact on the community (PFD 0.01 % - 0.1 %);
>
> SIL 2: major protection of the installation and of production, or risk of injury to employees (PFD 0.1 % - 1 %);
>
> SIL 1: minor protection of the installation and of production (PFD 1 % - 10 %);
>
> SIL 0: SIL 1 light, lowest level of safety impact;

Results has been proven that if a product changes from one SIL to an other (e.g. SIL 2 to SIL 3), the costs are increased by about 100 %.

”A product that is classified as SIL 0, is a product that has a good quality, but does not have any safety impact.” (Supplier)

This is a common interpretation of SIL 0, which is not the purpose. EN 50128 includes a lot more requirements regarding the lowest level of SIL, SIL 0. The purpose of including more requirements was not to include products/processes that are not safety related. SIL 0 was intended to work as a SIL 1 light, in order for some processes to be easier to handle. This was not explained clear enough in this standard, which led to a confusing for the users. The question *"Why should we use this standard for safety related software, on non safety-related products?"*, was asked, when the requirements for SIL 0 was added. As EN 50128:2011 in used, no integrity level are dealing with products with no safety impact. I.e. all products connected to this standard have some kind of safety impact. Again this confusion could be clarified, if clear explanations were added.

EN 50657 - *Railway applications - Rolling stock applications - Software on board of rolling stock, excluding railway control and protection applications* is intended to adapt EN 50128:2011 for the application in the Rolling Stock domain. In EN 50657, SIL 0 is replaced by 'Basic Integrity' (BI). The BI can be a software that is not safety related, in comparison to SIL 0, which is aimed to have the lowest level of safety impact. The BI does not need to be assessed and there is a lot less requirements to the BI, than to SIL 0. This was one of the reasons why EN 50657 was created, in order to easier work with non safety impact or safety impact below SIL 1 for Rolling Stock applications. Some experts would like the BI to be included in EN 50128:2011 as well. Does the aim of EN 50128 then be changed as well? Since it is focusing on 'Safety related Software'. The differences of SIL 0 and BI are explained in 4.4 in the standards and shown below:

> 4.4 in EN 50128:2011:
> At least the SIL 0 requirements of this European Standard shall be fulfilled for the software part of functions that have a safety impact below SIL 1. This is because uncertainty is present in the evaluation of the risk, and even in the identification of hazards. In the face of uncertainty it is prudent to aim for a low level of safety integrity, represented by SIL 0, rather than none.
>
> 4.4 in EN 50657:2016:
> The basic integrity requirements of this European Standard shall be fulfilled for the software part of functions that are not safety-related or that have a safety impact below SIL 1.

4.1.1.2
SIL 0 in 2001 version: no safety impact;
SIL 0 in 2011 version: lowest level of safety impact (SIL 1 light);

The new requirements for the SIL 0 products has cost TRV a lot. The invoice system is not designed to be detailed enough for these kind of invoices to be separated from other invoices from the same product, but estimates can be done.

"We are dealing with SIL 0 products in the same way as with products of other SIL." (Supplier)

The supplier is treating all of their products as SIL 4 products, independent of SIL. This is done since they have a working process of dealing with the highest level of SIL, which makes it easy to implement this working methods on the products of lower SIL as well. This makes the system safer, but for some projects/processes, this might be a to extensive and expensive work.

4.1.2 Tools

The process of fulfilling the new requirements in this standard regarding the tools, has been an extensive work. This part of this standard took approximately 160 hours to deal with for each product. Again, if these hours would be invoiced afterwards, TRV would share this cost with the other orderer of each product. This would result in a total cost of approximately:

$$160 \times 2000 = 320\,000\,SEK \tag{3}$$

for each product. For TRV would this be a cost of approximately:

$$\frac{320\,000}{10} = 32\,000\,SEK \tag{4}$$

for the tool classification process for each product/process following this European standard. The cost varies depending on the size and amount of tools used for the product/process. Again, this is only an approximation of what the cost would be if the costs would be invoiced in afterwards.

All of the tools in the system are divided in three groups, T1, T2 and T3, where a tool in T3 has the largest impact on the safety of the system. To each tool a description is added, including a description of usage, possible failure effects and mitigations (see an example in figure 4.1 of how the classification for a tool can look like). For tools in T3, additional information can also be added.

4.1.2.1
2011: Requirements regarding the tool classification are documented in clause '6.7 Support tools and languages'. T1, T2 and T3 are defined in 3.1. Not included in the 2001 version.

"It was hard to fully understand how the Tool Classification should be executed. More examples in the 2011 version of this standard would have been preferable." (Supplier)

Some parts was also hard to add to the correct folder and to know whether or not all of the tools should be included in the sorting. The standard states that the tools should be sorted, but not specific that all tools have to be sorted. The supplier translation of this part is that all tools should be included in the classification. They also says that it might not be wrong if a tool that is not significant important, should be excluded from this sorting.

All tools that were used for product 1 and product 2 before the classification were found to be justified.

"The texts with descriptions should have had a more clear purpose." (Supplier)

Is it really important that the tool 'Calculator' is included in this sorting of tools? If the answer on this question is 'yes', the standard should be clearer on this matter, and include a clear sentence that the sorting of tools should include all tools. The supplier ones again commented the lack of included examples in the standard. For this part, a list of examples of possible tools in each group would be preferable. Furthermore, an example of how the description of the tools in each class could look like.

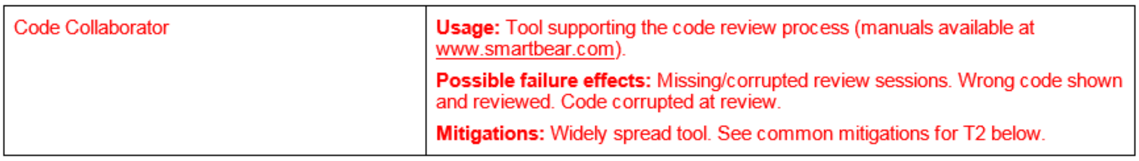
"The tools could have been more structured from the beginning." (Supplier)

The supplier working with this process, said that the tools should have been divided into some kind of groups even before this standard update process. It is of course easy to be wise after the event, but if there were a structure of the tools, less work would have had to be done.

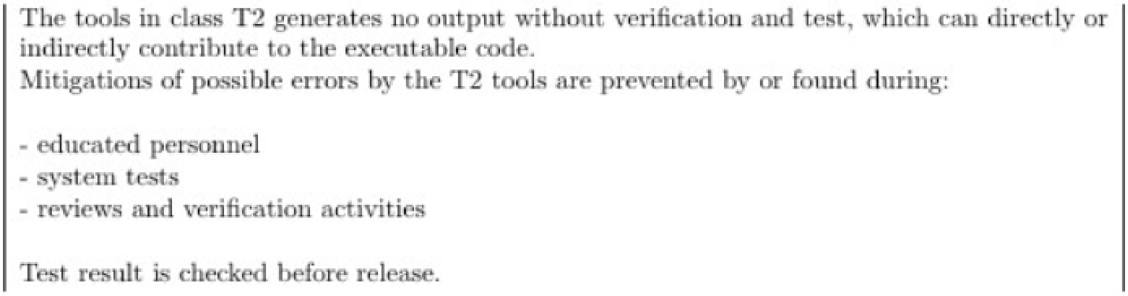
"It was a conscious decision to wait as long as possible with the classification of the tools. Afterwards it might be seen as a bad decision, but somebody has to have time to do it and it costs." (Supplier)

The work of dealing with this part of the standard started in the last minute, especially for product 2. The decision to wait with this update was a consciously decision according to the supplier. Later on, this decision was not seen as the best decision due to the time pressure, but somebody have to put aside their daily work to transact the tool classification. The classification work approximately required one full time working personnel for a month, for each product/process.

"A product that uses one or more tools, for example product 2, is referring to all evaluations that is done for each of the tools. Each tool are providing an analysis due to this European standard. See figure 4.1 for an example of how the description for a tool in class T2 could look like. For a tool in T3, more information and references to the documentations are added." (Supplier)



Code Collaborator	**Usage:** Tool supporting the code review process (manuals available at www.smartbear.com). **Possible failure effects:** Missing/corrupted review sessions. Wrong code shown and reviewed. Code corrupted at review. **Mitigations:** Widely spread tool. See common mitigations for T2 below.



The tools in class T2 generates no output without verification and test, which can directly or indirectly contribute to the executable code.
Mitigations of possible errors by the T2 tools are prevented by or found during:

- educated personnel
- system tests
- reviews and verification activities

Test result is checked before release.

Figure 4.1: An example of a description of a tool in class T2.

Examples of tools in T1 is MS word, MS exel, calculator, in T2 is RTRT and in T3 is data preparation tools.

"The Assessor and the Validator will support the tool classification process. The descriptions of the tools might not by them self give the full proof of the functionality, but by help from proofs of the related process, can test and result analysis help with the calculations of any faults." (Supplier)

Even though each tool has a description (see an example in figure 4.1), the descriptions might need support from tests and result analyses. From these analyses any failures of the tool can be detected, so there is a lot of work with this classification process. The Validator and the Assessor are supporting the supplier with feedback in order for the information to each tool to be correctly described and all possible failures to be detected.

4.1.2.2
2011: 6.7.4.5 includes requirements for the results of the tool validation.

4.1.3 Organization

"The clarification of the roles are good." (Supplier)

In the 2011 version of the standard there are a lot of new requirements regarding the roles and the organization. Each role is described in detail, which is appreciated by the supplier, and which results in a lower cost for TRV, due to the clarification. This detailed listing of the roles are presented in Appendix D, section 8.4.

4.1.3.1
The description of the roles and responsibilities are included in Appendix B, in the 2011 version. Not included in the 2001 version.

"We had all of the required roles within the company, so no new employments had to be done." (Supplier)

These new requirements could have been an extensive process to handle if the required qualifications of the staff were not within the organization. Since product 1 and product 2 is developed by a large organization and the products are involving a large amount of staff, the new requirements was not a problem for the supplier.

"If staff was missing, the projects could easily borrow staff from each other within the organization." (Supplier)

If there were problems in the process with lack of required staff, the organizations of the products can borrow staff from each other, for certain tasks. The supplier describes this as a result of the luck of having the same staff in the organization for a long time. Even though this makes the process easier, TRV has to pay for the extra working hours in the process, due to this extension in roles of this standard.

"We are lucky that nobody in the projects has quit or been log term ill." (Supplier)

The organization has also been fortunate to not have anybody on sick leave or similar in the process, when they have been working with the 2011 version. Small mistakes in the organization were detected in the process of developing product 2. In order to make sure that the process is following the 2011 version, the supplier had to go back in the process, and correct the mistakes.

"Experienced staff might experience new requirements as unnecessary, the same might not be the case for new staff to the process." (Supplier)

Experienced staff can in many cases be a valuable resource to the companies. These experienced staff can share information about the products within the company, but they can also be afraid or against changes in a process (see section 1.1.3). They can in many cases see these changes as a burden in a working process. It is important that the required updates/changes are applied even if they might experience it to be unnecessary, otherwise it will give the companies trouble as the experienced staff are ending their work in the projects.

"The role of the Validator in the 2011 version of this standard has been discussed. The role is to extensive and are to much alike the role of the Assessor." (Supplier)

Especially the Validator was experienced to be a to large role for one person to handle. The responsibilities for the Validator was experienced to cover the entire process, which felt unnecessary, since the responsibilities also included some of the responsibilities of the Assessor. The Validator has been divided between more than one person in order to work better in the large processes, e.g. for product 1 and 2. The description of the roles are presented in Appendix D, section 8.4 and the new requirements connected to the roles are presented in Appendix B, section 8.2. It will not be more costly to divide the role into more than one role, but unnecessary resources are spent on the parts where the Validator and the Assessor are doing the same work.

4.1.3.2
2011: Table B.7: Validator and Table B.8: Assessor.

"We do not have any specific documentation where the knowledge of the staff are documented, but we do have a safety log where competences and experiences are documented. We will start doing 'mini-safety-audits' as well." (Supplier)

In the 2011 version of this standard, requirements regarding the roles are added, see Appendix B, section 8.2. Besides the fact that the amount of roles are extended, restrictions regarding the recording of the roles are also added. The supplier are using Safety logs, where they are recording the competences and the experiences of the staff. The supplier also has the intension of doing internal 'mini-safety-audits'. These internal audits will help the supplier to be more prepared when the Assessor are performing safety audits. The internal 'mini-safety-audits' are also performed in order for the supplier to have a better control of the organization and documentations, since the system developed/produced are safety related.

4.1.3.3
2011: Requirement 5.1.2.3: The personnel assigned to the roles involved in the development or maintenance of the software shall be named and recorded.

Do we need the 'roles'? Maybe it should be enough with the activities and competences defined, integrated over phases?" (Experts)

Even though the clarification of the roles and related responsibilities gets a positive review, the roles can also be experienced to be unnecessary. Some parties believes that it would be enough to document activities and the competences. The work that has to be done should be detailed explained over phases, with competences for the work, but it might not necessary be defined as roles.

"The Assessor are performing audits. He/she is interviewing the staff of the project, in order to make sure that everybody is doing what they are intended to do." (Supplier)

A safety audit is an independent examination to determine whether activities and related results conform to planned arrangements and whether these arrangements are implemented effectively and are suitable to achieve the organization's policy and objectives [32]. The assessor are at these audits controlling that the required documents are easy to assess, that the staff knows where to find the documentations, that the required documents gives a correct description of the process, the competences of the staff etc. These audits were experienced to be extensive and the staff had to be well prepared to these audits. Additional requirements for the work of the Assessor are added to this standard, see Appendix B, section 8.2, even as these requirements does not affect the supplier, it is important that the supplier/user are making sure that they are aware of these requirements.

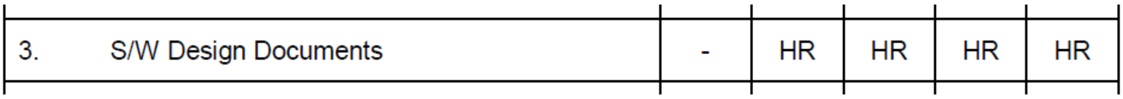
"The Traceability needed to be better in the process of the lifecycle documentations." (Supplier)

The objective of Traceability is to ensure that all requirements can be shown to have been properly met and that no untraceable material has been introduced [15]. The traceability are continuously in a developing stage, it can always be better, explains the supplier. The 2011 version of this standard has the purpose of making the system more traceable, compare to the 2001 version of this standard. To all of the lifecycle documentations, requirements regarding the traceability of the system is added, see Appendix B, section 8.2. How traceability is performed is translatable for the user/supplier, but it is in their interest to have a system that is as traceable as it can be.

4.1.3.4
2011: Requirements 5.3.2.7 to 5.3.2.10.
2001: Requirement 7.2.6.

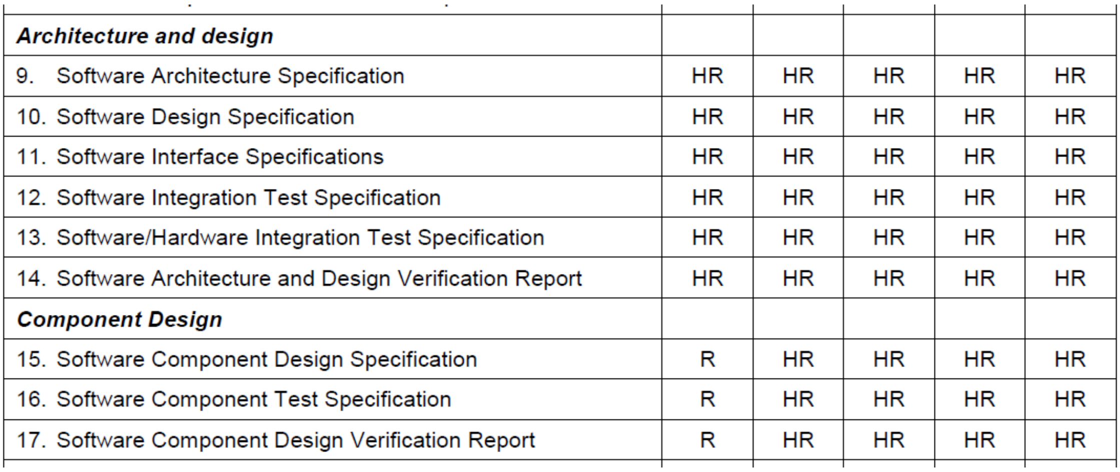
4.1.4 Lifecycle Documents

In the 2011 version of this European standard, 46 life cycle documents have to be created. In the 2001 version of this standard there were 29 lifecycle documents to be documented. The number of lifecycle documentations seen in table A.1 for the 2001 version, see Appendix A, section 8.1 can be misleading, since all 29 lifecycle documents are not included in this table of lifecycle documents. This table is only including the "document-group" of the lifecycle documentations. This can be misleading, since more than one document can be required in order to fulfill that lifecycle document, due the 2001 version of the standard. The lifecycle documentations in the 2011 version of this standard can be seen as an extension or a clarification of what the lifecycle documentations in the old version should include. An example of this is described in figure 4.2 and figure 4.3. These figures shows how one lifecycle document from the 2001 version of this standard is expanded into nine lifecycle documents in the 2011 version of this standard. As the V-model (figure 3.7) or the Document Cross-Reference (figure 8.3 for the 2001 version of this standard, it can be seen that some documents are added to each "document-group" of the lifecycle documents. This way the lifecycle documents of the 2011 version of this standard are much clearer than the 2001 version. Even though there is a lot more lifecycle documents, this might not be seen as a lot of new lifecycle documents for the user/supplier. If the company using this standard is in the forefront in the software development market and are using a good assessor, they will probably already be using all of the lifecycle documentations required in the 2011 version of this standard, as the 2001 version of this standard are used. This is the case for this supplier. In some cases the lifecycle documentations are named differently by the supplier, compared to the names used in this standard. The standard does not say that the name has to be the same as it is in the standard, as long as the content of the lifecycle documentations are following the requirements of this standard.



3. S/W Design Documents	-	HR	HR	HR	HR

Figure 4.2: An extract of the lifecycle documentations of the 2001 version of the standard EN 50128, EN 50128:2001.



Architecture and design					
9. Software Architecture Specification	HR	HR	HR	HR	HR
10. Software Design Specification	HR	HR	HR	HR	HR
11. Software Interface Specifications	HR	HR	HR	HR	HR
12. Software Integration Test Specification	HR	HR	HR	HR	HR
13. Software/Hardware Integration Test Specification	HR	HR	HR	HR	HR
14. Software Architecture and Design Verification Report	HR	HR	HR	HR	HR
Component Design					
15. Software Component Design Specification	R	HR	HR	HR	HR
16. Software Component Test Specification	R	HR	HR	HR	HR
17. Software Component Design Verification Report	R	HR	HR	HR	HR

Figure 4.3: An extract of the lifecycle documentations of the 2011 version of the standard EN 50128, EN 50128:2011.

Some of the descriptions of how to create lifecycle documents in the 2001 version of the standard EN 50128, are long and hard to understand. These long descriptions have been divided into more than one description in the 2011 version of this European standard. These requirements regarding the lifecycle documentations are formed as in figure 1.1 for all of the different lifecycle documentations. This is a clarification in words of the Document Control Summary (see section 8.1.2).

"Almost all of the new lifecycle documentations were already in use. The assessor required some of the lifecycle documents from the 2011 version of this European standard, before this version of the standard were in use." (Supplier)

These new requirements in this European standard are experienced as a clarification by the supplier, not a new process to perform due to this European standard. The new version might be seen to be more extended than it is, due to all of the clarifications.

"The Waterfall model does not support our process. All lifecycle documents are included in the process, but not used in the order suggested by these methods." (Supplier)

The supplier are creating all of the requested lifecycle documentations required by this European standard, but not in the order recommended by the Waterfall model. The Waterfall model is only a recommendation of in which order the lifecycle documentations should be created. If an other order fits the process better, that order should be used. The Waterfall model seems to be a bit unclear to this standard, even standardization experts connected to this standard where unsure what the Waterfall method intended.

4.1.4.1
Requirement 7.1.2.2 describes that the order of the lifecycle documentations in table A.1 is the ideal order, the waterfall model. However, it is not a requirement to follow this order.

"If different persons are coding one code, the specification are required in the beginning, so each persons knows whats expected of their part of the code. If only one person is creating the code, the requirements can be written as specifications in the code. These parts can then be cut out and used

as specifications afterwards." (Supplier)

This is two ways of fulfilling the requirements of the specifications. There are new lifecycle specifications added to this European standard, see Appendix B, section 8.2. Since there is no requirements from the standard or TRV in how the lifecycle documentations should be produced, only that they have to be produced, there are a lot of different ways of producing the documents. There is also no requirements regarding how the lifecycle documentations can be combined, so the combination of documentations were in many cases made in order for the standard update to be as smooth as possible, and as a lifecycle documentation feels unnecessary, it can be combined into an other lifecycle documentation. This is another reason of why the update in standard were hard to measure in time and costs.

4.1.4.2
2011: 5.3 Lifecycle issues and documentation.
7.1 Lifecycle and documentation for generic software.
2001: 7. Lifecycle issues and documentation.
The lifecycle documentations are also listed in table A.1, Appendix A (Appendix A, section 8.1 in this report).

4.1.5 Test

There are new requirements to the testing procedure, to all testing phases. The supplier did not experience this part of the 2011 version as an extensive work to update. Many of the new requirements for testing was already in use in the system of product 1 and product 2.

"The requirements for tracing and testing has been in the process even as the older version of this standard was followed. We are and will continuing doing improvements in this area. New requirements regarding this did not affect our process in this area." (Supplier)

As mentioned earlier, there is a lot of requirements in the 2011 version that were implemented to the system, as the 2011 version of this standard already were in use. The continuously process of improving the system will always be the goal, says the supplier.

"Blackbox testing are only performed on the overall system." (Supplier)

This European standard does not state that Blackbox testing has to be performed on all phases. Therefore, 'Greybox testing' and 'Whitebox testing' are performed on most of the phases of the process. These tests are cheaper to perform, but not as safe as if blackbox testing would be done in all phases. This is a part where TRV should ask for the blackbox testing in all phases, if that is requested by them, and if they are up for paying for it.

4.1.5.1
2011 and 2001: 'Functional/Black-box Testing' in Table A.5 (M), A.6 (HR), A.7 (M) and all of A.14.

"We can not do all testing on the products, like 'Overall testing'. This is due to the fact that there might be a load missing or the correct environment is not available." (Supplier)

As it can be motivated why parts of this standard can not be followed, the parts can be approved excluded from the system. For product 1 and product 2 this is the case with table A.11 - Data preparation technique, table A.16 - Diagrammatic Languages for Application Algorithms, table A.22 - Object Oriented Software Architecture, table A.23 - Object Oriented Detailed Design and partly for the subsection in table A.1 - System configured by application data/algorithms, see Appendix A, section 8.1. This is the case since product 1 and product 2 are Generic Products (GP). Parts of the standard that can not be covered by the GP, has to be covered by the Generic Application (GA) or the Specific Application (SA). In these cases it is important to document all parts of the standard that are covered by a product, in order for the final product to be covered by this standard. This work is extended in the 2011 version of this standard, since tables that can not be covered by the

GP are added. Object oriented design and architecture are processes that are looking at the whole system, with interacting objects, hardware/software etc.

4.1.5.2
2011: 6.1: Software testing.
7.7: Overall Software Testing/Final Validation.
2001: Some requirements from 7.7 (2011) in 13: Software validation. Most of the requirements are new.

4.1.6 Maintenance

As the 2011 version of this standard seasonly has been introduced to product 1 and product 2, not much maintenance work has been done as the new version of this standard has been followed. The supplier does not see the new requirements to this section of the standard as an extensive work. They are commenting on the fact that since they have not followed the 2011 version of this standard for a long time, extra costs could be a result as updates in the system are made.

"It is hard to define whether the maintenance is major or minor." (Experts)

As requirement 9.2.4.2 (see Appendix B, section 8.2) says, the supplier should decide whether the maintenance work is major or minor, before any maintenance work starts. As a decision is made, the Assessor has to approve the decision. If the supplier can not decide whether the maintenance is major or minor, the Assessor has to make the decision. This part is new to this standard. In the 2001 version, the decision did not have to be assessed by the Assessor.

The process of defining whether a maintenance work is major or minor are experienced to be hard. The differences between major and minor is the safety impact of on the system of the maintenance work. If the environment or a part of the software are changed, the maintenance is valued as a major maintenance. It could be tricky to decide, even with this knowledge of the meaning.

4.1.6.1
Requirement 9.2.4.1 and 9.2.4.2 in the 2011 version and 16.4.6 in the 2001 version. The decision has to be evaluated by the Assessor in the 2011 version.

4.1.7 Tables

In Appendix A, section 8.1, the tables for product 1 and product 2 are listed with used methods/techniques for each product. In this section of the result, some interesting parts connected to the results in the tables are presented.

This European standard states, in the section of the tables that:

"The combination of techniques or measures are to be stated in the Software Quality Assurance Plan or in another document referenced by the Software Quality Assurance Plan with one or more techniques or measures being selected unless the notes attached to the table makes other requirements. These notes can include reference to approved techniques or approved combinations of techniques. If such techniques or combinations of techniques, including all respective mandatory techniques, are used, then the Assessor shall accept them as valid and shall only be concerned that they have been correctly applied. If a different set of techniques is used and can be justified, then the Assessor may find this acceptable."

This means that if you are good enough in arguing of why you should not use the recommended set of documentations/methods/techniques and have the agreement of the assessor, you do not have to follow the recommendations in the tables of this standard, but anyway be approved by this standard. A to good argumentation can lead to an unsafe system, and at the same time be approved by this standard.

"We want and need to use 'Formal methods' on all of our products. This is a good way of testing

all the logic in the code." (Supplier)

As can be seen in Appendix A, section 8.1, product 1 does not use 'Formal methods' and 'Formal proof' in their process. It is recommended to use this method in order to test the logic of the code, but it is not a requirement to use it, only a recommendation. Once again it can be pointed out that as no other requirement from an orderer or other outsider are added to work with this standard, the user/supplier can use the standard as they like, as long as it is approved by the assessor.

4.1.7.1
2011: 'Formal methods' in Table A.2, A.3, A.4 and A.17.
'Formal proof' in Table A.5 and A.11.
2001: 'Formal methods/proof' in Table A.2 and A.4.
'Semi-Formal methods' in Table A.2 and A.18.

"Since the products are Generic products, some of the tables in this standard were not used." (Supplier)

Some of the tables in this standard were not used by the supplier, since the supplier are developing/produces a generic product (GP). As has been mentioned earlier, as the user/supplier argues well why certain part of the standard should not be used and the assessor approves the argumentations, the user/supplier can exclude that specific part of this standard from their process.

4.1.7.2
The tables that are not used in the 2011 version:
Table A.11: Data Preparation Techniques.
Table A.16: Diagrammatic Languages for Application Algorithms.
Table A.22 Object Oriented Software Architecture.
Table A.23 Object Oriented Detailed Design.

The supplier are following a process called 'engineering process'. This process describes the whole chain from a Generic Product (GP) to a Generic Application (GA), to a Specific Application (SA). The goal is that GA and SA take care of the parts of this standard where the GP did not cover it. This way, all required parts of the standard should be covered in the end of the 'Engineering process'.

"Woolly and unclear with checklists." (Supplier)

In the 2011 version of this standard, checklists are included more frequently in the tables. The aim of the checklist is to provide a stimulus to critical appraisal of all aspects of the system rather than to lay down specific requirements [15]. A checklist does not seems to be a point that is hard to implement, but it can be hard to understand what to include in the checklists the first time it is about to be applied to a process. Even though there is a description of what a checklist is and what the purpose of it is added the Annex D of the 2011 version of this standard, examples of what a checklist could include and what to think about as it is produced should be added to this standard, says the supplier. Clarification and examples added to this part of the standard, as well as to other parts, could be a saving in resources for TRV.

4.1.7.3
2011: Checklists are mandatory (M) in Table A.11 (SIL 3 and 4) and recommended (R) in Table A.9 and A.19.
2001: Checklists are highly recommended (HR) in Table A.9 and recommended (R) in Table A.19.

5 Analysis

In this section an analysis of the methods used, the overall experiences and thought due to the affect of the update from version 2001 to 2011 of EN 50128 is presented.

5.1 Analysis of the result

From an outside perspective, the change in version from 2001 to 2011 for a company might be seen as an extensive change. This is due to the fact that the 2011 version contents almost double the amount of requirements, compare to the 2001 version. The conclusion after studying the version update of this standard is that, it is not experienced to be as extensive as it might seem.

How can this change in version of this standard be done smoothly by the companies following the standard? (Question 1) The supplier using this standard were comparing the 2011 version of this standard with their current process of methods/techniques, documentations etc. This way the supplier could get a good view of what they had to add/modify to their process, in order to follow the 2011 version. As the process was updated for one product, it was a less extensive work to update other products within the same company. For the supplier, this way of executing the change process seems to be the most efficient. As will be discussed further on, the work could have gone smoother with e.g. additional guides of how to use the standard, both added to the standard and from TRV as an orderer.

TRV is spending a lot of money on this update in standard, without being a part of the update process. It should be in the interest of the orderer to participate in the process of updating a standard. *How much should TRV be involved in the work of the supplier?* This question has been asked to more than one persons involved in this process and the answers to it varies a lot. Some persons would like to see a deeper collaboration on a technical level, compared to how it is handled today, and some persons indicates that it is the best if everybody takes care of their case. All parts of the process can obviously not be shared with the orderer, but in many parts of the process, it might be good for both the supplier and the orderer, if the orderer would be a participant. The contents of the documentations and how the documentations are combined should be in the interest of the orderer, as well as how the testing procedure is planned to be proceeded.

Maybe the testing that is planned by the supplier, is not the way TRV would like it to be. The tables in Appendix A, section 8.1 should also be an agreement between TRV and the supplier. These tables are presenting the methods/techniques used by the supplier, both for the 2001 (where details were documented) and 2011 version of the standard. The tables are complemented with an additional section, where the associated requirements for these tables are presented and discussed. The supplier should use these tables and suggest methods/techniques, in order for TRV to approve the suggested combination, in the beginning of a project. Overall, a suggestion is for the orderer and the supplier/developer to work closer together, in order for the final product to be as safe as possible and understood by more parties.

TRV should make a plan of how the complete system should work in terms of safety, in a deeper, more technical way than it is done today. They should be clear about who is responsible for what and make sure that all parts of a complete system is safe, in all aspects. Additionally, they should control every aspect of the integration, for all parts of the system. Due to this, they should make sure that the requirements of the safety standards are all followed as the integration is executed. This could help the supplier follow the requirements that they indicated was hard to execute, e.g. overall testing (see section 4.1.5). Overall, it can be seen in this study that it should be in their interest to act as a "spider in the web" in a deeper level. This deeper control will cost them a lot in terms of short term costs, but might not be expensive in the long run.

"It is expensive and extensive for us to participate in the process as an Integrator." (TRV)

An increased participation in the suppliers process by TRV will in the end, probably save them a lot of money, even though this role would costs them a lot of resources. Especially a role like

Integrator, where TRV has the opportunity to be the 'Spider in the web' as they are expected to be. It is important in this case that the roles are divided between involved parties in beforehand, since in most of the projects the final costs are decided in beforehand. If the supplier is taking a role as Integrator, they will include costs for that, in the total cost of the order. Additionally, TRV will as a result of their participation have an easier access to relevant documentations and will be able to control whether or not the standards are used as TRV wants them to be. It can be pointed out that it has been seen in this study, that the supplier is following the standard. However, the question is whether or not TRV would like them to use it the way they are or *are there parts of the standard that could be used in an other way to their advantage?* Once again, *How much should TRV be involved in the work of the supplier?* Today, a revision is done from TRV every other year, in order to see whether or not the supplier is following the standard as they should and the way TRV want them to do. *Is this revision every other year the best way to proceed? How deep relation can an orderer and a supplier have?*

One of the most common comments from the supplier regarding the changing process from 2001 to 2011 version, is:

"I think we had to add some information, but not any extensive work." (Supplier)

Many requirements are added to the existing methods/techniques. Since the existing methods/techniques already were used by the supplier, the added requirements were not experienced to be extensive. Prototyping, Release note, Configuration management are examples of methods/techniques that was used by the supplier, and has been extended with requirements. Rollback procedure, Diagnostic information, embedded self-identification mechanisms are examples of methods/techniques that are new to this standard, but not to the supplier. These new methods/techniques in EN 50128:2011 are added in order to make these methods/techniques requirements in the standard. This is due to the fact that these methods/techniques might not be as obvious to new users, as they are to experienced software developer. Additionally, this is a sign of how dated the standard is. It is also a sign of that the methods/techniques in this standard are tested before they are added to the standard, which is positive. Future improvements within software development will probably be included in a future version of this standard.

"Our Validators and Assessor have thought about that even before we started to use the 2011 version of this standard." (Supplier)

Overall, many of the new documentations and procedures in the 2011 version of this standard have more or less been implemented to the process by the supplier. The Assessor is often a part of the maintenance process of a standard, and is when something well functioning is decided, implementing it to the process for the supplier. This way the working process are continuously developed, and will therefore, avoid a heavy process of updating the standard. The supplier are commenting on the fact that they are having many products in a continuous updating process, in order to always have the most modern version of tools, processes, documents etc. Due to this continuous development of their processes, the work of estimate the exact amount of work and resources spent in order to update their process to the new version, has been hard to do.

The fact that an Assessor are supporting the process of a user of the standard, might be seen as a good solution, as this makes the user in the forefront of the software development. *Is this the way the relationship between the user and the Assessor is intended to be according to the standard?* Like many other parts of the standard, this is interpretable. The responsibility of the Assessor is to make sure that the user are following the standard, so they should of course help them if necessary, but there is also an other side to this. *Is a person that is supporting a process as neutral as they should be? And what is neutral enough?* This question is like this standard, translatable. It can be translated in many ways, which can have positive and negative sequence effects.

All users of a standard, should have the will and the pursuit of participate in the maintenance of the specific standard. Today, this is not in the interest of many companies. They find it to be waste of resources to participate, but are in fact missing the opportunity to affect the standard to their advantage, and thus save money. Additionally, the standardization work could be done more efficiently than it is today, if perhaps less travel and more meetings over e.g. Skype could be arranged. The participants of the standardization working groups are mainly delegates from authorities, the users of the standard are missing [43]. From these persons perspective, the standard should make the system/product as safe as possible, not a lot of focus might be on whether the changes will make the process expensive to implement, or whether or not the changes will make the companies profits larger. It is up to the users of the standard to participate in the standardization work. The users can affect the content of the standard in their favor, and due to this save a lot of resources, for them selves as well as for the orderer of their products.

Participation from all sectors of society gives standards users confidence that standards reflect not only the scientific and technical state of the art, but that they also take into consideration the concerns and priorities of wider society.

A conclusion that can be made from this, is that the highest quality of the standards might be created as the working groups equal represented by end-users and authorities, and similar.

When a maintenance work are about to start, the experts attending the work should not be the only one adding comments and suggestions. Questions should be sent to the users of the standard, even if they are not involved in the standardization processes, in order to get comments from companies/persons who deals with the requirements every day. Many of the experts in the area are having roles as an Assessor, a role that is included in the process, but the process is seen from an other angle. All angles of the process dealing with a standard should be included in the maintenance work of a standard.

"The role of the Validator in the 2011 version of this standard has been discussed. The role is too extensive and are to much alike the role of the Assessor." (Supplier)

The role Validator seems to be a proof of missing end-users in the work of maintaining a standard. The responsibilities of the role Validator is extended, with the purpose of having a better control of the whole process. By the supplier, the responsibilities of the Validator was experienced to be to much like the responsibilities of the Assessor. They were indicating that it might be unnecessary to have two roles, with almost the same responsibilities.

How much money have they spent to update their process to follow the 2011 version? (Question 2) This question has been surprisingly hard to find answers too. As questions regarding this were asked TRV, they were sure about that a lot of money was spent on different parts of the changing process, but the exact amounts are not specified in details. *Perhaps it would be interesting to document more detailed information like this in the future?* The costs of a project is agreed on in beforehand. Therefore, it would have been interesting if the costs agreed of for the software development, in particular the costs agreed of for this standard update would have been documented. The changing process might have cost the supplier more than they expected or vise versa. TRV might have paid more than they should have had.

The changing process took approximately 500 hours per product to execute, which is calculated in the result to be a cost of approximately 1 000 000 SEK, to share between the companies that are a orderer of these products. As mentioned in the result, this would have been the case if the work would have been invoiced afterwards, but this is not the case. It is impossible to see the exact sum that has been spent in beforehand on this change in standard, from TRV. Next time a standard change will be done within a process, *It would be interesting to see how much that is exactly spent on this in beforehand!?* Most of these resources was spent on the tool classification process, approximately 160 hours. As is documented in the results, if these hours would be invoiced afterwards, it would result in a cost of 320 000 SEK. This cost would be divided between the orderer, approximately 10, which would result in a cost for TRV of approximately 32 000 SEK. If the supplier was not the large organization as they are, the costs due to this update process would probably increase. This is especially the case of the changes in the organizational part of the standard.

What parts of the process has been the most extensive and expensive to change due to this update in standard? (Question 3) As can clearly be seen in the result of this study, the classification of tools was the most extensive part of the changing process. Other parts of the process that were experienced to be hard due to the change from 2001 to 2011 version was the update of the organization, responsibilities and the SIL classification.

"The tools could have been more structured from the beginning." (Supplier)

The tool classification was, as mentioned above, the most extensive part of this standard changing process. The supplier experienced this part of the 2011 version to be a good addition to the standard. They said that they should have had a better structure of their tools before this update in standard. Resources spent on this well needed update work, could be seen as resources well spent. Clear examples added to the suggested guide (more about this later on) of this standard could result in decreased costs and even a better result of the tool classification.

The supplier also mentioned that they were waiting with the update in standard for their process. The supplier indicates that the decision of waiting was intentionally, since they were unsure about the outcome. As this subject is discussed with involved persons from TRV, they seem to all agree of the fact that the supplier waited too long with this change. The agreements between the two companies indicated that the 2011 version of this standard has to be followed before the old one expires. This could also be connected to the fact mentioned earlier, the orderer and the supplier are taking care of their case and not the other ones. As a result of this, TRV could only wait and make sure that the supplier was following the 2011 version, the date when the 2001 version expires, which they were.

"We want and need to use 'Formal methods' on all of our products. This is a good way of testing all the logic in the code." (Supplier)

To add methods like Formal methods to the system is good in a safety aspect. The supplier indicates that they will add Formal methods to their system, since this is a good way of testing all the logic in their codes. Involved persons in TRV was surprised as this was mentioned. It is not a requirement in the standard that the user has to use Formal methods, only a recommendation. This is an example of where it might be good for the orderer to be involved in a more technical way in order to be able to influence the methods/techniques used in the process.

"This European standard lacks in examples." (Supplier)

"An effective standard is one that should help developers, Assessors and users of such systems. For developers the standard should help them build the system cost-efficient, and it should be clear what is required in order to conform to the standard." [10]

Even though this text was written almost 20 years ago, this do not seem to be the case for all standards. 'Help' and 'clear' is keywords in this text and this seems to be what the users of this standard are missing and are longing for.

A descriptive, additional standard could be a suitable addition to this standard, in order for it to be easier to understand and follow. Many standards are divided into more than one part in order to cover different parts within the same area, e.g. EN 50129-1, EN 50129-2 and EN 50129-3. Since EN 50128:2011 has the recurrent lack of examples and intuitive explanations, an additional edition would be a simple and easy implemented solution to this problem. This additional edition could work as a guide of how to use the main part of this standard. In this descriptive guide to EN 50128:2011, topics that e.g. could be described are:

- The tool classification process - why/how should this process be executed, intuitive examples of how the description for each tool should be executed (like the example in figure 4.1), a list of tools that should be in each class and why etc.;

- SIL - purpose, meaning of the different levels, how to perform a SIL study etc;
- The long requirements in this standard could be shortened and referred to a descriptive part of the additional edition, where the descriptions could be even longer, in order to avoid misunderstandings;
- A more deeply explanation of the contents of the different lifecycle documents. The current explanations could be moved to this additional edition and extended with additional information;
- Illustration and information of the V-model, Document control summary and the Waterfall model;
- Examples of how to execute checklists in different scenarios. Even examples of how the checklists could look like and how to make the checklists work in the process;
- Annex B, C and D should also be moved to this explanatory part.

If relevant examples are included in an other standard, there should be no point of adding the same example, but EN 50128 should include a reference to that standard and the specific chapter where the example are located in that standard. Without examples, mistakes, confusions, time consuming and costly procedures could be the outcome.

"A product that is classified as SIL 0, is a product that has a good quality, but does not have any safety impact." (Supplier)

Is SIL the best way of handling safety related products? SIL was a large question mark when this standard was maintained. Basically, because of the changed meaning of SIL 0. The first mistake made was that the meaning of SIL 0 was changed, but the name was not. In the 2001 version, SIL 0 products did not have any safety impact, but in the 2011 version SIL 0 products has the lowest level of safety impact (SIL 1 light). This led to a large dissatisfaction, especially for the users that worked with SIL 0 products. They asked the question many times: *Why should we deal with non safety products in a safety related software standard? Why do we need all of these new requirements for a non safety related software?* The outcome of this confusion was that many of the Assessors, users and developer where misunderstanding the meaning of SIL 0, and the dissatisfaction was large. This resulted in a new standard, EN 50657. As mentioned earlier, this standard will adapt EN 50128:2011 for the application of the Rolling stock domain. Whether or not it is the correct way to go as a dissatisfaction appears is a discussion on its own. A conclusion to this is that, this standard has to be clearer and has to include more examples, in order for it to work as the effective standard it should be.

"Maybe the requirements and recommendations for the methods/techniques should not only depend on the SIL. There is other aspects such as size of the component/system etc. to take into account." (Experts)

Since both suppliers and experts are misunderstanding the meaning of SIL 0, and at some points even the other SIL as well, *is this really the safest way to go?* The basic for a safe system, should be that everybody understand how to use it. Maybe the recommendations for techniques and methods should depend on other aspects than only SIL. A large and complex software component might not need the same recommendations as a small and simple software component, both of the same SIL. *Should these products have the same recommendations?*

"It is hard to define whether the maintenance is major or minor." (Experts)

This is an other example of where the experts creating the standard does not understand the meaning of definitions. As new definitions are created, the meaning has to be clear documented, together with clear examples of how to deal with that specific part of the standard.

5.2 Analysis of the method

Since this study is based on experiences and thought, the technique of in depth interviewing with an open end, has been an efficient way of studying this subject. It was efficient to be able to send additional questions after the official interviews.

Persons of interest was contacted through e-mail. If possible, 'face-to-face' meetings was scheduled. These meetings were preferable since they were the most effective way to get the required information, with additional personal thoughts and experiences. Questions could be answered over e-mail, but in many times the meaning was mistaken, and most importantly, thoughts and experiences were not included in these answers.

Many times when the snowball selection was used, at least one person with the relevant knowledge was reached. In some cases, the chain ended with persons who was expected to have information, but they did not.

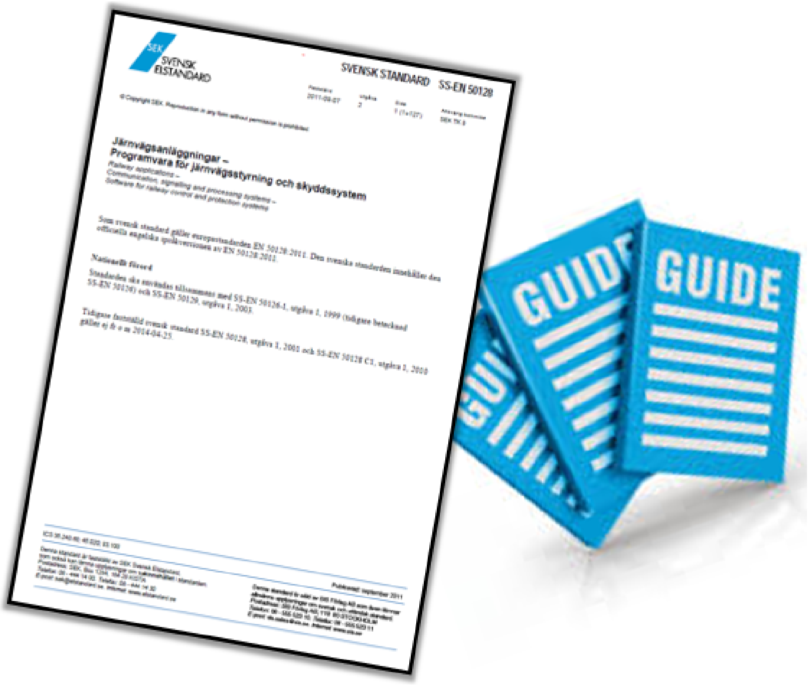
The majority of the persons contacted, did answer in a short period of time, with a friendly and helpful attitude. In some cases, the person contacted did not have the specific area of knowledge, but they tried to help anyway. If the person who was contacted, was recommended from an other person, this person was always added as a reference in the e-mail.

Many, short, unstructured interviews has been done a well, especially in the end of this study. These were executed in order to confirm that the solutions and conclusions of this study was correct. In some cases, the persons being interviewed had different answers and thoughts. The questions then had to be deeper researched in order to find the true answers. Some questions did not seem to have an answer. In these cases, the questions were formed into a suggestion, future research or remained to be unsolved.

The participation in an international working group has been an excellent opportunity to reach out to persons with good knowledge in the area. This has also enriched the network for the future. The international meetings and also the lectures at Svensk elstandard (SEK), has contributed to a wide knowledge in the subject and a joy to continue studying and working in this area.

6 Discussions

From my point of view, it seems to be a lack in the communication in the technical level of the process. *How can this lack in 'technical' communication between the supplier and TRV be improved?* Many involved persons, that I have been in contact with does not see this as a problem, while other do see this as a problem. There is of course not an easy solution to this, and as been discussed earlier: *How much should the orderer be involved in the work of the supplier?* I would say, definitely more than today, for the best of all involved parties. I think that a good solution for TRV would be to create a guide of how they want the standards to be used, when they are ordering a product. TRV should in this guide require specific methods/techniques/documentations in areas where it affects them, e.g. when and how to execute overall tests, integrations, relevant and important documentations etc. Methods/techniques, languages etc. for the pure coding should be left for the supplier to handle. The guide of how to use a standard would make it easier to see which areas that will require a lot of resources and which parts that will not. The guide will require some resources in order to be created, but as this is done, it will probably save time and money for all involved parties. Additionally, the revisions can be made in an easier way and costs due to this standard can be easier to calculate.



SEK SVENSK ELSTANDARD

SVENSK STANDARD SS-EN 50128

Fastställd 2011-09-07 Utgåva 2 Sida 1 (1+127) Ansvarig kommitté SEK TK 9

© Copyright SEK. Reproduction in any form without permission is prohibited.

Järnvägsanläggningar –
Programvara för järnvägsstyrning och skyddssystem

Railway applications –
Communication, signalling and processing systems –
Software for railway control and protection systems

Som svensk standard gäller europastandarden EN 50128:2011. Den svenska standarden innehåller den officiella engelska språkversionen av EN 50128:2011.

Nationellt förord

Standarden ska användas tillsammans med SS-EN 50126-1, utgåva 1, 1999 (tidigare betecknad SS-EN 50126) och SS-EN 50129, utgåva 1, 2003.

Tidigare fastställd svensk standard SS-EN 50128, utgåva 1, 2001 och SS-EN 50128 C1, utgåva 1, 2010 gäller ej fr o m 2014-04-25.

ICS 35.240.60; 45.020; 93.100

[illegible]

Publicerad: september 2011

[illegible]

Figure 6.1: A guide of how this standard should be used, is a good addition to the standard.

From my experience of this study, it seems like the orderer knows to little of the technical part of the process. I.e. the technical knowledge should be expanded within the organization. There should be a process where the supplier and the orderer works closer together. Of course, as mentioned above, TRV should not be involved in parts where they do not know more than the supplier and vise versa. This would be easier to perform if TRV was the only customer of the products, but that is not the case. Even if this is the case, I believe that a closer relation than it is today would be good.

Furthermore, I believe that it would be good if more companies were included in the standardization process thus the standard would be seen from other eyes as well. Today, as mentioned earlier, it is more or less the same persons that are involved in all of the railway standardization work. I know that it is open(!) to join the work of standardization, but I am experience it to not be many persons who knows about this. It is, at least in Sweden, a closed group of people working with the railway standardization. A suggestion would be for TRV to invite the supplier to the standardization process, in order for the two parties to participate together. This way a band can be connected between the two parties and at the same time, the standardization work will be done in the favor of both parties.

The standardization has to be done more efficiently in order to get more stakeholders involved in the process. I think that the frequent traveling to the meetings might keep interested stakeholders away from participating. Of course gatherings have to be held, but more meetings could be executed over Skype or similar. A start to include more stakeholders is to send out the commenting sheets etc. to possible participants, even outside the national committees. I think that this will increase the interest to participate. This might result in a better quality of the standards.

It would be an easy solution to make all SIL 1 and SIL 2 into SIL 2, due to the earlier mentioned confusion of the differences. The only differences for software is that SIL 2 product allows a smaller mistake range and it can affect the safety of humans. SIL 1 products does only affect the safety on the hardware. If all SIL 1 products/processes would become a SIL 2 product/process, this would not make any difference in the working process with the software, but it would result in a safer system. The same is valid for SIL 3 and SIL 4, I.e all of these products should be SIL 4. My impression is that it is too confusing with five levels of SIL. Three would probably be better in this case. The problem in this case is that the software has to execute and communicate through hardware, which has stricter differences of the different SIL. It is advantageously to have the same SIL on the software and the hardware. As this is the case, it would be preferable to add the different SIL for hardware in the describing additional part to this standard, in order for the different levels of SIL for software to get a better purpose.

Is the SIL classification done correctly? How can we be sure about that? And what if it is done the wrong way? The way of performing a SIL study is complicated. Therefore, I believe that these questions are important to think about. SIL is a way of separating products that are affecting the safety differently. Everything that is very complicated will in some cases not be performed as it should. This is my experience, so a new way which is easier to perform and to understand might be a good idea. I do not see the solution to this now, but I can see that it would be preferable to think about this question. However, my suggestion mentioned above, I.e. to reduce the SIL levels to three, would most certainly also make the classification easier.

7 Conclusions

The purpose of this study was to document how this update of the standard EN 50128, affected TRV and their suppliers. The work of changing the version of the standard for the supplier took approximately 500 hours. The tool classification took approximately 160 hours of these 500 hours, so this was clearly the most extensive part of this update in standard. Since the cost for this standard update has been added to the total costs for software developing and invoiced in before hand, it has not been possible to see whether or not TRV has payed more or less than these calculated hours.

The difference in content between the 2001 and 2011 version was considered to be quite extensive, since it has almost the double amount of requirements, compare to the 2001 version of this standard. However, the update in the process of the supplier was not experienced to be as extensive as expected. Conclusions to this, is that many of the new requirements, including methods/techniques, had already been used in the process for a long time, but had not earlier been requirements. The Assessor was introducing new methods/techniques, documentations etc. to the supplier before the 2011 version was used.

The lack of examples, explanation and the unclear parts of the standard have to be clarified. The unclear parts of this standard is experienced to be SIL, major and minor maintenance etc. Suggested solutions to these problems have been introduced in this study:

- Additional describing part of EN 50128 - EN 50128 should be more clear if an additional describing part was added. This part should be used as a guide of how to use the standard and should include examples, definitions, descriptions and illustrations etc.
- A guide of how the orderer want the standard to be used - In this guide, the parts that are in the interest of the orderer, should be described, such as overall tests, integration, documentations etc.
- More users in the standardization working groups - The work of maintaining and creating a standard should include more users than today. It is important to include all parts of the process, a good and complete standard can not be created without involving all the users of the standard.

The supplier was positive to the extended explanations of the roles and the related responsibilities. They had all of the required roles within the organization, but in some cases they had to share the same role for different products within the organization. A conclusion from this is that the version update was not experienced, by the supplier, to be an extensive work. If the supplier was a small company, and a newer software developer, the version update would probably be experienced to be more extensive.

In summary, it can be said that some parts of the 2011 version was experienced as extensive, but overall not a lot of changes had to be done to the process.

"I think we had to add some information, but not any extensive work" (Supplier)

It is also clear from this study that some parts of the standard has to be clarified, in order to save resources and make sure that no mistakes are made. Since this standard is handling safety related software, clearance is extremely important.

7.1 Future research

In order to get an even wider knowledge on the subject, it would be interesting, if the same study could be done with more suppliers/developers, in order to get a wider picture on how the standard could be interpreted. A way of taking this study forward, would also be to see where and how the standard could be closed for dangerous interpretations. The standard is presently open to many interpretations. *Which interpretations are classified as dangerous and which interpretations are approved from a safety point of view?* An incorrect interpretation could lead to catastrophic situation. For a standard like EN 50128, that are dealing with safety related software, it is of course extremely important that interpretation possibilities are reduced to a minimum! Risk analysis is done on systems, but *has it been done to a standard?* A standard is created and then right away used in a process. *Are there enough studies made on how the standard are working in the reality? What would be enough in this case?*

A study of the integration between standards, also for the integration of different products, would be interesting to execute. *Is the integration between the different products and organizations safe enough?* The companies might be good in handling the safety for their own product, but *who is responsible for the parts in the lines between different products?* I have discovered in this study that, it is possible to find areas where the standard is not fulfilled to the fullest, especially in between phases and products. It seems that all parties are following the standard as complete as they can. *But who are responsible for the standard in between products?* In a development of a new system, it is important that all parts are covered in terms of safety and functionality in order for the system to be successful.

The supplier and the experts interviewed in this study, has all experiences in the area of software development and standardization procedures. It would be interesting to extend this study with experiences and thoughts from new users of this standard and from other perspectives as well. New perspectives could come from persons that are in contact with other standards. They might experience this update in version as a much more extensive process, than the experienced users. Additionally, they might with ”new eyes”, see limitations and mistakes in the process which the involved experts have missed.

7.2 Evaluation of the study

The goal was to study the affect of the update of EN 50128. The study intended to include costs and time of the work with this update. TRV requested the result of this, from their point of view. This has been hard to execute, since consumed time and costs have not been clearly and separately accounted. Therefore, the focus had to be changed. The outcome of the study is basically focused on the experiences of this update in version, parts that were experienced as extensive and the approximately timed consumed on the most extensive parts of the process. New interesting questions and conclusions have been added along the way of this study, e.g. lack of specific cost reports, standardization processes and the communication between involved parties.

The study was hard to execute, since the 2011 version of this standard are new to the working process of the supplier. Furthermore, it has been hard to find persons with knowledge in the subject. There has not been much written in the subject and it has been time consuming to sort out interesting areas in the documentations found close to this specific subject.

In the recurring part of the case, I.e. when the same specific question was asked to different persons, different answers and thought were received. This made the outcome more interesting, but also harder to evaluate and thus create a result based on the received answers.

The study gave me a good knowledge in the subject, which I will use in future work in this and similar areas. The time of the study has made me realize that this continuously developing subject is an interesting and exiting area.

References

Books

[1] Jean-Louis Boulanger. *CENELEC 50128 and IEC 62279 Standards.* Wiley-Iste, 2015.

[2] J W. Creswell. *Research Design - Qualitative, Quantitative, and Mixed Methods Approaches,* vol 3. California: SAGE, 2009.

[3] A. Tjora. *Från nyfikenhet till systematisk kunskap - Kvalitativ forskning i praktiken..* Lund, Sweden: Studentlitteratur, 2010.

[4] A. Bryman. *Samhällsvetenskapliga metoder,* vol 2. Stockholm, Sweden: Liber, 2011.

Articles

[5] L. Eriksson, ”Hela resan, Så funkar vi i förändringar - en intervju med Anna Almberg”, 2011.

Reports

[6] Henrik Thane, *GAP-Analys EN50128 2001-2011 CENELEC EN50128:2001 vs EN50128:2011*, Safety Integrity AB, Sweden, 2015.

[7] *GB Involvement in the Development of Euro norms in the Field of Railway Applications - CENELEC TECHNICAL COMMITTEE TC9X*. RSSB and bsi. June 2015.

[8] IEC/CENELEC, ”IEC 62279: Railway applications – Communications, signalling and processing systems – Software for railway control and protection systems (EN 50128)”, 2008.

[9] R. Sahlberg and others. *Så skapar vi framtidens attraktiva kollektivtrafik* Trafikverket and VINNOVA. Karlstad, 2012.

[10] N. E. Fenton and M. Neil. *A strategy for Improving Safety Related Software Engineering Standards*, vol 24. IEEE, 1998.

[11] railwaysignalling.eu, ”The ERTMS/ETCS signalling system - An overview on the Standard European Interoperable signalling and train control system”, 2014.

[12] Fisher/Emerson Process Management, ”Basic Fundamentals Of Safety Instrumented Systems”, 2005

[13] Ansaldo STS, ”The Global Leading ERTMS”.

Standards

[14] *EN 50128:2001*, CENELEC, 2001.

[15] *EN 50128:2011*, CENELEC, 2011.

[16] *EN 50657:2016 draft*, CENELEC, 2016.

[17] *EN 50126:1999*, CENELEC, 1999.

[18] *EN 50129:2003*, CENELEC, 2003.

Online Sources

[19] HEICON. (2015, September 10). *EN 50128 – Functional Safety in the railway industry* [Online]. Available: http://blog.heicon-ulm.de/en-50128-functional-safety-in-the-railway-industry. Visited: 2016-12-02.

[20] Wikipedia. (2016) *Standard* [Online] Available: https://sv.wikipedia.org/wiki/Standard. Visited: 2017-02-12.

[21] SIS. (2016). *Guide till standarder och standardisering* [Online]. Available: http://www.sis.se /tema/iforvaltning/guide/. Visited: 2017-02-12.

[22] Wikipedia. (2014, May 31) *Standardisering* [Online]. Available: https://sv.wikipedia.org/wiki /Standardisering. Visited: 2017-02-15.

[23] SEK Svensk Elstandard. (2017) *Om SEKs katalog* [Online]. Available: http://www.elstandard .se/standarder/index.asp. Visited: 2017-02-15.

[24] Boverket. (2015, December 7) *Olika typer av standarder* [Online]. Available: http://www.boverket.se/sv/byggande/vagledning-om-standarder/vad-ar-standarder/olika-typer-av-standarder/. Visited: 2017-02-15.

[25] Wikipedia. (2016) *European Committee for Electrotechnical Standardization* [Online]. Available: https:// en.wikipedia.org/wiki/European_Committee_for_Electrotechnical_Standardization. Visited: 2017-02-16.

[26] CENELEC. (2017) *Who we are* [Online]. Available: https://www.cenelec.eu/aboutcenelec /whoweare/index.html. Visited: 2017-03-07.

[27] Olle Vejde. *Snöbollsurval* [Online]. Available: http://www.ollevejde.se/statistikord/snobollsurval.htm. Visited: 2017-03-22.

[28] Icarus-ORM - Operational Risk Management. (2015, 19 Mars) *What is SIL? A crash course* [Online] Available: https://www.youtube.com/watch?v=Af-CbZ7aTCY.

[29] Market Research Guy. (2011, 10 October). *My Market Research Methods* [Online] Available: http://www.mymarketresearchmethods.com/quantitative-vs-qualitative-research-whats-the- difference/. Visited: 2017-03-30.

[30] Wikipedia. (2017, 2 February) *Qualitative research* [Online]. Available: https://en.wikipedia.org /wiki/Qualitative_research. Visited: 2017-03-29.

[31] Wikipedia. (2017, 2 February) *Quantitative research* [Online]. Available: https://en.wikipedia. org/wiki/Quantitative_research. Visited: 2017-03-29.

[32] ASA - The society of Accredited Safety Auditors Limited *What are safety audit?* [Online]. Available: http://www.sasa.org.hk/audit.htm Visited: 2017-04-19

[33] Dr. D. O'Grady. (1994, 15 July) *The Change Game* [Online]. Available: http://www.drogrady. com/73/the-change-game/. Visited: 2017-05-08.

[34] Trafikverket. (2016, 26 April) [Online]. Available: www.trafikverket.se. Visited: 2017-05-09.

[35] T. Kivikas. (2011, 14 May) *Förändring möts alltid av motstånd* [Online]. Available: http:// kivikas.com/2011/05/forandring-mots-alltid-av-motstand/. Visited: 2017-05-12

[36] Wikipedia. (2017, 17 Mars) *Trafikverket (Sverige)* [Online]. Available: https://sv.wikipedia.org /wiki/Trafikverket_(Sverige). Visited: 2017-05-12

[37] Railway Pro - communication platform. (2016, 25 July) *Jernbaneverket divides ERTMS procurement in three packages* [Online]. Available: http://www.railwaypro.com/wp/jernbaneverket-divides-ertms-procurement-in-three-packages/. Visited: 2017-05-14

[38] Wikipedia. (2017, 5 May) *European Train Control System* [Online]. Available: https://en.wikipedia.org/wiki/European_Train_Control_System. Visited: 2017-05-14

[39] Engineering 360. (2017) [Online]. Available: http://standards.globalspec.com. Visited: 2017-05-15

[40] Thales. (2017) *European Train Control System (ETCS)* [Online]. Available: http://www.thalesgroup.com/en/european-train-control-system-etcs. Visited: 2017-05-20

[41] European comission - Mobility and Transport. (2017, 28 May) *ERTMS - Levels and Modes* [Online]. Available: https://ec.europa.eu/transport/modes/rail/ertms/what-is-ertms/levels_and_ modes_sl. Visited: 2017-05-28

Unpublished

[42] Sam Berggren, Tågordning och standardisering, unpublished.

[43] M. Fusani, "Can the efficacy of Standards for safety-critical software be improved?" System and Software Evaluation Center. Pisa, Italy, 2017.

8 Appendix

Documented data and comments are added to the Appendix. Tables (A), new requirements (B), questions from the interviews (C), the roles with responsibilities (D) and interesting documentations (E) are all presented in this section.

8.1 Appendix A - Tables

The tables are presented for product 1 and 2 with additional requirements and comments. The section is divided in two parts, first the listed tables, where details of whether or not parts of the tables are used, both for the 2001 and the 2011 version. In the second part, additional requirements to each table are presented including resulting comments.

The definitions of the different levels of recommendations are presented here, in order for the reader to easily understand the meaning of the tables. [15]

- 'M' - this symbol means that the use of a technique is mandatory,
- 'HR' - this symbol means that the technique or measure is Highly Recommended for this safety integrity level. If this technique or measure is not used then the rationale for using alternative techniques shall be detailed in the Software Quality Assurance Plan or in another document referenced by the Software Quality Assurance Plan,
- 'R' - this symbol means that the technique or measure is Recommended for this safety integrity level. This is a lower level of recommendation than an 'HR' and such techniques can be combined to form part of a package,
- '-' - this symbol means that the technique or measure has no recommendation for or against being used,
- 'NR' - this symbol means that the technique or measure is positively Not Recommended for this safety integrity level. If this technique or measure is used then the rationale behind using it shall be detailed in the Software Quality Assurance Plan or in another document referenced by the Software Quality Assurance Plan.

The tables with associated levels of recommendations and with descriptions of whether or not the specific method/technique is used by that specific product or not, is presented in Appendix A, section 8.1. The different level of usage is marked with:

- 'yes' - The method/technique is used to the fullest for that product;
- 'partly' - The method/technique is partly used for the product;
- '-' - The method/technique is ether removed from this standard, or argued to not be used since the product does not have to be used;
- 'no' - The method/technique is not used by this product.

The numbering in the list represent the number in each table and version. Examples: (7,8) states that the technique/measure is number seven in that table in EN 50128:2001 and number eight in that table in EN 50128:2011. (-,2) states that the technique/measure does not exist in EN 50128:2001 and is number two in EN 50128:2011.

Product 1 SIL 3/SIL 4

Tables	2001	2011	2001	2011
A.1 Lifecycle Issues and Documentation				
Documentation				
Planning	HR			
Software Quality Assurance Plan (-,1)		HR		yes
Software Quality Assurance Verification Report (-,2)		HR		yes
Software Configuration Management Plan (-,3)		HR		yes
Software Verification Plan (-,4)		HR		yes
Software Validation Plan (-,5)		HR		yes
Software requirements	HR			
Software Requirements Specification (-,6)		HR		yes
Overall Software Test Specification (-,7)		HR		yes
Software Requirements Verification Report (-,8)		HR		yes
Architecture and Design				
S/W Design Documents (3,-)	HR			
Software Architecture Specification (-,9)		HR		yes
Software Design Specification (-,10)		HR		yes
Software Interface Specification (-,11)		HR		yes
Software Integration Test Specification (-,12)		HR		yes
Software/Hardware Integration Test Specification (-,13)		HR		yes
Software Architecture and Design Verification Report (-,14)		HR		yes
Component Design				
Software Component Design Specification (-,15)		HR		yes
Software Component Test Specification (-,16)		HR		yes
Software Component Design Verification Report (-,17)		HR		yes
S/W Module Documents (4,-)	HR			
Component Implementation and Testing				
Software Source Code and supporting documentation (5,18)	HR	HR		yes
Software Component Test Report (-,19)		HR		yes
Software Source Verification Report (-,20)		HR		yes
Integration				
Software Integration Test Report (-,21)		HR		yes
Software/Hardware Integration Test Report (7,22)	HR	HR		yes
Software Integration Verification Report (-,23)		HR		yes
S/W Test Reports (6,-)	HR			
Overall Software Testing/Final Validation				
Overall Software Test Report (-,24)		HR		yes
Software Validation Report (8,25)	HR	HR		yes
Tools Validation Report (-,26)		HR		yes
Release Note (-,27)		HR		yes
System configured by application data/algorithms				
Application Requirements Specification (-,28)		HR		partly
Application Preparation Plan (-,29)		HR		yes
Application Test Specification (-,30)		HR		partly
Application Architecture and Design (-,31)		HR		no
Application Preparation Verification Report (-,32)		HR		partly
Application Test Report (-,33)		HR		partly
Source Code of Application Data/Algorithms (-,34)		HR		partly
Application Data/Algoritms Verification Report (-,35)		HR		partly

Product 1 SIL 3/SIL 4

Software deployment				
Software Release and Deployment Plan (-,36)		HR		yes
Software Deployment Manual (-,37)		HR		partly
Release Notes (-,38)		HR		yes
Deployment Records (-,39)		HR		partly
Deployment Verification Report (-,40)		HR		partly
Software maintenance				
Software maintenance Plan (-,41)		HR		yes
Software Change Records (-,42)		HR		yes
Software Maintenance Records (10,43)	HR	HR		yes
Software Maintenance Verification Report (-,44)		HR		yes
Software assessment				
Software Assessment Plan (-,45)		HR		yes
Software Assessment Report (9,46)	HR	HR		yes
A.2 Software Requirements Specification				
Technique/Measure				
Formal Methods (1,1)	HR	HR		no
Semi-Formal Methods (2,-)	HR			
Structured Methodology (3,3)	HR	HR		yes
Modelling (A.17) (-,2)		HR		yes
Decision Tables (-,4)		HR		-
A.3 Software Architecture				
Technique/Measure				
Defensive Programming (1,1)	HR	HR		yes
Fault Detection & Diagnosis (2,2)	HR	HR		yes
Error Correcting Codes (3,3)	-	-		partly
Error Detecting Codes (4,4)	HR	HR		yes
Failure Assertion Programming (5,5)	HR	HR		partly
Safety Bag Techniques (6,6)	R	R		no
Diverse Programming (7,7)	HR	HR		no
Recovery Block (8,8)	R	R		no
Backward Recovery (9,9)	NR	NR		no
Forward Recovery (10,10)	NR	NR		no
Retry Fault Recovery Mechanism (11,11)	R	R		no
Memorising Executed Cases (12,12)	HR	HR		no
Artificial Intelligence - Fault Correction (13,13)	NR	NR		no
Dynamic Reconfiguration of software (14,14)	NR	NR		no
Software Error Effect Analysis (15,15)	HR	HR		partly
Graceful Degradation (-,16)		HR		no
Information Hiding (-,17)		-		yes
Information Encapsulation (-,18)		HR		yes
Fully Defined Interface (-,19)		M		yes
Formal Methods (-,20)		HR		no
Modelling (-,21)		HR		yes
Structured Methodology (-,22)		HR		partly
Modelling supported by computer aided design (-,23) and specification tools		HR		partly
Fault Tree Analysis (16,-)	HR			

Product 1 SIL 3/SIL 4

A.4 Software Design and Implementation				
Technique/Measure				
Formal Methods (1,1)	HR	HR		no
Modelling (A.17) (-,2)		HR		yes
Structured Methodology (3,3)	HR	HR		partly
Modular Approach (4,4)	M	M		yes
Components (A.20) (-,5)		HR		yes
Design and Coding Standards (A.12) (5,6)	M	M		yes
Analysable Programs (6,7)	HR	HR		yes
Strongly Typed Programming Language (7,8)	HR	HR		yes
Structured Pragramming (8,9)	HR	HR		yes
Programming Language (A.15) (9,10)	HR	HR		yes
Language Subset (10,11)	HR	HR		yes
Object Oriented Programming (A.22) (19,12)	R	R		no
Procedural Programming (-,13)		HR		yes
Metaprogramming (-,14)		R		no
Semi-Formal Methods (2,-)	HR			
Validated Translator (11,-)	HR			
Translator Proven in Use (12,-)	HR			
Library of Trusted/Verified Modules and Components (13,-)	R			
Functional/Black-box Testing (14,-)	M			
Performance Testing (15,-)	HR			
Interface Testing (16,-)	HR			
Data Recording and Analysis (17,-)	M			
Fuzzy Logic (18,-)	-			
A.5 Verification and Testing				
Technique/Measure				
Formal Proof (1,1)	HR	HR		no
Static Analysis (A.19) (3,2)	HR	HR		yes
Dynamic Analysis and Testing (A.13) (4,3)	HR	HR		yes
Metrics (5,4)	R	R		yes
Traceability (6,5)	HR	M		yes
Software Error Effect Analysis (7,6)	HR	HR		partly
Test Coverage for code (A.21) (-,7)		HR		yes
Functional/Black-box Testing (A.14) (-,8)		M		yes
Performance Testing (A.18) (-,9)		HR		partly
Interface Testing (-,10)		HR		yes
Probabilistic Testing (2,-)	HR			
A.6 (Software/Hardware) Integration				
Technique/Measure				
Functional and Black-box Testing (A.14) (1,1)	HR	HR		yes
Performance Testing (A.18) (2,2)	HR	HR		partly
A.7 2001: Software Validation, 2011: Overall Software Testing				
Technique/Measure				
Performance Testing (A.18) (2,1)	M	M		partly
Functional and Black-box Testing (A.14) (3,2)	M	M		yes
Modelling (A.17) (4,3)	R	R		partly

Probabilistic Testing (1,-)	HR			
A.8 2001: Clauses to be assessed				
S/W Safety Integrity Levels	HR		-	-
Personnel & Responsibility	HR		-	-
Lifecycle & Documentation	HR		-	-
S/W Requirements Specification	HR		-	-
S/W Architecture	HR		-	-
Design & Development	HR		-	-
Verification	HR		-	-
S/W/H/W Integration	HR		-	-
S/W Validation	HR		-	-
Quality Assurance	HR		-	-
Maintenance	HR		-	-
A.8 2011: Software Analysis Techniques				
Static Software Analysis (A.19)		HR		yes
Dynamic Software Analysis (A.13 + A.14)		HR		yes
Cause Consequence Diagrams		R		no
Event Tree Analysis		R		no
Software Error Effect Analysis		HR		yes
A.9 2001: Software Assessment				
Assessment Techniques				
Checklists	HR			
Static Software Analysis	HR			
Dynamic Software Analysis	HR			
Cause Consequence Diagrams	R			
Event Tree Analysis	R			
Fault Tree Analysis	HR			
Software Error Effect Analysis	HR			
Common Cause Failure Analysis	HR			
Markov Models	R			
Reliability Block Diagram	R			
Field Trial Before Commissioning	HR			
A.9 2011 = A.10 2001 Software Quality Assurance				
Technique/Measure				
Accredited to EN ISO 9001 (1,1)	HR	HR		yes
Compliant with EN ISO 9001 (-,2)		M		yes
Compliant with ISO/IEC 90003 (2,3)	M	R		yes
Company Quality System (3,4)	M	M		yes
Software Configuration Management (4,5)	M	M		yes
Checklists (-,6)		HR		yes
Traceability (-,7)		M		yes
Data Recording and Analysis (-,8)		M		yes
A.11 2001 = A.10 2011 Software Maintenance				
Technique/Measure				
Impact Analysis (1,1)	M	M	yes	yes

Data Recording and Analysis (2,2)	M	M	yes	yes
A.11 2011 Data Preparation Techniques				
Technique/Measure				
Tabular Specification Methods		R		-
Application Specific Language		R		-
Simulation		HR		-
Functional testing		M		-
Checklists		M		-
Fagan inspection		R		-
Formal design reviews		HR		-
Formal proof of correctness (of data)		HR		-
Walkthrough		HR		-
A.12 (Design and) Coding Standards				
Technique/Measure				
Coding Standard (1,1)	HR	M		yes
Coding Style Guide (2,2)	HR	HR		yes
No Dynamic Objects (3,3)	HR	HR		yes
No Dynamic Variables (4,4)	HR	HR		yes
Limited Use of Pointers (5,5)	R	R		yes
Limited Use of Recursion (6,6)	HR	HR		yes
No Unconditional Jumps (7,7)	HR	HR		yes
Limited size and complexity of Functions,				
Subroutines and Methods (-,8)		HR		yes
Limited number of subroutine parameters (-,9)		R		yes
Limited use of Global Variables (-,10)		M		yes
A.13 Dynamic Analysis and Testing				
Technique/Measure				
Test Case Execution from Boundary Value Analysis (1,1)	HR	HR		yes
Test Case Execution from Error Guessing (2,2)	R	R		yes
Test Case Execution from Error Seeding (3,3)	R	R		no
Performance Modelling (4,4)	HR	HR		partly
Equivalence Classes and Input Partition Testing (5,5)	HR	HR		yes
Structure-Based Testing (6,6)	HR	HR		yes
A.14 Functional/Black Box Test				
Technique/Measure				
Test Case Execution from Cause Consequence Diagrams (1,1)	R	R		no
Prototyping/Animation (2,2)	R	R		no
Boundary Value Analysis (3,3)	HR	HR		yes
Equivalence Classes and Input Partition Testing (4,4)	HR	HR		yes
Process Simulation (5,5)	R	R		yes
A.15 (Textual) Programming Languages				
Technique/Measure				
ADA (1,1)	R	HR		no
MODULA-2 (2,2)	R	HR		no
PASCAL (3,3)	R	HR		no

Product 1 SIL 3/SIL 4

C or C++ (6,4)	R	R	yes	yes
PL/M (7,5)	NR	NR		no
BASIC (8,6)	NR	NR		no
Assembler (9,7)	-	R		no
C# (-,8)		R		no
JAVA (-,9)		R		no
Statement List 12,10)	R	R		no
Fortran 77 (4,-)	R			no
C or C++ (unrestricted) (5,-)	NR			no
Ladder Diagrams (10,-)	R			no
Functional Blocks (11,-)	R			no
A.16 2001 = A.17 2011 Modelling				
Technique/Measure				
Data Modelling (-,1)		HR		no
Data Flow Diagrams (1,2)	R	HR		no
Control Flow Diagrams (-,3)		HR		yes
Finite State Machines or State Transition Diagrams (2,4)	HR	HR		yes
Time Petri Nets (5,5)	HR	HR		no
Decision/Truth Tables (-,6)		HR		no
Formal Methods (3,7)	HR	HR		no
Performance Modelling (4,8)	HR	HR		partly
Prototyping/Animation (6,9)	R	R		no
Structure Diagrams (7,10)	HR	HR		yes
Sequence Diagrams (-,11)		HR		yes
A.16 2011 Diagrammatic Languages for Application Algorithms				
Technique/Measure				
Functional Block Diagrams		R		-
Sequential Function Charts		HR		-
Ladder Diagrams		R		-
State Charts		HR		-
A.17 2001 = A.18 2011 Performance Testing				
Technique/Measure				
Avalanche/Stress Testing (1,1)	HR	HR		partly
Response Timing and Memory Constraints (2,2)	HR	HR		partly
Performance Requirements (3,3)	HR	HR		partly
A.19 Static Analysis				
Technique/Measure				
Boundary Value Analysis (1,1)	HR	HR		yes
Checklists (2,2)	R	R		yes
Control Flow Analysis (3,3)	HR	HR		partly
Data Flow Analysis (4,4)	HR	HR		partly
Error Guessing (5,5)	R	R		yes
Walkthroughs/Design Reviews (9,6)	HR	HR		yes
Fagan Inspections (6,-)	HR			
Sneak Circuit Analysis (7,-)	R			
Symbolic Execution (8,-)	HR			

A.18 2001 Semi-Formal Methods				
Technique/Measure				
Logic/Function Block Diags	HR			-
Sequence Diagrams	HR			-
Data Flow Diagrams	R			-
Finite State Machines/State Transition Diagrams	HR			-
Time Petri Nets	HR			-
Decision/Truth Tables	HR			-
A.20 2001: Modular Approach 2011: Components				
Technique/Measure				
Information Hiding (-,1)		-		yes
Information Encapsulation (-,2)		HR		yes
Information Hiding/Encapsulation (2,-)	HR			
Parameter Number Limit (3,3)	R	R		no
Fully Defined Interface (5,4)	M	M		yes
Module Size Limited (1,-)	HR			
One Entry/One Exit Point in Subroutines and Functions (4,-)	HR			
A.21 2011 Test Coverage for Code				
Test coverage criterion				
Statement		HR		yes
Branch		HR		yes
Compound Condition		HR		yes
Data flow		HR		no
Path		HR		no
A.22 2011 Object Oriented Software Architecture				
Technique/Measure				
Traceability of the concept of the application domain to the				
classes of the architecture		HR		-
Use of suitable frames, commanly used combinations of				
classes and design patterns		HR		-
Object Oriented Detailed Design		HR		-
A.23 2011 Object Oriented Detailed Design				
Technique/Measure				
Classes should have only one objective		HR		-
Inheritance used only if the derived class is a refinement				
of its basic class		HR		-
Depth of inheritance limited by coding standards		HR		-
Overriding of operations (methods) under strict control		HR		-
Multiple inheritance used only for interface classes		HR		-
Inheritance from unknown classes		NR		-

Product 2 SIL 3/SIL 4

Tables	2001	2011	2001	2011
A.1 Lifecycle Issues and Documentation				
Documentation				
Planning	HR		yes	
Software Quality Assurance Plan (-,1)		HR		yes
Software Quality Assurance Verification Report (-,2)		HR		yes
Software Configuration Management Plan (-,3)		HR		yes
Software Verification Plan (-,4)		HR		yes
Software Validation Plan (-,5)		HR		yes
Software requirements	HR		yes	
Software Requirements Specification (-,6)		HR		yes
Overall Software Test Specification (-,7)		HR		yes
Software Requirements Verification Report (-,8)		HR		yes
Architecture and Design				
S/W Design Documents (3,-)	HR		yes	
Software Architecture Specification (-,9)		HR		yes
Software Design Specification (-,10)		HR		yes
Software Interface Specification (-,11)		HR		yes
Software Integration Test Specification (-,12)		HR		yes
Software/Hardware Integration Test Specification (-,13)		HR		yes
Software Architecture and Design Verification Report (-,14)		HR		yes
Component Design				
Software Component Design Specification (-,15)		HR		yes
Software Component Test Specification (-,16)		HR		yes
Software Component Design Verification Report (-,17)		HR		yes
S/W Module Documents (4,-)	HR		yes	
Component Implementation and Testing				
Software Source Code and supporting documentation (5,18)	HR	HR	yes	yes
Software Component Test Report (-,19)		HR		yes
Software Source Verification Report (-,20)		HR		yes
Integration				
Software Integration Test Report (-,21)		HR		yes
Software/Hardware Integration Test Report (7,22)	HR	HR	yes	yes
Software Integration Verification Report (-,23)		HR		yes
S/W Test Reports (6,-)	HR		yes	
Overall Software Testing/Final Validation				
Overall Software Test Report (-,24)		HR		yes
Software Validation Report (8,25)	HR	HR	yes	yes
Tools Validation Report (-,26)		HR		yes
Release Note (-,27)		HR		yes
System configured by application data/algorithms				
Application Requirements Specification (-,28)		HR		partly
Application Preparation Plan (-,29)		HR		yes
Application Test Specification (-,30)		HR		partly
Application Architecture and Design (-,31)		HR		no
Application Preparation Verification Report (-,32)		HR		partly
Application Test Report (-,33)		HR		no
Source Code of Application Data/Algorithms (-,34)		HR		partly
Application Data/Algoritms Verification Report (-,35)		HR		partly

Software deployment				
Software Release and Deployment Plan (-,36)		HR		yes
Software Deployment Manual (-,37)		HR		partly
Release Notes (-,38)		HR		yes
Deployment Records (-,39)		HR		partly
Deployment Verification Report (-,40)		HR		partly
Software maintenance				
Software maintenance Plan (-,41)		HR		yes
Software Change Records (-,42)		HR		yes
Software Maintenance Records (10,43)	HR	HR	yes	yes
Software Maintenance Verification Report (-,44)		HR		yes
Software assessment				
Software Assessment Plan (-,45)		HR		yes
Software Assessment Report (9,46)	HR	HR	yes	yes
A.2 Software Requirements Specification				
Technique/Measure				
Formal Methods (1,1)	HR	HR	yes	yes
Semi-Formal Methods (2,-)	HR		yes	
Structured Methodology (3,3)	HR	HR	partly	yes
Modelling (A.17) (-,2)		HR		yes
Decision Tables (-,4)		HR		-
A.3 Software Architecture				
Technique/Measure				
Defensive Programming (1,1)	HR	HR	yes	yes
Fault Detection & Diagnosis (2,2)	HR	HR	yes	yes
Error Correcting Codes (3,3)	-	-	no	no
Error Detecting Codes (4,4)	HR	HR	yes	yes
Failure Assertion Programming (5,5)	HR	HR	yes	partly
Safety Bag Techniques (6,6)	R	R	no	no
Diverse Programming (7,7)	HR	HR	yes	yes
Recovery Block (8,8)	R	R	no	no
Backward Recovery (9,9)	NR	NR	no	no
Forward Recovery (10,10)	NR	NR	no	no
Retry Fault Recovery Mechanism (11,11)	R	R	yes	no
Memorising Executed Cases (12,12)	HR	HR	yes	no
Artificial Intelligence - Fault Correction (13,13)	NR	NR	no	no
Dynamic Reconfiguration of software (14,14)	NR	NR	-	no
Software Error Effect Analysis (15,15)	HR	HR	-	partly
Graceful Degradation (-,16)		HR		no
Information Hiding (-,17)		-		yes
Information Encapsulation (-,18)		HR		yes
Fully Defined Interface (-,19)		M		yes
Formal Methods (-,20)		HR		partly
Modelling (-,21)		HR		yes
Structured Methodology (-,22)		HR		partly
Modelling supported by computer aided design (-,23)		HR		partly
and specification tools				
Fault Tree Analysis (16,-)	HR		-	

Product 2 SIL 3/SIL 4

A.4 Software Design and Implementation				
Technique/Measure				
Formal Methods (1,1)	HR	HR	yes	partly
Modelling (A.17) (-,2)		HR		yes
Structured Methodology (3,3)	HR	HR	yes	partly
Modular Approach (4,4)	M	M	yes	yes
Components (A.20) (-,5)		HR		yes
Design and Coding Standards (A.12) (5,6)	M	M	yes	yes
Analysable Programs (6,7)	HR	HR	no	yes
Strongly Typed Programming Language (7,8)	HR	HR	yes	yes
Structured Pragramming (8,9)	HR	HR	yes	yes
Programming Language (A.15) (9,10)	HR	HR	yes	yes
Language Subset (10,11)	HR	HR	yes	yes
Object Oriented Programming (A.22) (19,12)	R	R	differ	no
Procedural Programming (-,13)		HR		yes
Metaprogramming (-,14)		R		no
Semi-Formal Methods (2,-)	HR		yes	
Validated Translator (11,-)	HR		no	
Translator Proven in Use (12,-)	HR		yes	
Library of Trusted/Verified Modules and Components (13,-)	R		yes	
Functional/Black-box Testing (14,-)	M		yes	
Performance Testing (15,-)	HR		yes	
Interface Testing (16,-)	HR		yes	
Data Recording and Analysis (17,-)	M		yes	
Fuzzy Logic (18,-)	-		no	
A.5 Verification and Testing				
Technique/Measure				
Formal Proof (1,1)	HR	HR	yes	yes
Static Analysis (A.19) (3,2)	HR	HR	yes	yes
Dynamic Analysis and Testing (A.13) (4,3)	HR	HR	yes	yes
Metrics (5,4)	R	R	no	yes
Traceability (6,5)	HR	M	yes	yes
Software Error Effect Analysis (7,6)	HR	HR	yes	partly
Test Coverage for code (A.21) (-,7)		HR		yes
Functional/Black-box Testing (A.14) (-,8)		M		yes
Performance Testing (A.18) (-,9)		HR		partly
Interface Testing (-,10)		HR		yes
Probabilistic Testing (2,-)	HR		no	
A.6 (Software/Hardware) Integration				
Technique/Measure				
Functional and Black-box Testing (A.14) (1,1)	HR	HR	yes	yes
Performance Testing (A.18) (2,2)	HR	HR	yes	partly
A.7 2001: Software Validation, 2011: Overall Software Testing				
Technique/Measure				
Performance Testing (A.18) (2,1)	M	M	yes	yes
Functional and Black-box Testing (A.14) (3,2)	M	M	yes	yes
Modelling (A.17) (4,3)	R	R	yes	partly

Probabilistic Testing (1,-)	HR		yes	
A.8 2001: Clauses to be assessed				
S/W Safety Integrity Levels	HR		-	-
Personnel & Responsibility	HR		-	-
Lifecycle & Documentation	HR		-	-
S/W Requirements Specification	HR		-	-
S/W Architecture	HR		-	-
Design & Development	HR		-	-
Verification	HR		-	-
S/W/H/W Integration	HR		-	-
S/W Validation	HR		-	-
Quality Assurance	HR		-	-
Maintenance	HR		-	-
A.8 2011: Software Analysis Techniques				
Static Software Analysis (A.19)		HR		yes
Dynamic Software Analysis (A.13 + A.14)		HR		yes
Cause Consequence Diagrams		R		no
Event Tree Analysis		R		no
Software Error Effect Analysis		HR		yes
A.9 2001: Software Assessment				
Assessment Techniques				
Checklists	HR			yes
Static Software Analysis	HR			yes
Dynamic Software Analysis	HR			yes
Cause Consequence Diagrams	R			yes
Event Tree Analysis	R			yes
Fault Tree Analysis	HR			yes
Software Error Effect Analysis	HR			yes
Common Cause Failure Analysis	HR			yes
Markov Models	R			no
Reliability Block Diagram	R			no
Field Trial Before Commissioning	HR			yes
A.9 2011 = A.10 2001 Software Quality Assurance				
Technique/Measure				
Accredited to EN ISO 9001 (1,1)	HR	HR		yes
Compliant with EN ISO 9001 (-,2)		M		yes
Compliant with ISO/IEC 90003 (2,3)	M	R		yes
Company Quality System (3,4)	M	M		yes
Software Configuration Management (4,5)	M	M		yes
Checklists (-,6)		HR		yes
Traceability (-,7)		M		yes
Data Recording and Analysis (-,8)		M		yes
A.11 2001 = A.10 2011 Software Maintenance				
Technique/Measure				
Impact Analysis (1,1)	M	M	yes	yes

Data Recording and Analysis (2,2)	M	M	yes	yes
A.11 2011 Data Preparation Techniques				
Technique/Measure				
Tabular Specification Methods		R		-
Application Specific Language		R		-
Simulation		HR		-
Functional testing		M		-
Checklists		M		-
Fagan inspection		R		-
Formal design reviews		HR		-
Formal proof of correctness (of data)		HR		-
Walkthrough		HR		-
A.12 (Design and) Coding Standards				
Technique/Measure				
Coding Standard (1,1)	HR	M		yes
Coding Style Guide (2,2)	HR	HR		yes
No Dynamic Objects (3,3)	HR	HR		yes
No Dynamic Variables (4,4)	HR	HR		yes
Limited Use of Pointers (5,5)	R	R		yes
Limited Use of Recursion (6,6)	HR	HR		yes
No Unconditional Jumps (7,7)	HR	HR		yes
Limited size and complexity of Functions,				
Subroutines and Methods (-,8)		HR		yes
Limited number of subroutine parameters (-,9)		R		yes
Limited use of Global Variables (-,10)		M		yes
A.13 Dynamic Analysis and Testing				
Technique/Measure				
Test Case Execution from Boundary Value Analysis (1,1)	HR	HR		yes
Test Case Execution from Error Guessing (2,2)	R	R		yes
Test Case Execution from Error Seeding (3,3)	R	R		yes
Performance Modelling (4,4)	HR	HR		yes
Equivalence Classes and Input Partition Testing (5,5)	HR	HR		yes
Structure-Based Testing (6,6)	HR	HR		yes
A.14 Functional/Black Box Test				
Technique/Measure				
Test Case Execution from Cause Consequence Diagrams (1,1)	R	R		no
Prototyping/Animation (2,2)	R	R		no
Boundary Value Analysis (3,3)	HR	HR		yes
Equivalence Classes and Input Partition Testing (4,4)	HR	HR		yes
Process Simulation (5,5)	R	R		yes
A.15 (Textual) Programming Languages				
Technique/Measure				
ADA (1,1)	R	HR		no
MODULA-2 (2,2)	R	HR		no
PASCAL (3,3)	R	HR		no

Product 2 SIL 3/SIL 4

C or C++ (6,4)	R	R	yes	yes
PL/M (7,5)	NR	NR		no
BASIC (8,6)	NR	NR		no
Assembler (9,7)	-	R		no
C# (-,8)		R		no
JAVA (-,9)		R		no
Statement List 12,10)	R	R		no
Fortran 77 (4,-)	R			no
C or C++ (unrestricted) (5,-)	NR			no
Ladder Diagrams (10,-)	R			no
Functional Blocks (11,-)	R			no
A.16 2001 = A.17 2011 Modelling				
Technique/Measure				
Data Modelling (-,1)		HR		no
Data Flow Diagrams (1,2)	R	HR		no
Control Flow Diagrams (-,3)		HR		partly
Finite State Machines or State Transition Diagrams (2,4)	HR	HR		yes
Time Petri Nets (5,5)	HR	HR		no
Decision/Truth Tables (-,6)		HR		partly
Formal Methods (3,7)	HR	HR		partly
Performance Modelling (4,8)	HR	HR		partly
Prototyping/Animation (6,9)	R	R		no
Structure Diagrams (7,10)	HR	HR		yes
Sequence Diagrams (-,11)		HR		yes
A.16 2011 Diagrammatic Languages for Application Algorithms				
Technique/Measure				
Functional Block Diagrams		R		-
Sequential Function Charts		HR		-
Ladder Diagrams		R		-
State Charts		HR		-
A.17 2001 = A.18 2011 Performance Testing				
Technique/Measure				
Avalanche/Stress Testing (1,1)	HR	HR		partly
Response Timing and Memory Constraints (2,2)	HR	HR		partly
Performance Requirements (3,3)	HR	HR		partly
A.19 Static Analysis				
Technique/Measure				
Boundary Value Analysis (1,1)	HR	HR		yes
Checklists (2,2)	R	R		yes
Control Flow Analysis (3,3)	HR	HR		partly
Data Flow Analysis (4,4)	HR	HR		partly
Error Guessing (5,5)	R	R		yes
Walkthroughs/Design Reviews (9,6)	HR	HR		yes
Fagan Inspections (6,-)	HR			
Sneak Circuit Analysis (7,-)	R			
Symbolic Execution (8,-)	HR			

Product 2 SIL 3/SIL 4

A.18 2001 Semi-Formal Methods				
Technique/Measure				
Logic/Function Block Diags	HR			-
Sequence Diagrams	HR			-
Data Flow Diagrams	R			-
Finite State Machines/State Transition Diagrams	HR			-
Time Petri Nets	HR			-
Decision/Truth Tables	HR			-
A.20 2001: Modular Approach 2011: Components				
Technique/Measure				
Information Hiding (-,1)		-		yes
Information Encapsulation (-,2)		HR		yes
Information Hiding/Encapsulation (2,-)	HR			
Parameter Number Limit (3,3)	R	R		no
Fully Defined Interface (5,4)	M	M		yes
Module Size Limited (1,-)	HR			
One Entry/One Exit Point in Subroutines and Functions (4,-)	HR			
A.21 2011 Test Coverage for Code				
Test coverage criterion				
Statement		HR		yes
Branch		HR		partly
Compound Condition		HR		partly
Data flow		HR		no
Path		HR		no
A.22 2011 Object Oriented Software Architecture				
Technique/Measure				
Traceability of the concept of the application domain to the				
classes of the architecture		HR		-
Use of suitable frames, commanly used combinations of				
classes and design patterns		HR		-
Object Oriented Detailed Design		HR		-
A.23 2011 Object Oriented Detailed Design				
Technique/Measure				
Classes should have only one objective		HR		-
Inheritance used only if the derived class is a refinement				
of its basic class		HR		-
Depth of inheritance limited by coding standards		HR		-
Overriding of operations (methods) under strict control		HR		-
Multiple inheritance used only for interface classes		HR		-
Inheritance from unknown classes		NR		-

8.1.1 Appendix A - Requirements to the tables

Table	Requirement	Change Comments	Costs/Comments
A.1 Lifecycle Issues and Documentation	2011: *NOTE 1 According to 5.3.2.11 and 5.3.2.12, documents can be combined differently.* *NOTE 2 Documents 29, 30 and 31 being HR or R depends on the importance defined in the process and where the verification takes place. E.g. data may only be needed to be verified but tested in the system domain while more functional properties need both test and verification. In this case HR has been marked but can be optional R.* 2001: Compliance with EN ISO 9000-3 implies the production of adequate documentation for all Software Safety Integrity Levels. For Software Safety Integrity Level 0, the designer shall choose suitable types of document.	(5.3.2.11) The contents of all documents shall be recorded in a form appropriate for manipulation, processing and storage, (5.3.2.12) When documents which are produced by independent roles are combined into a single document, the relation to the parts produced by any independent role shall be traced within the document.	• Product 1: Used most of the required documentations before the version update. No specific notations of differences. • Product 2: This product is older than product 1, which means that they are testing in a different way for some parts of the process. This gives them approved reasons to merge a lot of the documentations together. Used most of the required documentations before the version update. No specific notations of differences.

A.2 Software Requirements Specification	2011: 1. The Software Requirements Specification shall include a description of the problem in natural language and any necessary formal or semi-formal notation. 2. The table reflects additional requirements for defining the specification clearly and precisely. One or more of these techniques shall be selected to satisfy the Software Safety Integrity Level being used. 2001: 1. The Software Requirements Specification will always require a description of the problem in natural language and any necessary mathematical notation that reflects the application. 2. The table reflects additional requirements for defining the specification clearly and precisely. One or more of these techniques shall be selected to satisfy the Software Safety Integrity Level being used.	-	Formal methods is used for product 2, but not for product 1. Product 1 should and want to implement some formal method to the process, in order to prove all the logic in the code. It is an extensive work to implement these kind of methods, especially in a late stage of the process. The supplier has planned to implement formal methods to product 1.

A.3 Software Architecture	2011: 1. Approved combinations of techniques for Software Safety Integrity Levels 3 and 4 are as follows: • 1, 7, 19, 22 and one from 4, 5, 12 or 21, • 1, 4, 19, 22 and one from 2, 5, 12, 15 or 21. 2. Approved combinations of techniques for Software Safety Integrity Levels 1 and 2 are as follows: 1, 19, 22 and one from 2, 4, 5, 7, 12, 15 or 21. 3. Some of these issues may be defined at the system level. 4. Error detecting codes may be used in accordance with the requirements of EN 50159. *NOTE: Technique/measure 19 is for External Interfaces.* 2001: 1. Approved combinations of techniques for Software Safety Integrity Levels 3 and 4 shall be as follows: • 1, 7 and one from 4, 5 or 12, • 1, 4 and 12, • 1, 2 and 4, • 1 and 4, and one of 15 and 16. 2. With the exception of entries 3, 9, 10, 13 and 14, one or more of these techniques shall be selected to satisfy the requirements for Software Safety Integrity Levels 1 and 2. 3. Some of these issues may be defined at the system level. 4. Error correcting codes may be used in accordance with the requirements of EN 50159-1 and EN 50159-2.	All of the techniques/methods from EN 50128:2001 are included in EN50128:2011, except 'Fault Tree Analysis'. There is eight new techniques/methods added in this table and the approved combination is changed in the 2011 version. For SIL 4, there are the same recommendations for the existing methods/techniques. As can be seen from this, the approved combinations of techniques/methods are changed in the 2011 version. There are still no requirements regarding SIL 0. SIL 1 and SIL 2 has to use at least two methods/techniques as the 2011 version of the standard is used. In the 2001 version of the standard there were only one technique that had to be chosen.	Formal methods is also missing for product 1 in this part of the process.

A.4 Software Design and Implementation	2011: 1. An approved combination of techniques for Software Safety Integrity Levels 3 and 4 is 4, 5, 6, 8 and one from 1 or 2. 2. An approved combination of techniques for Software Safety Integrity Levels 1 and 2 is 3, 4, 5, 6 and one from 8, 9 or 10. 3. Metaprogramming shall be restricted to the production of the code of the software source before compilation. 2001: 1. A suitable set of techniques shall be chosen according to the software safety integrity level. 2. At software safety integrity level 3 or 4, the approved set of techniques shall include one of techniques 1, 2 or 3, together with one of techniques 11 or 12. The remaining techniques shall still be treated according to their recommendations.	The requirements connected to this table is stricter in the 2011 version of the standard. In the 2001 version, there were no restrictions for SIL 0, SIL 1 and SIL 2. In the 2011 version, there is restrictions for SIL 1 and SIL 2, but non for SIL 0. A lot of the techniques/measure are deleted from this standard.	Formal methods is also missing for product 1 in this part of the process.

A.5 Verification and Testing	2011: 1. For software Safety Integrity Levels 3 and 4, the approved combination of techniques is 3, 5, 7, 8 and one from 1, 2 or 6. 2. For Software Safety Integrity Level 1 and 2, the approved combinations of techniques is 5 together with one from 2, 3 or 8. *NOTE 1 Techniques/measures 1, 2, 4, 5, 6 and 7 are for verification activities.* *NOTE 2 Techniques/measures 3, 8, 9 and 10 are for testing activities.* 2001: 1. For Software Safety Integrity Level 3 or 4, the approved combinations of techniques shall be: • 1 and 4, • 3 and 4, or • 4, 6 and 7. 2. For Software Safety Integrity Level 1 or 2, the approved technique shall be 1 or 4.	Some methods/techniques are added to this table. For SIL 0, there were no specific recommendations in the 2001 version, compare to the 2011 version, where recommendations are added for SIL 0 as well. As can be seen in these requirements, there is an increased number of methods/techniques that are required to be used. For SIL 3 and SIL 4 the number of required techniques, have increased from two or three, to five. For SIL 1 and SIL 2 the number of required methods/techniques has increased from one to two.	It is not specified in this standard whether the Blackbox testing has to be done for all phases, only that it has to be done. Blackbox testing is not done in all phases, only overall blackbox tests. For the different phases, "greybox testing" is used. In greybox testing, parts of the code is known, and some parts unknown. This way of interpretation of blackbox testing is approved by the Assessor. The Assessor did also recommend the supplier to use "whitebox testing" for some phases. Blackbox testing was a included in the 2001 version as well. The requirements (text) does not agree with the requirements in this table. Two requirements handling the same area have to agree. Requirement 6.2.4.5 indicates that you should chose methods/techniques that satisfy 4.8, 4.9 and 4.10. Non of these requirements says anything about approved combinations.

A.6 2011: Integration A.6 2001: Software /Hardware Integration	-	This table has not changed content nor recommendations. The name of the table is changed from 'Software/Hardware Integration' to 'Integration'.	Everything can not be tested for these generic products by the supplier themselves, the whole system might affect parts of the process for the product, for example the amount of trains on the track at the same time, the speed of the trains, a load missing etc. It might be more correct to mark the 'Performance Testing': 'partly', than 'yes' even for EN 50128:2001, in the table.
A.7 2011: Overall Software Testing A.7 2001: Software Validation	2001: For Software Safety Integrity Level 1, 2, 3 or 4, an approved combination of techniques shall be 2 and 3. 2011: For Software Safety Integrity Level 1 and 2 an approved combination of techniques is 1 and 2.	The name of this table is changed from 'Software Validation' to 'Overall Software Testing'. Since method/technique one and two are mandatory, the requirements for SIL 3 and SIL 4 are removed.	'Overall Software testing' is partly done, since the products could be affected by other parts, as an integration with the system are performed. For product 2, 'Performance testing' should be marked with 'partly' instead of 'yes'.
A.8 2011: Software Analysis Techniques	2011: One or more of these techniques shall be selected to satisfy the Software Safety Integrity Level being used.	This table is new to this European standard.	'Cause Consequence Diagrams' might be used for product 2, the supplier are uncertain about this part.

A.8 2001: Clauses to be assessed	-	This table is deleted from this European standard.	This table was not used by the supplier, since this table is directed to the work of the Assessor. The supplier made sure that all of the required parts were covered. In the 2011 version of this standard, this table is deleted and the description of the responsibilities of the Assessor are added.
A.9 2001: Software Assessment	2001: One or more of these techniques shall be selected to satisfy the Software Safety Integrity Level being used.	This table is removed from this standard. It was criticized by more than one of the national committees for not being to precise. Responsible persons for the methods/techniques and approved combinations was missing according to the committees.	See comments on A.8 2001.
A.9 2011 = A.10 2001 Software Quality Assurance	2011: This table shall be applied to different roles and all phases.	This table now includes eight methods/techniques, instead of four.	The new parts of this table were in use before the update in version. The only part that is new, is the 'checklists'. The checklists are not new to this standard, but they are added to more parts of it. Checklists are not hard to implement, but it can be tricky to know what to include. This part can thereby be time consuming, but not in the long run.

A.10 2011 = A.11 2001 Software Maintenance	-	There is no changes in content or recommendation levels for this table.	-
A.11 2011: Data Preparation Techniques	2011: 1. For Software Safety Integrity Level 1 and 2 an approved combination of techniques is 1 and 4. 2. For Software Safety Integrity Level 3 and 4 the approved combinations of techniques are 1, 4, 5 and 7 or 2, 3 and 6. *NOTE The description of the reference D.29 is on programs while technique 8 in this context applies to formal proof of the correctness of data.*	This table is new to the standard. In the 2001 version, 'checklist' was only a part of table A.9: 'Software Assessment'. For 'Data Preparation Techniques', 'checklist' is a requirement for SIL 3 and SIL 4.	Since product 1 and 2 are Generic products this table is not used. This decision is approved by the Assessor. The argumentation of why this table is deleted from the process, should be documented in some of the documentations.
A.12 2011: Coding standards A.12 2001: Design and Coding Standards	2011: It is accepted that techniques 3, 4 and 5 may be present as part of a validated compiler or translator. 2001: 1. It is accepted that techniques 3, 4 and 5 may be present as part of a validated compiler or translator. 2. A suitable set of techniques shall be chosen according to the software safety integrity level.	The name is changed from 'Design and Coding Standards' to 'Coding Standards'. There is three new methods/techniques added to this table. In the 2011 version there is two mandatory methods/techniques. In the 2001 version, there were no mandatory methods/techniques.	-

A.13 Dynamic Analysis and Testing	2011: The analysis for the test cases is at the sub-system level and is based on the specification and/or the specification and the code. 2001: 1. The analysis for the test cases is at the sub-system level and is based on the specification and/or the specification and the code. 2. A suitable set of techniques shall be chosen according to the software safety integrity level.	There is no changes in content or recommendations of this table.	'Test case Execution from Error Seeding' is only done with product 2. In order to do this, the programmer has to be experienced. The product also has to be old enough, in order to have a lot of known errors. Product 1 is a lot newer than product 2, and product 2 have had the same programmers for a long time.
A.14 Functional /Black Box Test	2011: The completeness of the simulation will depend upon the extent of the software safety integrity level, complexity and application. 2001: 1. The completeness of the simulation will depend upon the extent of the software safety integrity level, complexity and application. 2. A suitable set of techniques shall be chosen according to the software safety integrity level. 2011: The completeness of the simulation will depend upon the extent of the software safety integrity level, complexity and application.	There is no changes in content or recommendation levels for this table.	-

A.15 2011: Textual Programming Languages A.15 2001: Programming Languages	2011: 1. The selection of the languages shall be based on the requirements given in 6.7 and 7.3. 2. There is no requirement to justify decisions taken to exclude specific programming languages. *NOTE 1 For information on assessing the suitability of a programming language see entry in D.54 ‘Suitable Programming Languages’.* *NOTE 2 If a specific language is not in the table, it is not automatically excluded. It should, however, conform to D.54.* *NOTE 3 Run-time systems associated with selected languages which are necessary to run application programs should still be justified for usage according to the Software Safety Integrity Level.* 2001: 1. At Software Safety Integrity Level 3 and 4 when a subset of languages 1, 2, 3 and 4 are used the recommendation changes to HR. 2. For certain applications the languages 7 and 9 may be the only ones available. At Software Safety Integrity Level 3 and 4 where a Highly recommended option is not available it is strongly recommended that to raise the recommendation to ’R’ there should be a subset of these languages and that there should be a precise set of coding standards. 3. For information on assessing the suitability of a programming language see entry in the bibliography for ’Suitable Programming Language’, B.62.	The changes in this table are: ’Fortran 77’, ’Ladder diagrams’ and ’Functional Blocks’ are removed, ’C or C++’ are described as one part and ’JAVA’ and ’C#’ is new. ’ADA’, ’MODULA’ and ’PASCAL’ have changed recommendations. They are now highly recommended (HR) for SIL 3 and 4. In the 2001 version, these languages were marked as recommended (R) for SIL 3 and SIL 4.	The only languages used for both of the products are C and C++. These languages are less controlled and more widely used than many of the other languages. ADA, MODULA-2 and PASCAL is highly recommended since these languages are stricter controlled in how to use them, this makes them safer, but harder to use if the programmer is used to a ”free” language.

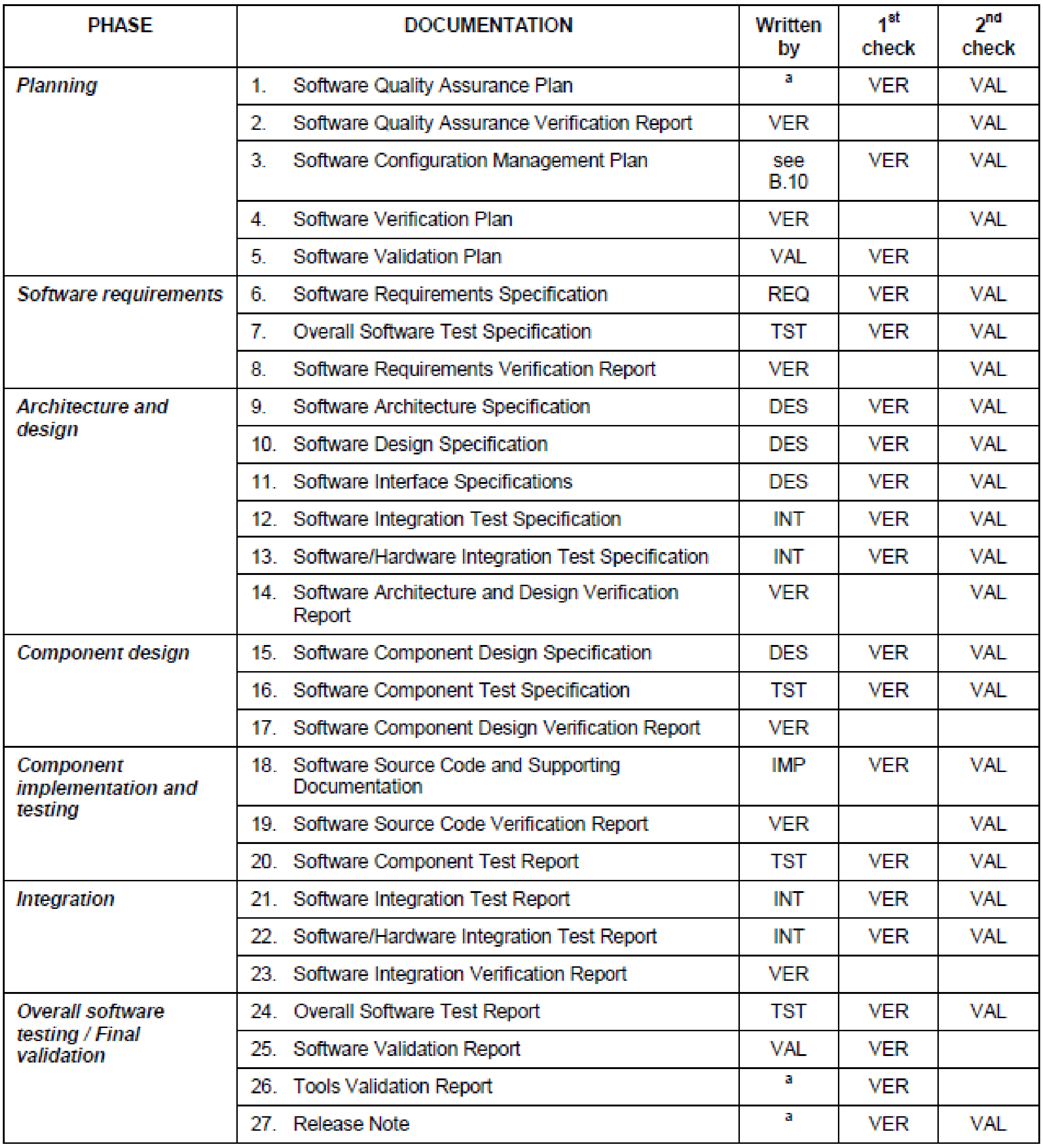
	4. If a specific language is not in the table, it is not automatically excluded. It should, however, conform to B.62.		
A.16 2011: Diagrammatic Languages for Application Algorithms	-	This table is new to this standard.	Since product 1 and 2 are generic products, this table is excluded. The argument to this exclusion is that the Application algorithms are produced outside of the generic products and then applied to these products after the process is done. This is approved by the Assessor.
A.17 2011 = A.16 2001 Modelling	2011: 1. A modeling guideline shall be defined and used. 2. At least one of the highly recommended (HR) methods/techniques shall be chosen. 2001: A suitable set of methods/techniques shall be chosen according to the software safety integrity level.	There is four new methods/techniques added to this table: 'Data Modelling', 'Control Flow Diagrams', 'Decision/Truth Tables' and 'Sequence Diagrams'. All of these new methods/techniques are highly recommended (HR) for SIL 3 and SIL 4.	-

A.18 2011 = A.17 2001 Performance Testing	2001: A suitable set of methods/techniques shall be chosen according to the software safety integrity level.	There is no changes in content or recommendation levels for this table.	All tests can not be done completely for the generic products. There can be a load missing or the the amount of trains at the same time on the track can affect the result, and this can not be tested by the supplier alone. Since there is no requirements related to this table and non of the performance tests are mandatory, non of these tests has to be done.
A.18 2001 Semi-Formal Methods	2001: A suitable set of techniques shall be chosen according to the software safety integrity level.	This table is removed from this European Standard.	-
A.19 Static Analysis	2001: A suitable set of methods/techniques shall be chosen according to the software safety integrity level.	'Fagan Inspections', 'Sneak Circuit Analysis' and 'Symbolic Execution' is removed from this table.	The removed methods/techniques of this table were not used by the supplier.
A.20 2011: Components A.20 2001: Modular Approach	2011: Information Hiding and encapsulation are only highly recommended if there is no general strategy for data access. *NOTE Technique/measure 4 is for Internal Interfaces.* 2001: A suitable set of methods/techniques shall be chosen according to the software safety integrity level.	'Module' is changed to 'Component' in this standard. 'Information Hiding' and 'Information Encapsulation' is new to this standard. 'Information Hiding/Encapsulation'. 'Module Size Limited' and 'One Entry/One Exit Point in Subroutines and Functions' are removed.	For the supplier, 'Information hiding' and 'Information Encapsulation' were used before the update of this standard.

A.21 2011 Test Coverage for Code	2011: 1. For every SIL, a quantified measure of coverage shall be developed for the test undertaken. This can support the judgment on the confidence gained in testing and the necessity for additional techniques. 2. For SIL 3 or 4 test coverage at component level should be measured according to the following: • 2 and 3; or • 2 and 4; or • 5 3. or test coverage at integration level should be measured according to one or more of 2, 3, 4 or 5. 4. Other test coverage criteria can be used, given that this can be justified. These criteria depend on the software architecture (see Table A.3) and the programming language (see Table A.15 and Table A.16). 5. Any code which it is not practicable to test shall be demonstrated to be correct using a suitable technique, e.g. static analysis from Table A.19. *NOTE 1 Statement coverage is automatically achieved by items 2 to 5.* *NOTE 2 The test coverage criteria in this table are used for structure-based (code-based, white box) testing. Techniques/measures for functional (specification-based, black box) testing are given in Table A.14.* *NOTE 3 A high percentage of coverage is usually difficult to achieve. The use of test case execution from boundary values (D.4) and equivalence classes and input partition testing (D.18) can enable a sufficient coverage to be achieved with a smaller number of tests.*	This table is new to this European Standard.	Since non of product 1 and 2 are using ’Data flow’ or ’Path’ Test coverages for code, the only approved combination for this table left is ’Branch’ + ’Compound conditions’ Tests. • Product 1: Product 1 is testing every part of the code, so these techniques can be fully used. • Product 2: Product 2 is only testing the code ”road by road”, which means that the table is partly covered, not completely. Since the code for product 2 is older than the code for product 1, and this table with requirements is new, product 2 might have to change the way of performing these tests in the code, which can be an extensive work.

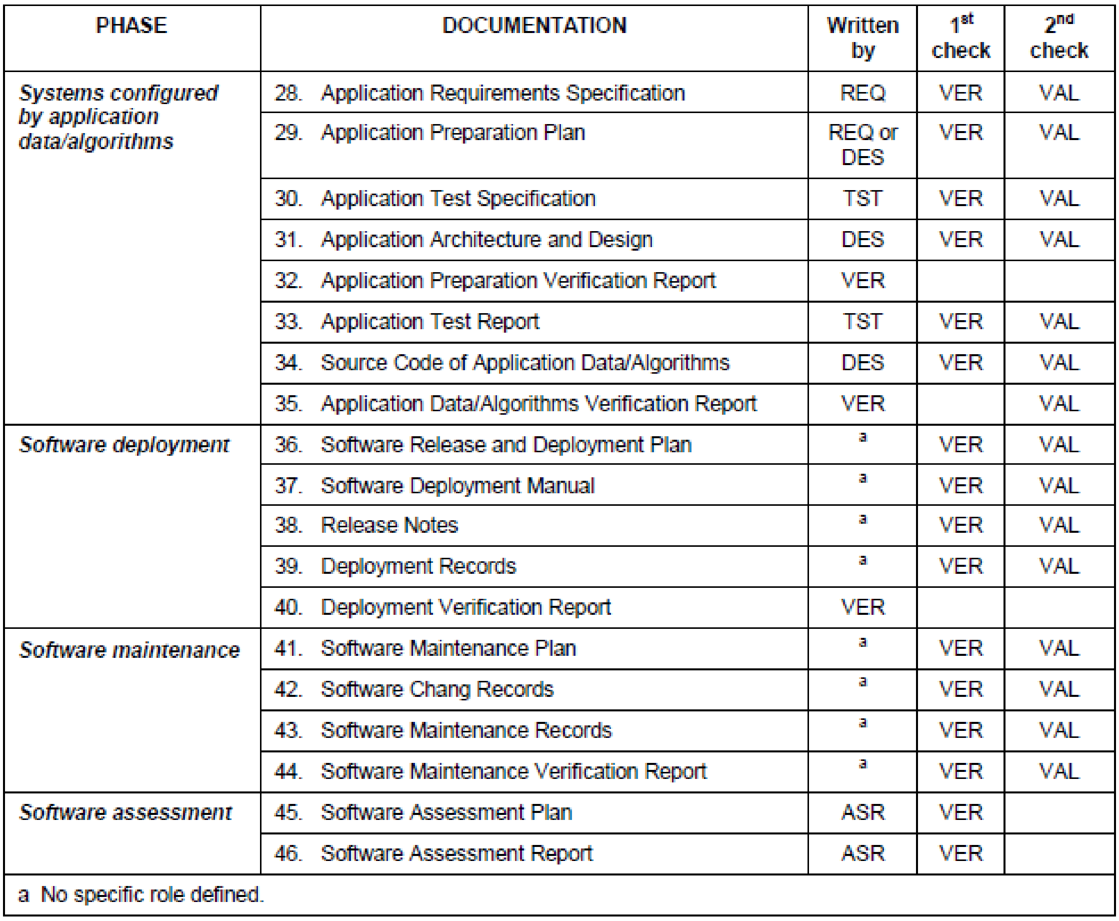
	NOTE 4 The difference between 2 and 3 depends in practice on the level of the programming language and the use of compound conditions. When single conditions are used only, for example as a result of compilation, 2 and 3 are considered identical.		
A.22 2011: Object Oriented Software Architecture	When using existing frames and design patterns, the requirements of pre-existing software apply to these frames and patterns. *NOTE 1 The object-oriented approach presents information differently from procedural approaches, the following list contains recommendations that need specific consideration:* *- understanding class hierarchies, and identification of the software function(s) that will be executed upon the invocation of a given method (including when using an existing class library);* *- structure-based testing (Table A.13).* *Traceability from application domain to class architecture is less important.* *NOTE 2 For a part of the intended software a frame might exist from pre-existing software that has successfully solved a similar task and that is well known to the development personnel. Then use of that frame is recommended.*	This table is new to this European standard.	This table is not used on product 1 and 2, since these products are generic products. The argumentation for this is approved by the Assessor.
A.23 2011: Object Oriented Detailed Design	2011: 1. One class is characterized by having one responsibility, i.e. taking care of closely connected data and the operations on these data. 2. Care is required to avoid circular dependencies between objects.	This table is new to this European standard.	This table is not used on product 1 and 2, since these products are generic products. The argumentation for this is approved by the assessor.

8.1.2 Appendix A - Lifecycle documents



PHASE	DOCUMENTATION	Written by	1st check	2nd check
Planning	1. Software Quality Assurance Plan	[a]	VER	VAL
	2. Software Quality Assurance Verification Report	VER		VAL
	3. Software Configuration Management Plan	see B.10	VER	VAL
	4. Software Verification Plan	VER		VAL
	5. Software Validation Plan	VAL	VER	
Software requirements	6. Software Requirements Specification	REQ	VER	VAL
	7. Overall Software Test Specification	TST	VER	VAL
	8. Software Requirements Verification Report	VER		VAL
Architecture and design	9. Software Architecture Specification	DES	VER	VAL
	10. Software Design Specification	DES	VER	VAL
	11. Software Interface Specifications	DES	VER	VAL
	12. Software Integration Test Specification	INT	VER	VAL
	13. Software/Hardware Integration Test Specification	INT	VER	VAL
	14. Software Architecture and Design Verification Report	VER		VAL
Component design	15. Software Component Design Specification	DES	VER	VAL
	16. Software Component Test Specification	TST	VER	VAL
	17. Software Component Design Verification Report	VER		
Component implementation and testing	18. Software Source Code and Supporting Documentation	IMP	VER	VAL
	19. Software Source Code Verification Report	VER		VAL
	20. Software Component Test Report	TST	VER	VAL
Integration	21. Software Integration Test Report	INT	VER	VAL
	22. Software/Hardware Integration Test Report	INT	VER	VAL
	23. Software Integration Verification Report	VER		
Overall software testing / Final validation	24. Overall Software Test Report	TST	VER	VAL
	25. Software Validation Report	VAL	VER	
	26. Tools Validation Report	[a]	VER	
	27. Release Note	[a]	VER	VAL

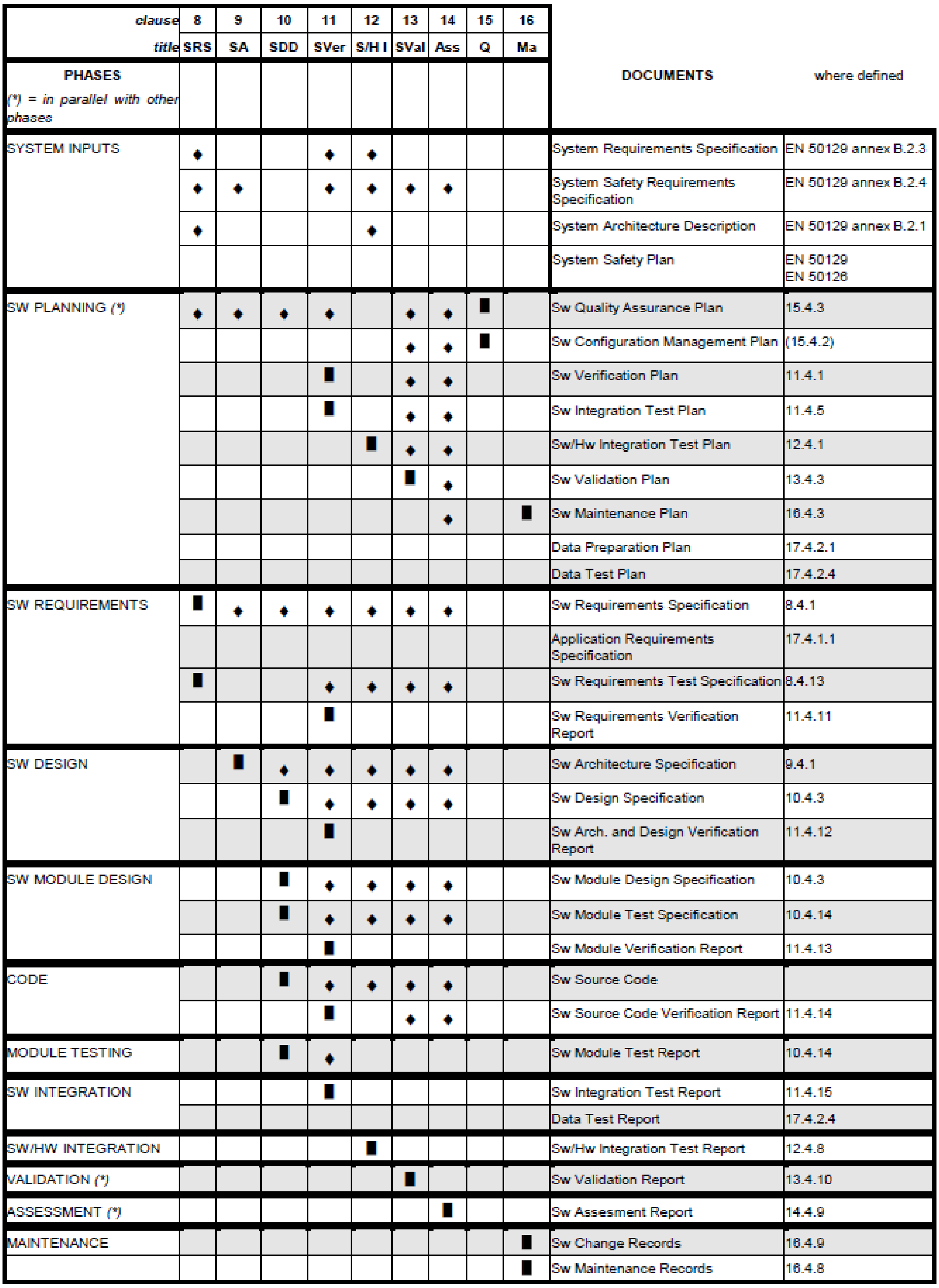
Figure 8.1: The Document Control Summary for the 2011 version of this standard, part 1.



PHASE	DOCUMENTATION	Written by	1st check	2nd check
Systems configured by application data/algorithms	28. Application Requirements Specification	REQ	VER	VAL
	29. Application Preparation Plan	REQ or DES	VER	VAL
	30. Application Test Specification	TST	VER	VAL
	31. Application Architecture and Design	DES	VER	VAL
	32. Application Preparation Verification Report	VER		
	33. Application Test Report	TST	VER	VAL
	34. Source Code of Application Data/Algorithms	DES	VER	VAL
	35. Application Data/Algorithms Verification Report	VER		VAL
Software deployment	36. Software Release and Deployment Plan	[a]	VER	VAL
	37. Software Deployment Manual	[a]	VER	VAL
	38. Release Notes	[a]	VER	VAL
	39. Deployment Records	[a]	VER	VAL
	40. Deployment Verification Report	VER		
Software maintenance	41. Software Maintenance Plan	[a]	VER	VAL
	42. Software Chang Records	[a]	VER	VAL
	43. Software Maintenance Records	[a]	VER	VAL
	44. Software Maintenance Verification Report	[a]	VER	VAL
Software assessment	45. Software Assessment Plan	ASR	VER	
	46. Software Assessment Report	ASR	VER	

[a] No specific role defined.

Figure 8.2: The Document Control Summary for the 2011 version of this standard, part 2.



clause	8	9	10	11	12	13	14	15	16		
title	SRS	SA	SDD	SVer	S/H I	SVal	Ass	Q	Ma		
PHASES (*) = in parallel with other phases										DOCUMENTS	where defined
SYSTEM INPUTS	♦			♦	♦					System Requirements Specification	EN 50129 annex B.2.3
	♦	♦		♦	♦	♦	♦			System Safety Requirements Specification	EN 50129 annex B.2.4
	♦				♦					System Architecture Description	EN 50129 annex B.2.1
										System Safety Plan	EN 50129 EN 50126
SW PLANNING (*)	♦	♦	♦	♦		♦	♦	█		Sw Quality Assurance Plan	15.4.3
						♦	♦	█		Sw Configuration Management Plan	(15.4.2)
				█		♦	♦			Sw Verification Plan	11.4.1
				█		♦	♦			Sw Integration Test Plan	11.4.5
					█	♦	♦			Sw/Hw Integration Test Plan	12.4.1
						█	♦			Sw Validation Plan	13.4.3
							♦		█	Sw Maintenance Plan	16.4.3
										Data Preparation Plan	17.4.2.1
										Data Test Plan	17.4.2.4
SW REQUIREMENTS	█	♦	♦	♦	♦	♦	♦			Sw Requirements Specification	8.4.1
										Application Requirements Specification	17.4.1.1
	█			♦	♦	♦	♦			Sw Requirements Test Specification	8.4.13
				█						Sw Requirements Verification Report	11.4.11
SW DESIGN		█	♦	♦	♦	♦	♦			Sw Architecture Specification	9.4.1
			█	♦	♦	♦	♦			Sw Design Specification	10.4.3
				█						Sw Arch. and Design Verification Report	11.4.12
SW MODULE DESIGN			█	♦	♦	♦	♦			Sw Module Design Specification	10.4.3
			█	♦	♦	♦	♦			Sw Module Test Specification	10.4.14
				█						Sw Module Verification Report	11.4.13
CODE			█	♦	♦	♦	♦			Sw Source Code	
				█		♦	♦			Sw Source Code Verification Report	11.4.14
MODULE TESTING			█	♦						Sw Module Test Report	10.4.14
SW INTEGRATION				█						Sw Integration Test Report	11.4.15
										Data Test Report	17.4.2.4
SW/HW INTEGRATION					█					Sw/Hw Integration Test Report	12.4.8
VALIDATION (*)						█				Sw Validation Report	13.4.10
ASSESSMENT (*)							█			Sw Assesment Report	14.4.9
MAINTENANCE									█	Sw Change Records	16.4.9
									█	Sw Maintenance Records	16.4.8

Figure 8.3: The Document Cross-Reference Table for the 2001 version of this standard.

8.2 Appendix B - New requirements in EN 50128

Sub-Clause	New requirements	Comments

3 Objectives, Conformance and Software Safety Integrity Levels Comments: Harder restrictions regarding the Safety integrity level (SIL), especially for SIL 0. SIL 1 and SIL 2 has the same requirements and SIL 3 and SIL 4 has stricter requirements in the new version of the standard [6]. In the 2001 version, SIL 0 did not have any requirements, which means that this part of the standard will require a lot more work and resources. During the process of maintain this standard, the discussion about SIL was the main concern by the national committees, especially regarding SIL 0. The Czech and the French national committees commented that the requirements regarding SIL has to be maintained and needs more requirements. They were especially pointing at SIL 0, SIL 2 and SIL 4. The Dutch national committee stuck to their opinion that SIL 0 should be excluded from this European standard. They argued that SIL 0 will not have any affect on the safety of the system, and since this standard is about safety related software, SIL 0 should be excluded. They also argued that EN 50129 should be better related to EN 50128, and since EN 50129 did not has the SIL 0, EN 50128 should not include it ether. The other national committees argued that it is almost impossible to prove that a software is of "zero risk", so non-safety related software will overlap safety related software, and thereby be safety related. The meaning of SIL 0 has been changed from 'no safety impact' to 'the lowest level of safety impact'. Requirements regarding the methods/techniques are much harder in the 2011 version.		
	(4.5) To conform to this European Standard it shall be shown that each of the requirements has been satisfied to the software safety integrity level defined and therefore the objective of the sub clause in question has been met.	Product 1 and 2 are both graded to be SIL 4. Therefore has this standard been followed according to the recommendations and requirements to this SIL level. The supplier has products that are SIL 0, these products is following the same working methods as the SIL 4 products. This way the supplier is on the safe side regarding requirements and safety. The question is if this really is the most time saving way to work.
	(4.6) Where a requirement is qualified by the words "to the extent required by the software safety integrity level", this indicates that a range of techniques and measures shall be used to satisfy that requirement.	-

	(4.7) Where 4.6 is applied, tables from normative Annex A shall be used to assist in the selection of techniques and measures appropriate to the software safety integrity level. The selection shall be documented in the Software Quality Assurance Plan or in another document referenced by the Software Quality Assurance Plan. Guidance to these techniques is given in the informative Annex D. Check Annex A compliance: • A.1 • A.2 • A.3 • A.4 • A.5 • A.6 • A.7 • A.8 • A.9 • A.10 • A.11	The supplier made lists for both product 1 and 2 of the tables in Annex A and checked of each method/technique used, for that specific SIL, see section 8.1. This part of the update process took a lot of time for both of the products. Product 1 made this checklists before product 2, this way product 2 could follow the templates that product 1 created. The supplier saved some time from that. On the other hand did the developer of product 2 include longer and more detailed description to each part of the tables, which took a lot of time to produce.
	(4.8) If a technique or measure which is ranked as highly recommended (HR) in the tables is not used, then the rationale for using alternative techniques shall be detailed and recorded either in the Software Quality Assurance Plan or in another document referenced by the Software Quality Assurance Plan. This is not necessary if an approved combination of techniques given in the corresponding table is used. The selected techniques shall be demonstrated to have been applied correctly.	The supplier uses approved combinations of methods/techniques for product 1, but product 2 is not fulfilling the requirements to table 21 - Test Coverage for Code to the fullest. The supplier says that documentations regarding why this part is not fulfilled will be included in the CENELEC compliance document.
	(4.9) If a technique or measure is proposed to be used that is not contained in the tables then its effectiveness and suitability in meeting the particular requirement and overall objective of the sub clause shall be justified and recorded in either the Software Quality Assurance Plan or in another document referenced by the Software Quality Assurance Plan.	'The Engineering process' describes the whole chain from how the supplier takes a GP (Generic Product) and produces a GA (Generic Application) and then a SA (Specific Application) before installation. This way all methods/techniques (that are possible to use) of the standard will be used in the end.

	(4.10) Compliance with the requirements of a particular sub clause and their respective techniques and measures detailed in the tables shall be verified by the inspection of documents required by this European Standard. Where appropriate, other objective evidence, auditing and the witnessing of tests shall also be taken into account.	This requirement is fulfilled. This part is important to fulfill, since external controls and interviews are done in order to make sure that the inspections and the rest of the process is done due to the requirements of the standard and by the approved role.
5 Software management and organization Comments: During the process of maintaining this standard, one of the main parts commented on, was the roles and the responsibilities. In the 2001 version of the standard, the competence level was presented as "Appropriate training, experience and qualifications". What level that is "appropriate" and who to approve that appropriate level is not described. In this 2011 version, this part is including a lot more restrictions regarding qualifications and responsibilities. It is also including who to be responsible of approving the correct level of training, experience and qualifications of the personnel. The number of roles has increased to 10 roles instead of five. The roles are more detailed described, with purpose and responsibilities. The purpose of the roles and their area of responsibilities can be found in Appendix D, section 8.4.		
5.1 Organization, Roles and Responsibilities	(5.1.2.3) The personnel assigned to the roles involved in the development or maintenance of the software shall be named and recorded.	The personnel of product 1 and 2 was having the same roles before and after the update. The supplier does not have the names and competences documented, but they do have a Safety log where the competence and the experiences are documented. The Assessor is also performing Safety Audits, where he/she control the correct experience and education of the staff. The supplier says that they will be doing internal mini-safety-audits as well.
	(5.1.2.13) The roles Requirements Manager, Designer and Implementer for one component can perform the roles Tester and Integrator for a different component.	Product 1 has at certain points borrowed personnel from product 2, in order to fulfill the new requirements regarding the roles. This has been a simple solution, since product 1 and product 2 is developed within the same company.

5.2 Personnel competence	(5.2.2.1) The key competencies required for each role in the software development are defined in Annex B. If additional experience, capabilities or qualifications are required for a role in the software life cycle, these shall be defined in the Software Quality Assurance Plan.	Some of the qualifications and areas of responsibility was seen to be to wide for some roles. The Validator and Verifier should be involved and understand the whole process. In a long and complex process, this can be to hard to execute. The experience has been that, it would be better to split this role for large processes. The supplier has a good knowledge in the different responsibilities for the roles (described in Annex D, section 8.4).
	(5.2.2.2) Documented evidence of personnel competence, including technical knowledge, qualifications, relevant experience and appropriate training, shall be maintained by the supplier's organization in order to demonstrate appropriate safety organization.	The evidence of the competences, qualifications and relevant education is documented in the safety log. Safety Audits are also performed by the Assessor and the supplier intend to perform internal mini-safety-audits.
	(5.2.2.3) The organization shall maintain procedures to manage the competence of personnel to suit appropriate roles in accordance to existing quality standards.	The supplier does not have any specific information documented regarding the competences of the staff. They do have safety logs, where the experiences and competences are documented. As mentioned earlier, Safety audits are also performed by the Assessor, where the Assessor controls that the staff has the correct knowledge and education.
	(5.2.2.4) Once it has been proved to the satisfaction of an assessor or by a certification that competence has been demonstrated for all personnel appointed in various roles, each individual will need to show continuous maintenance and development of competence. This could be demonstrated by keeping a logbook showing the activity is being regularly carried out correctly, and that additional training is being undertaken in accordance with EN ISO 9001 and ISO/IEC 90003:2004, 6.2.2 "Competence, awareness and training".	The supplier was working according to the ISO 9001 and ISO 9003 even before the update of this standard. This was a restriction from TRV that these standards were followed by the supplier. As mentioned earlier, a safety log is used by the supplier in order to document the competences and the experience of the personnel.

5.3 Lifecycle issues and Documentations Comments: This part of the standard has been extended to 49 lifecycle documents instead of 29. This is not all new documentations. Instead, some of the requirements has been divided into more parts, since these requirements was long and hard to understand in the 2001 version of this standard. More requirements are added regarding the plans. Documentations in the development process can be divided and combined, as long as all of the requirements in this standard still are met [6]. This part was included in the old version of this standard as well, but it was not clear how this could be done and who to approve the split or combining of documents. In the 2011 version of this standard, roles are added to approve this combining or splitting of documentations. Both the Validator and the Assessor has to approve these changes.		
	(5.3.2.2) The lifecycle model shall take into account the possibility of iterations in and between phases.	The supplier is aware of the Waterfall and the V-model, but they are at some stages of the process using a different order of the lifecycle documents. The V-model is illustrated in Figure 3.8 for the 2011 version of this standard.
	(5.3.2.12) When documents which are produced by independent roles are combined into a single document, the relation to the parts produced by any independent role shall be traced within the document.	Many documents are combined into one single document as the 2011 version of the standard is used. This is especially the case for product 2, since this product is older. Traceability is important for the supplier in all phases of the process.
	(5.3.2.14) Where any alternative lifecycle or documentation structure is adopted it shall be established that it meets all the objectives and requirements of this European Standard.	For product 1 and 2 the documentations recommended/required is used, but the order of the documentations is not always due to the recommendations. This change in order of the documentations is replicating the way that the supplier works with the documentations. This is approved by the Assessor.

	New parts of the Life cycle documentations: • Software Quality Assurance Verification Report • Software Release and Deployment Plan • Software Assessment Plan • Software interface specifications • Software Integration Verification Report • Overall Software Test Report • Tools Validation Report • Release Note • Application Architecture and Design • Application Preparation Verification Report • Source Code of Application Data/Algorithms • Application Data/Algorithms Verification Report • Software Deployment Manual • Release Notes • Deployment Records • Deployment Verification Report • Software Maintenance Verification Report	Most of this new documentations were used as the 2001 version of this standard were in use. In the 2001 version, the requirements/recommendations for documentations were stated as groups of documentations. These groups were translated by the supplier and the Assessor to be almost the same as the 2011 version of this standard intend it to be. These new parts of the Lifecycle documentations are thereby not all new, the interpretation from the 2001 version are now displayed in this section of the standard, which makes it clearer. See figure 4.2 and figure 4.3 as an example of how the display of the lifecycle documentations has changed. As the V-models are compared, it can be seen that in the 2001 version of this standard, the expansion of the lifecycle documents are presented, see figure 3.7.
6 Software assurance		
6.1 Software Testing Comments: This part of the 2001 version was commented in the process of maintenance to be to unclear, by all of the national committees. "What is enough?" was asked about more than one of the requirements for this part. As a result, this part has been more clear in the 2011 version. Some requirements, for the process of writing and what to include in the test specifications and test reports has been combined into one clear restrictions.		

	(6.1.4.1) Tests performed by other parties such as the Requirements Manager, Designer or Implementer, if fully documented and complying with the following requirements, may be accepted by the Verifier.	The designer is performing integration tests and some SW/HW integration tests. This is not documented. These integration tests are a sub-set of the system tests. The supplier has always been doing the integration tests and the components tests separately. The reports from these tests are not combined.
	(6.1.4.2) Measurement equipment used for testing shall be calibrated appropriately. Any tools, hardware or software, used for testing shall be shown to be suitable for the purpose.	All tools are divided into groups of suitable safety impact for product 1 and 2. The purpose and the areas of use for the tools were described for each tool as this grouping of tools were done. The purpose is described for each tool, see Figure 4.1 for an example of this description. The classification of the tools were the most extensive work in this process of change and took approximately 160 hours for each product to execute.
6.2 Software Verification Comments: The verification plan is new to this part of the standard. It is a combination of all Verifications, the Software Integration test plans and the Software/Hardware integration test plans from the 2001 version of this standard [6]. Overall is this part of the standard clearer and stricter in the 2011 version.		
	(6.2.4.11) Once the Software Verification Plan has been established, verification shall address • that the Software Verification Plan meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 6.2.4.3 to 6.2.4.9, • the internal consistency of the Software Verification Plan. • The results shall be recorded in a Software Quality Assurance Verification Report.	This requirement is a guidance and a clarification of the process of working with the Software Verification Plan. There were not any significant changes to be done in the update process, so no extra resources to be considered. Requirements 5.3.2.7, 5.3.2.10, 6.5.4.14, 6.5.4.17, 6.2.4.3 and 6.2.4.9 are all included in the 2001 version of this standard.

6.3
Software Validation
Comments:
The roll Validator has been clearer and the software can not be released before it has been approved by the Validator. The Validator can ask for additional tests, analysis and audits.
The system has to be validated in a real environment, not a simulated one. If simulations should be approved, they have to be proved to be really precise [6].

	(6.3.4.6) The Software Validation Plan shall identify the steps necessary to demonstrate the adequacy of the Overall Software Test Specification as a test against the Software Requirements Specification.	The Validator is defining what has to be done in the area of testing, including mapping between tests and requirements. This has been done by the Validator, even as the 2001 version of this standard were used. The test-requirements mapping is not performed 100 %, samples are taken, so 30-40 % of the test-requirement mapping is checked.
	(6.3.4.7) A Software Validation Report shall be written, under the responsibility of the Validator, on the basis of the input documents.	The supplier are following the document control summary, see section 8.1.2. Since the roles of the Validator and the Verifier are extensive in the 2011 version of this standard, these roles are sometimes divided into more than one role. This is approved by the Assessor.
	(6.3.4.10) The Software Validation Report shall fully state the software baseline that has been validated.	The Software Validation Report is not new to this standard, so the supplier was using this Report even before the update.
	(6.3.4.12) A Software Validation Verification Report shall be written, under the responsibility of the Verifier, on the basis of the input documents from 6.3.2.	See comments for 6.3.4.7.

	(6.3.4.13) Once the Software Validation Plan has been established, verification shall address • that the Software Validation Plan meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 6.3.4.4 to 6.3.4.6, • the internal consistency of the Software Validation Plan.	The new requirements in this requirement are 6.3.4.6, 6.5.4.15 and 6.5.4.16. The requirements for the plans are more extensive in the 2011 version, but as most of the lifecycle documents were already in use, the added requirements did not cost a lot of resources for the supplier to implement. See each of the new requirements in this requirement for additional information about how each of these parts are handled by the supplier.
	(6.3.4.14) Once the Software Validation Report has been established, verification shall address • that the Software Validation Report meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 6.3.4.8 to 6.3.4.11 and 7.7.4.7 to 7.7.4.11, • the internal consistency of the Software Validation Report. • The results shall be recorded in a Software Validation Verification Report.	The new requirements in this requirement is 6.3.4.10, 6.5.4.15, 6.5.4.16, 7.7.4.7, 7.7.4.9 and 7.7.4.10. The Software Validation Verification Report are not a new lifecycle documentation in this standard, so the report were already used by the supplier. The supplier only had to make sure that all of the added requirements to this report were fulfilled. See the new requirements to this requirement for additional information of how this requirement is handled by the supplier.
	(6.3.4.15) The Validator shall be empowered to require or perform additional reviews, analyses and tests.	Because of the widely extended responsibilities for this role, it can be hard for the Validator to be involved enough in each part of the process in order to produce the best result of a large process.
	(6.3.4.16) The software shall only be released for operation after authorization by the Validator.	The Validator always has to approve the process, otherwise there will not be any release, this has been the case for the supplier even before this update in standard. The release is stopped by the Validator if there is low quality, unclear test results, bad test results etc.
6.4 Software Assessment Comments: The assessment plan is a new requirement in this part of the standard. How the assessment process should proceed is overall a stricter process in the 2011 version of this standard [6].		

	(6.4.3) Output documents: 1. Software assessment Plan 2. Software assessment Report (was included in the earlier version of the standard) 3. Software assessment verification Report	This part of the standard were not included in the developing process of the generic product. It was read through in order to be aware about the requirements regarding the assessment and left for the Assessor to take responsibility for.
	(6.4.4.4) A Software Assessment Plan shall be written, under the responsibility of the Assessor, on the basis of the input documents from 6.4.2. Where appropriate, an existing documented generic Software Assessment Plan or procedure may be used. The requirement in 6.4.4.5 refers to the Software Assessment Plan.	The plan is send to the supplier in advance, in order for the supplier to be able to argue about the content in the plan. This way the supplier can be prepared of what to be assessed on and if something is added that should not be in the assessment plan, an argumentation can be done in order for the parts to be removed. The plan is then used as a template for the report. Even though this part of the standard is new, the method was used even before the update.
	(6.4.4.5) The Software Assessment Plan shall include the following scope: - aspects with which the assessment deals; - activities throughout the assessment process and their sequential link to engineering activities; - documents to be taken into consideration; - statements on pass/fail criteria and the way to deal with non-conformance cases; - requirements with regard to content and form of the Software Assessment Report.	This requirement is in the hand of the Assessor. Some parts of the Software Assessment Plan were deeper explained than necessary.
	(6.4.4.6) A Software Assessment Verification Report shall be written, under the responsibility of the Verifier, on the basis of the input documents from 6.4.2.	The Document Control Summary is followed by the supplier. These requirements were followed by the supplier even before the update in standard, but in the 2011 version, the Document Control Summary is clearer in "who to control what".

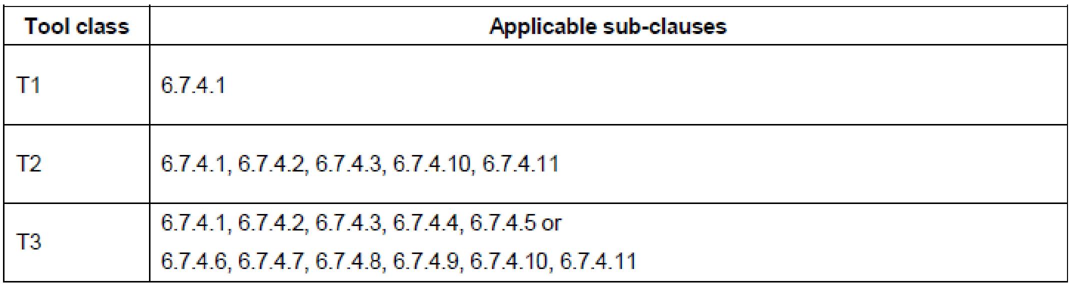
	(6.4.4.7) Once the Software Assessment Plan has been established, verification shall address • that the Software Assessment Plan meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 6.4.4.5, • the internal consistency of the Software Assessment Plan. • The results shall be recorded in a Software Assessment Verification Report.	The Software Assessment Plan is produced by the Assessor, so the supplier are not producing this plan, but they are making sure that it is produced. The Software Assessment Plan is the template for the Software Assessment Report.
	(6.4.4.10) The Assessor shall assess the configuration and change management system and the evidences on its use and application.	This requirement is in the responsibility of the Assessor, so the supplier are only clear about what the requirement includes, and then it is up to the Assessor to follow this part of the standard.
	(6.4.4.11) The Assessor shall review the evidence of the competency of the project staff according to Annex B and shall assess the organization for the software development according to 5.1.	see comment on 6.4.4.10.
	(6.4.4.12) For any software containing safety-related application conditions, the Assessor shall check for noted deviations, non-compliances to requirements and recorded non-conformities if these have an impact on safety, and make a judgment whether the justification from the project is acceptable. The result shall be stated in the assessment report.	see comment on 6.4.4.10.
	(6.4.4.13) The Assessor shall assess the verification and validation activities and the supporting evidence.	see comment on 6.4.4.10.
	(6.4.4.17) The Software Assessment Report shall meet the requirements of the Software Assessment Plan and provide a conclusion and recommendations.	In order to meet this requirement, the Software Assessment Report is almost identical to the Software Assessment Plan, only with additional results, like checkpoints etc.

	(6.4.4.18) The Assessor shall record his/her activities as a consistent base for the Software Assessment Report. These shall be summarized in the Software Assessment report.	see comment on 6.4.4.10.
6.5 Software Quality Assurance Comments: The Verifier should write the Quality Assurance Plan, which now also has to be written for SIL 0. The Quality Assurance plan has to be Verified [6].		
	(6.5.4.1) All the plans shall be issued at the beginning of the project and updated during the lifecycle.	The supplier has started some projects from the beginning, following the 2011 version of this standard. They have had all of the plans ready in the beginning of these projects. This has also been the case before this update in version.
	(6.5.4.3) A Software Quality Assurance Plan shall be written, under the responsibility of the Verifier, on the basis of the input documents from 6.5.2.	The supplier are following the Document Control Summary. Since the Software Quality Assurance Plan not are a new lifecycle document in this European standard, this new requirement did not cost any resources for the supplier to perform.
	(6.5.4.6) Quality assurance activities, actions, documents, etc. required by all normative sub-clauses of this European Standard shall be specified or referenced in the Software Quality Assurance Plan and tailored to the specific project.	To include quality assurance in the processes has been a requirement from the client to the supplier as the 2001 version of the standard was followed as well. This means that this part of the standard not were a new part for the supplier.
	(6.5.4.7) A Software Quality Assurance Verification Report shall be written, under the responsibility of the Verifier, on the basis of the input documents from 6.5.2.	The supplier are following the Document Control Summary. Even as this lifecycle document is new to this standard, the supplier did not experience this new requirement as resource costly.

	(6.5.4.8) Once the Software Quality Assurance Plan has been established, verification shall address • that the Software Quality Assurance Plan meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 6.5.4.4 to 6.5.4.6, • the internal consistency of the Software Quality Assurance Plan. • The results shall be recorded in a Software Quality Assurance Verification Report.	New requirements in this requirement is 6.5.4.6, 6.5.4.15, 6.5.4.16 and the Software Quality Assurance Verification Report. Since the Software Quality Assurance Plan not is a new requirement to this standard, this requirement was not resource costly. See 6.5.4.6, 6.5.4.15 and 6.5.4.16 for more comments on these new requirements.
	(6.5.4.15) Within the context of this European Standard, and to a degree appropriate to the specified software safety integrity level, traceability shall particularly address • traceability of requirements to the design or other objects which fulfill them, • traceability of design objects to the implementation objects which instantiate them, • traceability of requirements and design objects to the tests (component, integration, overall test) and analyses that verify them. • Traceability shall be the subject of configuration management.	The traceability has always been important to the supplier, so this new requirement in the standard does not cost the supplier resources in the change of version of the standard.
	(6.5.4.16) In special cases, e.g. pre-existing software or prototyped software, traceability may be established after the implementation and/or documentation of the code, but prior to verification/validation. In these cases, it shall be shown that verification/validation is as effective as it would have been with traceability over all phases.	Requirement tracing and testing has been done more or less all the time by the supplier. Requirement tracing and mapping against tests and test cases (verification) is a general requirement that has been used for a long time. The supplier has been, and still are doing improvements in this area during the development process.

<table>
<tr><td>6.6 Modification and change control</td><td>(6.6.2) Input documents:
• Software Quality Assurance Plan
• Software Configuration Management Plan
• All relevant design, development and analysis documentation
• Change Requests
• Change impact analysis and authorization</td><td>These new lifecycle documents were not experienced by the supplier to be resource costly. Most of these lifecycle documents were already used as the 2001 version of this standard were used.</td></tr>
<tr><td></td><td>(6.6.3) Output documents:
• All changed input documents
• Software Change records
• New Configuration records</td><td>The supplier made a list of all documents that could have a connection to the two versions of this standard. The software was not affected by the update in version. An assignment specification is describing the parts of the process that has to be performed. This process was done even before the update of this standard.</td></tr>
<tr><td></td><td>(6.6.4.2) All changes shall initiate a return to an appropriate phase of the lifecycle. All subsequent phases shall then be carried out in accordance with the procedures specified for the specific phases in accordance with the requirements in this European Standard.</td><td>This requirement does not affect the process of the supplier.</td></tr>
<tr><td colspan="3">6.7 Support tools and languages
Comments:
This part of the standard has made an extensive extension of requirements and structure. The tools are now divided into three groups (T1, T2 and T3) depending on how much they can affect the safety of the system in operation [6]. All tools used has to be approved by this standard. Related to this part is table 15 - Programming Languages.
It came in some comments regarding the purpose and the content of this table from the French and the Dutch national committees, during the maintenance work. The Dutch national committee argued that this table was useless since the obvious choice of language is C or C++. The French national committee suggested extension of this table. They suggested additional object oriented languages such as JAVA and Visual Basics.
JAVA was added to this table and Fortan 77 deleted. Table 15 is included in Appendix A, section 8.1.</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

	(6.7.1.1) The objective is to provide evidence that potential failures of tools do not adversely affect the integrated toolset output in a safety related manner that is undetected by technical and/or organizational measures outside the tool. To this end, software tools are categorized into three classes namely, T1, T2 & T3 respectively (see definitions in 3.1). When tools are being used as a replacement for manual operations, the evidence of the integrity of tools output can be adduced by the same process steps as if the output was done in manual operation. These process steps might be replaced by alternative methods if an argumentation on the integrity of tools output is given and the integrity level of the software is not decreased by the replacement. • (3.1.42) tool class T1 generates no outputs which can directly or indirectly contribute to the executable code (including data) of the software *NOTE T1 examples include: a text editor or a requirement or design support tool with no automatic code generation capabilities;* *configuration control tools.* • (3.1.43) tool class T2 supports the test or verification of the design or executable code, where errors in the tool can fail to reveal defects but cannot directly create errors in the executable software *NOTE T2 examples include: a test harness generator; a test coverage measurement tool; a static analysis tool.* • (3.1.44) tool class T3 generates outputs which can directly or indirectly contribute to the executable code (including data) of the safety related system *NOTE T3 examples include: a source code compiler, a data/algorithms compiler, a tool to change set-points during system operation; an optimizing compiler where the relationship between the source code program and the generated object code is not obvious; a compiler that incorporates an executable run-time package into the executable code.*	This part of the 2011 version of the standard was by far the most extensive part of the process. The supplier would prefer there to be some examples on this part of the process, in order to more effectively proceed the classification of tools. The explanation should include examples of how to present the tools and with tools to present in each group. The sorting process took approximately 160 hours per product to proceed. The supplier did not found any tools that were not used at all, or tools that where dangerous to the system/product for product 1 and 2. The standard indicates that this requirement regards all tools, but does not specific say that is it including all tools.



Tool class	Applicable sub-clauses
T1	6.7.4.1
T2	6.7.4.1, 6.7.4.2, 6.7.4.3, 6.7.4.10, 6.7.4.11
T3	6.7.4.1, 6.7.4.2, 6.7.4.3, 6.7.4.4, 6.7.4.5 or 6.7.4.6, 6.7.4.7, 6.7.4.8, 6.7.4.9, 6.7.4.10, 6.7.4.11

This table describes the standards connected to each group of tools.

	(6.7.2) Input documents • Tools specification or manual.	Each of the tools for both product 1 and 2 was complemented with a description of the purpose and the usage area. These descriptions and sorted list are used as a tool specification. See an example of a tool description in figure 4.1.
	(6.7.3) Output documents • Tools validation report (when needed see 6.7.4.4 or 6.7.4.6).	The product that uses one or more tools, for example: • Product 2: Refers all evaluations that are made for each tool. Each tool providing an analysis due to this standard, in the Tools Validation report. This work has been an extensive part of the change process.
	(6.7.4.2) The selection of the tools in classes T2 and T3 shall be justified (see 7.3.4.12). The justification shall include the identification of potential failures which can be injected into the tools output and the measures to avoid or handle such failures.	See comments for 6.7.3 and 6.7.4.3.

	(6.7.4.3) All tools in classes T2 and T3 shall have a specification or manual which clearly defines the behavior of the tool and any instructions or constraints on its use.	The work of execute this requirement has been an extensive work in the update of this standard. An example of how the specification of a tool in class T2 could look like is illustrated in figure 4.1. For a tool in class T3, there is more information and references to the documentation and also more about the mitigates for that tool.
	(6.7.4.4) For each tool in class T3, evidence shall be available that the output of the tool conforms to the specification of the output or failures in the output are detected. Evidence may be based on the same steps necessary for a manual process as a replacement for the tool and an argument presented if these steps are replaced by alternatives (e. g. validation of the tool). Evidence may also be based on • a suitable combination of history of successful use in similar environments and for similar applications (within the organization or other organizations), • tool validation as specified in 6.7.4.5, • diverse redundant code which allows the detection and control of failures resulting in faults introduced by a tool, • compliance with the safety integrity levels derived from the risk analysis of the process and procedures including the tools, • other appropriate methods for avoiding or handling failures introduced by tools. *NOTE 1 A version history may provide assurance of maturity of the tool, and a record of the errors / ambiguities associated with its use in the environment.* *NOTE 2 The evidence listed for T3 may also be used for T2 tools in judging the correctness of their results.*	This has been an extensive work in the process of updating version of the standard. See comments of 6.7.4.3 for more information on the description of the classification of the tools.

	(6.7.4.5) The results of tool validation shall be documented covering the following results: • a record of the validation activities; • the version of the tool manual being used; • the tool functions being validated; • tools and equipment used; • the results of the validation activity; the documented results of validation shall state either that the software has passed the validation or the reasons for its failure; • test cases and their results for subsequent analysis; • discrepancies between expected and actual results.	The Tool validation report has been produced according to this standard. See earlier comments in this section for more information regarding the extensive work of the tool classification.

	(6.7.4.6) Where the conformance evidence of 6.7.4.4 is unavailable, there shall be effective measures to control failures of the executable safety related software that result from faults that are attributable to the tool. *NOTE 1 An example is the generation of diverse redundant code which allows the detection and control of failures resulting in faults introduced by a translator.* *NOTE 2 As an example, the fitness for purpose of a non-trusted compiler can be justified as follows.* *The object code produced by the compiler has been subjected to a combination of tests, checks and analyses which are capable of ensuring the correctness of the code to the extent that it is consistent with the target Safety Integrity Level. In particular, the following applies to all tests, checks and analyses.* • *Testing has been shown to have a sufficiently high coverage of the implemented code. If there is any code unreachable by testing, it has been shown by checks or analyses that the function concerned is executed correctly when the code is reached on the target.* • *Checks and analyses have been applied to the object code and shown to be capable of detecting the types of errors which might result from a defect in the compiler.* • *No more translation with the compiler has taken place after testing, checking and analysis.* • *If further compilation or translation is carried out, all tests, checks and analyses will be repeated.*	The Validator and the Assessor are supporting the supplier with feedback, whether or not the description supports the connected process of the tool. The tool might not have the correct evidence of performance alone, but together with reviews of results and/or tests, any shortcomings can be found. Sometimes the suppliers only uses one complier, but since the process says that tests have to be performed after the complier, the motivation for only using one complier can be that any mistakes by the complier will be captured by these tests or later tests.
	(6.7.4.9) Where automatic code generation or similar automatic translation takes place, the suitability of the automatic Translator for safety-related software development shall be evaluated at the point in the development lifecycle where development support tools are selected.	All tools that are used by the supplier has been evaluated in the process of tool classification. This has been an extensive work, but the supplier experience that it is good to have the tools organized. The supplier did not exclude any tool in the classification process, and the tools used already had a good evaluating strategy for their tools used.

	(6.7.4.10) Configuration management shall ensure that for tools in classes T2 and T3, only justified versions are used.	All tools that were used for product 1 and 2 before the classification of the tool was founded to be justified. No tools had to take the 'Backdoor'.
	(6.7.4.11) Each new version of a tool that is used shall be justified (see Table 1). This justification may rely on evidence provided for an earlier version if sufficient evidence is provided that • the functional differences (if any) will not affect tool compatibility with the rest of the toolset, • the new version is unlikely to contain significant new, unknown faults. *NOTE Evidence that the new version is unlikely to contain significant new unknown faults may be based on a credible identification of the changes made, and on an analysis of the verification and validation actions performed.*	Most of the lifecycle documentations in Table A.1 is used for product 1 and 2, see the tables in Appendix A, section 8.1.
	(6.7.4.12) The relation between the tool classes and the applicable sub-clauses is defined within Table 1.	See comments on 6.7.4.11.
7 Generic Software Development		
7.1 Lifecycle and documentations for generic software Comments: These requirements are a clarifications about that the 49 lifecycle documents listed in table A.1 should be produced [6]. The waterfall model is not a requirement, the important purpose of these requirements is that the documents are produced and they all have fully traceability and consistency.		

<table>
<tr><td></td><td>(7.1.2.2) The sequence of deliverable documents as they are described in Table A.1 reflects an ideal linear waterfall model. This model is however not intended to be a reference in the sense of schedule and linkage of activities, as it would usually be difficult to achieve a strict compliance in practice. Phases can overlap but verification and validation activities shall demonstrate the consistency of inputs and outputs (documents and software) within and between the phases.
However, the main purpose of the documentation foreseen is to provide a description of the software itself, from the higher levels of abstraction down to the detailed refinements, in order to create a frame for the demonstration of the achieved safety as well as for future maintenance actions.</td><td>Product 1 and 2 are including all of the lifecycle documentations required according to this standard, but non of the products are structuring these documents according to the waterfall model, the documents are structured in an appropriate order for the use of the documents.</td></tr>
<tr><td colspan="3">7.2
Software Requirements
Comments:
Many of the requirements are the same in the updated and the 2001 version of this standard. The roles are clearer, who is doing what [6]. In the maintenance work with this standard, many parts where criticized to be unclear regarding levels of adequacy and limitations. Names are changed, like “Overall Software Test Specification” instead of “Software Requirements Test Specification” [6].</td></tr>
<tr><td></td><td>(7.2.2) Input documents:
• System Requirements specification
• System Safety Requirements Specification
• System Architecture Description
• External Interface Specifications (e.g. Software/Software Interface Specification, Software/Hardware Interface Specification)
• Software Quality Assurance Plan
• Software Validation Plan</td><td>All of these documents are fulfilled, see table A.1 in section 8.1. The new part of this standard was not experienced to be an extensive part of the changing process.</td></tr>
<tr><td></td><td>(7.2.3) Output documents:
• Software Requirements Specification
• Overall Software Test specification
• Software Requirements Verification Report</td><td>All of these documents are fulfilled, see table A.1 in section 8.1. This new part of this standard was not experienced to be an extensive part of the changing process.</td></tr>
</table>

	(7.2.4.1) A Software Requirements Specification shall be written, under the responsibility of the Requirements Manager, on the basis of the input documents from 7.2.2.	The supplier are following the Document Control Summary. They were following this document even before the update in version.
	(7.2.4.15) The Software Requirements Specification shall be supported by techniques and measures from Table A.2. The selected combination shall be justified as a set satisfying 4.8 and 4.9.	An approved combination of methods/techniques are used from table A.2, see Appendix A, section 8.1. The changes in this table were not experienced to be extensive by the supplier.
	(7.2.4.16) An Overall Software Test Specification shall be written, under the responsibility of the Tester, on the basis of the Software Requirements Specification.	All requirements in the Document Control Summary are followed. This requirement was not experienced by the supplier to be an extensive work.
	(7.2.4.18) The Overall Software Test Specification shall choose techniques and measures from Table A.7. The selected combination shall be justified as a set satisfying 4.8 and 4.9.	The supplier uses something called ”Engineering process”, which explains the whole chain, where you take a GP (Generic product) and make a GA (Generic Application), and then a SA (Specific Application) before installation. The tests that will be performed that will be done in the different parts of the process is documented in the Overall Test Specification. This means that the supplier is indirectly performing ’Performance Testing’ and really are performing ’Functional Testing’ and Black-box Testing’. ”Maximum-testing” is performed for product 2, but also for product 1 in some cases.
	(7.2.4.20) A Software Requirements Verification Report shall be written, under the responsibility of the Verifier, on the basis of the System Safety Requirements Specification, Software Requirements Specification, Overall Software Test Specification and Software Quality Assurance Plan.	The supplier is following the Document Control Summary and uses all lifecycle documents in table A.1, see Appendix A, section 8.1.

	(7.2.4.22) Once the Software Requirements Specification has been established, verification shall address • the adequacy of the Software Requirements Specification in fulfilling the requirements set out in the System Requirements Specification, the System Safety Requirements Specification and the Software Quality Assurance Plan, • that the Software Requirements Specification meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 7.2.4.2 to 7.2.4.15, • the adequacy of the Overall Software Test Specification as a test against the Software Requirements Specification, • the definition of any additional activity in order to demonstrate the correct coverage of not testable requirements; • the internal consistency of the Software Requirements Specification, • the adequacy of the Software Requirements Specification in fulfilling or taking into account the constraints between hardware and software. • The results shall be recorded in a Software Requirements Verification Report.	Software Requirement Specification is not a new lifecycle document in this standard, so this requirement was not experienced to be an extensive work.
7.3 Software Architecture Specification Comments: New requirements regarding the Validation process, pre-existing software, Software Integration Specification etc. are added to this section of the 2011 version. The architectural table A.3 with technicians is updated and includes more technologies and places greater demands on SIL 0 systems [6].		

	(7.3.3) Output documents: • Software Architecture Specification • Software Design Specification • Software Interface Specification • Software Integration Test Specification • Software/Hardware Integration Test Specification • Software Architecture and Design Verification Report	Software Interface Specification, Software Integration Test Specification and Software/Hardware Integration Test Specification are new lifecycle documents to this standard. The supplier are using these lifecycle documents, see table A.1, Appendix A, section 8.1. The extension of the lifecycle documents were not experienced to be an extensive work, many of these new documentations were used as the 2001 version of this standard where used as well.
	(7.3.4.1) A Software Architecture Specification shall be written, under the responsibility of the Designer, on the basis of the Software Requirements Specification.	The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
	(7.3.4.6) Software components shall • cover a defined subset of software requirements, • be clearly identified and independently versioned inside the configuration management system.	-

	(7.3.4.7) The use of pre-existing software shall be subject to the following restrictions. • For all software safety integrity levels the following information shall clearly be identified and documented: – the requirements that the pre-existing software is intended to fulfill; – the assumptions about the environment of the pre-existing software; – interfaces with other parts of the software. • For all software safety integrity levels the pre-existing software shall be included in the validation process of the whole software. (Included in EN50128:2001) • For software safety integrity levels SIL 3 or SIL 4, the following precautions shall be taken (Included in EN50128:2001): – an analysis of possible failures of the pre-existing software and their consequences on the whole software shall be carried out; – a strategy shall be defined to detect failures of the pre-existing software and to protect the system from these failures; – the verification and validation process shall ensure ∗ that the pre-existing software fulfills the allocated requirements, ∗ that failures of the pre-existing software are detected and the system where the pre-existing software is integrated into is protected from these failures, ∗ that the assumptions about the environment of the pre-existing software are fulfilled. • The pre-existing software shall be accompanied by a sufficiently precise (e.g. limited to the used functions) and complete description (i.e. functions, constraints and evidence). The description shall include hardware and/or software constraints of which the integrator shall be aware and take into consideration during application. In particular it forms the vehicle for informing the integrator of what the software was designed for, its properties, behavior and characteristics.	Product 2 is based on an old system, so there will be old software to take into account as updates are made. Parts of this requirement are not new to this standard. The supplier have been dealing with this part even before this update. They are and will continuously be doing improvements in this area, in order to have fully control of the pre-existing software in the system.

	NOTE Statistical evidence may be used in the validation strategy of the pre-existing software.	
	(7.3.4.13) The Software Architecture Specification shall take into account the requirements from 8.4.8 when the software is configured by applications data or algorithms.	This requirement is more a clarification of what to include in the Software Architecture Specification, than a new requirement. Therefore, no extra work was necessary for the supplier, due to this requirement. The Software Architecture is not new to this standard, and all of the requirements in 8.4.8, besides 8.4.8.8 were included in the 2001 version of this standard. 17.4.3 in the 2001 version.
	(7.3.4.14) The Software Architecture Specification shall choose techniques and measures from Table A.3. The selected combination shall be justified as a set satisfying 4.8 and 4.9.	See table A.3 in Appendix B, section 8.1.
	(7.3.4.15) The size and complexity of the developed software architecture shall be balanced.	'Balanced' is a interpreted definition. This seem to be an unnecessary requirement.
	(7.3.4.16) Prototyping may be used in any phase to elicit requirements or to obtain a more detailed view on requirements and their consequences.	Prototyping has been used by the supplier for at least 25 years. More or less used depending on complexity. For a small product, prototyping might be excluded. For product 1, different types has been used on prototypes. An example of this is to provide core functions and sort of "dry run" with simplified GA (Generic Application) and SA (Specific Application). An other example that is applied on product 2, is to only develop the A-side and then develop the B-side.
	(7.3.4.17) Code from a prototype may be used in the target system only if it is demonstrated that the code and its development and documentation fulfills this European Standard.	See comments on 7.3.4.16.

	(7.3.4.18) A Software Interface Specification for all Interfaces between the components of the software and the boundary of the overall software shall be written, under the responsibility of the Designer, on the basis of the Software Requirements Specification and the Software Architecture Specification.	The supplier had early demands from their Assessor to have all the interfaces, even internal, specified before the 2011 version of this standard were applied on the processes. These specifications may have other names and were sometimes merged. With the help of design tools, such as Doxygen, they can fully control the interfaces.

	(7.3.4.19) The description of interfaces shall address • pre/post conditions, • definition and description of all boundary values for all specified data, • behavior when the boundary value is exceeded, • behavior when the value is at the boundary, • for time-critical input and output data: – time constraints and requirements for correct operation, – management of exceptions. • allocated memory for the interface buffers and the mechanisms to detect that the memory cannot be allocated or all buffers are full, where applicable, • existence of synchronization mechanisms between functions (see e). All data from and to the interfaces shall be defined for the whole range of values defined by the type of the data, including the ranges which are not used when processed by the functions: • definition and description of all equivalence classes for all specified data and each software function using them, • definition of unused or forbidden equivalence classes. *NOTE The data type includes the following:* • *input parameters and output results of functions and/or procedures;* • *data specified in telegrams or communication packets;* • *data from the hardware.*	See comments on 7.3.4.18.
	(7.3.4.26) The selection of a coding standard shall be justified to the extent required by the software safety integrity level.	Coding standards has always been used by the supplier. for SIL 4, Coding standards was mandatory (M) in the 2001 version as well as in the 2011 version. See table A.4 in Appendix A, section 8.1.

	(7.3.4.31) The Software Integration Test Specification shall address the following: • it shall be shown that each software component provides the specified interfaces for the other components by executing the components together; • it shall be shown that the software behaves in an appropriate manner when the interface is subjected to inputs which are out of specification; • the required input data with their sequences and their values shall be the base of the test cases; • the anticipated output data with their sequences and their values shall be the basis of the test cases; • it shall be shown which results of the component test (see 7.5.4.5 and 7.5.4.7) are intended to be reused for the software integration test.	'Software Integration Test Plan' are in the 2011 version called 'Software Integration Test Specification'. This is for the supplier covered by their test plan, therefore, no separate plan had to be created for this part. The supplier had to add some more information as the 2011 version of this standard were followed, but they were not experiencing this to be an extensive work.
	(7.3.4.36) The Software/Hardware Integration Test Specification shall address the following: • it shall be shown that the software runs in a proper way on the hardware using the hardware via the specified hardware interfaces; • it shall be shown that the software can handle hardware faults as required; • the required timing and performance shall be demonstrated; • the required input data with their sequences and their values shall be the basis of the test cases; • the anticipated output data with their sequences and their values shall be the basis of the test cases; • it shall be shown which results of the component test (see 7.5.4.5) and of the software integration test (see 7.6.4.3) are intended to be reused for the software/hardware integration test.	See comments on 7.3.4.31.

7.4 Component Design Comments: ’Software module’ has changed name to: ’Software Component’.		
	(7.4.4.1) For each component, a Software Component Design Specification shall be written, under the responsibility of the Designer, on the basis of the Software Design Specification.	The supplier is following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
	(7.4.4.8) The Software Component Test Specification shall be written in accordance with the generic requirements established for a Test Specification (see 6.1.4.4). (6.1.4.4) Each Test Specification shall document the following: – test objectives; – test cases, test data and expected results; – types of tests to be performed; – test environment, tools, configuration and programs; – test criteria on which the completion of the test will be judged; – the criteria and degree of test coverage to be achieved; – the roles and responsibilities of the personnel involved in the test process; – the requirements which are covered by the test specification; – the selection and utilization of the software test equipment;	Requirement 6.1.4.4 was earlier divided into three requirements [6]. The requirement is also much clearer in the 2011 version. The Software Component Test Specification was named Software Module Test Specification in the 2001 version. This new requirement was not experienced to be an extensive work.
	(7.4.4.10) The Software Component Test Specification shall choose techniques and measures from Table A.5. The selected combination shall be justified as a set satisfying 4.8 and 4.9.	The supplier are using approved combinations of techniques for product 1 and 2 from table A.5 (see Appendix A, section 8.1). Product 1 will implement ’Formal proof’ to their system. The update of table A.5 was not experienced to be an extensive work.

	(7.4.4.11) A Software Component Design Verification Report shall be written, under the responsibility of the Verifier, on the basis of the Software Design Specification, Software Component Design Specification and Software Component Test Specification.	The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
7.5 Software Component Implementation and Testing Comments: Implementation are introduced as a new step in the process in this section. The roll Implementer are also new to this standard [6].		
	(7.5.2) Input documents: • Software Component Design Specification; • Software Component Test Specification;	'Module' are changed to 'Component' in this European standard, but the specifications are not new to this standard.
	(7.5.4.1) The Software Source Code shall be written under the responsibility of the Implementer on the basis of the Software Component Design Specification. Requirements from 7.5.4.2 to 7.5.4.4 refer to the software source code.	The 'Software Source Code' documentation are not a new requirement to this standard. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
	(7.5.4.5) A Software Component Test Report shall be written, under the responsibility of the Tester, on the basis of the Software Component Test Specification and the Software Source Code.	The 'Software Component Test Report' was named 'Software Module Test Report' in the 2001 version. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
	(7.5.4.8) A Software Source Code Verification Report shall be written, under the responsibility of the verifier, on the basis of the Software Component Design Specification, the Software Component Test Specification and the Software Source Code.	The 'Software Source Code Verification Report' are not a new lifecycle documentation. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.

	(7.5.4.10) After the Software Source Code and the Software Component Test Report have been established, verification shall address • the adequacy of the Software Source Code as an implementation of the Software Component Design Specification (included in EN50128:2001, but now divided into more than one requirement), • the correct use of the chosen techniques and measures from Table A.4 as a set satisfying 4.8 and 4.9, • determining the correct application of the coding standards (included in EN50128:2001, but now divided into more than one requirement), • that the Software Source Code meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17, as well as the specific requirements in 7.5.4.1 to 7.5.4.4, • the adequacy of the Software Component Test Report as a record of the tests carried out in accordance with the Software Component Test Specification.	Parts of this requirement were included in the 2001 version of this standard as well. The supplier are using approved combinations of table A.4 (see Appendix A, section 8.1), and therefore no new documentations due to this requirement.
7.6 Integration Comments: Integrator is a new roll to this standard. The Test script for automatic tests and the reached 'test-coverage' shall be evaluated and described [6]. 'Modules' are changed to 'Components' in this standard.		
	(7.6.2) Input documents: • Software/Hardware Integration Test Specification • Software Integration Test Specification	These lifecycle documentations were named 'Software/Hardware Integration Test Plan' and 'Software Integration Test Plan' in the 2001 version of this standard.
	(7.6.4.3) A Software Integration Test Report shall be written, under the responsibility of the Integrator, on the basis of the Software Integration Test Specification.	This lifecycle documentation is not new to this standard. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.

	(7.6.4.7) A Software/Hardware Integration Test Report shall be written, under the responsibility of the integrator, on the basis of the Software/Hardware Integration Test Specification.	This lifecycle documentation is not new to this standard. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
	(7.6.4.11) A Software Integration Verification Report shall be written, under the responsibility of the Verifier, on the basis of the Software and Software/Hardware Integration Test Specifications and the corresponding test reports.	The supplier has all the time been having this part integrated with what they called 'test specification' and 'test report', which covers the independent system tests.
	(7.6.4.12) The Software Integration Verification Report shall be written in accordance with the generic requirements established for a Verification Report (see 6.2.4.13). (6.2.4.13) Each Software Verification Report shall document the following: – the identity and configuration of the items verified, as well as the Verifier names; – items which do not conform to the specifications; – components, data, structures and algorithms poorly adapted to the problem; – detected errors or deficiencies; – the fulfillment of, or deviation from, the Software Verification Plan (in the event of deviation the Verification Report shall explain whether the deviation is critical or not); – assumptions if any; – a summary of the verification results.	See comments on 7.6.4.11.
7.7 Overall Software Testing/Final Validation Comments: The Validator can now make the Tester perform extra tests, that has been specified by the Validator [6]. The Validator also has more requirements to follow, added in this section of this standard. Harder requirements on how the reporting should look like and be performed and that the system is validated in a real environment, not only simulated.		

	(7.7.4.1) An Overall Software Test Report shall be written, under the responsibility of the Tester, on the basis of the Overall Software Test Specification.	See comments on 7.6.4.11.
	(7.7.4.3) The Validator shall specify and perform supplementary tests on his discretion or have them performed by the Tester. While the Overall Software Tests are mainly based on the structure of the Software Requirements Specification, the added value the Validator shall contribute, are tests which stress the system by complex scenarios reflecting the actual needs of the user.	The Validator of the supplier are examine the test specification and then the test results. Sometimes, a request of execute extra tests is a result of this examination. This technique was used by the supplier before the update of this standard as well.
	(7.7.4.6) A Software Validation Report shall be written, under the responsibility of the Validator, on the basis of the Software Validation Plan.	This lifecycle documentation is not new to this standard. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
	(7.7.4.7) The Software Validation Report shall be written in accordance with the generic requirements established for the Validation Report (see 6.3.4.7 to 6.3.4.11).	Requirement 6.3.4.7 and 6.3.4.10 are new requirements in this standard, the content of these requirements are not including any new requests from the supplier. These requirements are only a clarification of what to be done in the area of this lifecycle document. The specific lifecycle document are not new to this standard and the supplier are following the Document Control Summary. This new requirement was not experienced to be an extensive work.
	(7.7.4.9) The Software Validation Report shall contain the confirmation that each combination of techniques or measures selected according to Annex A is appropriate to the defined software safety integrity level. It shall contain an evaluation of the overall effectiveness of the combination of techniques and measures adopted, taking account of the size and complexity of the software produced and taking into account the actual results of testing, verification and validation activities.	The Validation report are extended as the 2011 version of this standard are applied. The scope are the same, but in the extended version, the traceability to the standard are improved.

	(7.7.4.10) The following shall be addressed in the Software Validation Report: • documentation of the identity and configuration of the software; • statement of appropriate identification of technical support software and equipment; • statement of appropriate identification of simulation models used; • statement about the adequacy of the Overall Software Test Specification; • collection and keeping track of any deviations found; • review and evaluation of all deviations in terms of risk (impact); • a statement that the project has performed appropriate handling of corrective actions in accordance with the change management process and procedures and with a clear identification of any discrepancies found; • statement of each restriction given by the deviations in a traceable way; • a conclusion whether the software is fit for its intended application, taking into account the application conditions and constraints.	See comments on 7.7.4.9.
	(7.7.4.12) A Release Note which accompanies the delivered software shall include all restrictions in using the software. These restrictions are derived from • the detected errors, • non-compliances with this European Standard, • degree of fulfillment of the requirements, • degree of fulfillment of any plan.	The supplier has used Release Notes for at leased 25 years, they call it PVI (Product Version Information). In this Release Note the supplier are presenting the versions and subversions that will be released. Any limitations and mistakes that are included in this release are documented as well.
8 Systems configured by application data or algorithms: systems configured by application data or algorithms Comments: The requirements of the development, validation and verification process are harder and extended for the parameter set.		

	(8.4.1.1) An Application Preparation Plan shall be written, under the responsibility of the Requirements Manager or Designer, on the basis of the input documents from 8.2.	This documentation has been produced by the supplier for a long time. It is an extensive work to produce the In-depth risk analysis and collaboration with the users of the systems.
	(8.4.1.2) An Application Preparation Plan shall be produced in order to define and detail the application development process, including all the activities, deliverables and roles in charge of them. It can be produced either for each specific application or for a class of specific applications, i.e. for a generic application.	See comments on 8.4.1.1.
	(8.4.1.3) The Application Preparation Plan shall define a documentation structure for the application preparation process.	See comments on 8.4.1.1.
	(8.4.1.4) The Application Preparation Plan shall choose techniques and measures from Table A.11. The selected combination shall be justified as a set satisfying 4.8 and 4.9.	Since product 1 and 2 are Generic Products, table A.11 is not used (see Appendix A, section 8.1). The Assessor has approved this justification, which means that this new requirement are not used by the supplier.
	(8.4.1.6) The Application Preparation Plan shall include verification and validation activities to ensure that the application data/algorithms are complete, correct and compatible with each other and with the generic application, and to provide evidence that the application conditions of the generic application are met. These verification and validation activities and evidence can be replaced by verification and validation performed on the tools that produce the application data/algorithms. The results are gathered together in the Application Preparation Verification Report and the Application Test Report.	See comments on 8.4.1.1.
	(8.4.1.8) A risk analysis shall be carried out covering the application development process, including the application tools and procedures, in order to validate the Application Preparation Plan and to meet the required software safety integrity level. The Application Preparation Plan shall include the risk analysis.	See comments on 8.4.1.1.

	(8.4.1.13) Once the Application Preparation Plan has been established, verification shall address • that the Application Preparation Plan meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 8.4.1.2 to 8.4.1.11, • the internal consistency of the Application Preparation Plan. • The results shall be recorded in an Application Data/Algorithms Verification Report.	See comments on 8.4.1.1.
	(8.4.1.14) The implementation of the Application Preparation Plan shall be verified and validated for each specific application.	See comments on 8.4.1.1.
8.4.2 Application Requirements Specification	(8.4.2.1) An Application Requirements Specification shall be written, under the responsibility of the Requirements Manager, on the basis of the input documents from 8.2.	All of these steps have been made in one way or another for a long time by the supplier. Far back, before the current requirement for independence between TRV and suppliers, TRV (back then Banverket/SJ) was part of the verification chain. Nowadays, the steps of Application documentations are done internally by the supplier, and it was done even before 2011. The supplier are not always using the same names as are defined in this standard for the documentations, but the contents are the same.
	(8.4.2.3) The requirements related to the application data and algorithms processed by the generic software of the system shall be specified at this stage.	See comments on 8.4.2.1.
	(8.4.2.4) An Application Data/Algorithms Verification Report shall be written, under the responsibility of the Verifier, on the basis of the input documents from 8.2.	See comments on 8.4.2.1.

8.4.4 Application Data/Algorithms Production	(8.4.4.2) An Application Test Report shall be written, under the responsibility of the Tester, on the basis of the input documents from 8.2.	See comments on 8.4.2.1.
	(8.4.4.3) The Application Test Report shall document the correct and complete execution of the tests defined in Application Test Specification.	See comments on 8.4.2.1.
	(8.4.4.5) An Application Test Specification shall be written, under the responsibility of the Tester, on the basis of the input documents from 8.2.	See comments on 8.4.2.1.
	(8.4.4.6) The Application Test Specification shall specify tests to be carried out at intermediate or final stage of data/algorithms preparation, in order to ensure • coherency and completeness of data/algorithms with respect to application principles, • coherency and completeness of data/algorithms with respect to specific application architecture.	See comments on 8.4.2.1.
	(8.4.4.7) An Application Data/Algorithms Verification Report shall be written, under the responsibility of the Verifier, on the basis of the input documents from 8.2.	See comments on 8.4.2.1.
	(8.4.4.8) Once the Application Test Specification has been established, verification shall address • that the Application Test Specification meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 8.4.4.6, • the internal consistency of the Application Test Specification. • The results shall be recorded in an Application Data/Algorithms Verification Report.	See comments on 8.4.2.1.

8.4.5 Application Integration and Testing Acceptance	(8.4.5.2) An Application Test Specification shall be written, under the responsibility of the Tester, on the basis of the input documents from 8.2.	See comments on 8.4.2.1.
	(8.4.5.3) The Application Test Specification shall specify tests to be carried out to ensure • correct integration of data/algorithms on generic hardware and software, if needed, • correct integration of data/algorithms with complete installation.	See comments on 8.4.2.1.
	(8.4.5.4) An Application Data/Algorithms Verification Report shall be written, under the responsibility of the Verifier, on the basis of the input documents from 8.2.	See comments on 8.4.2.1.
	(8.4.5.5) Once the Application Test Specification has been established, verification shall address that the Application Test Specification meets the specific requirements in 8.4.5.3.	See comments on 8.4.2.1.
8.4.7 Application preparation procedures and tools	(8.4.7.2) Any compilation process shall be validated and assessed. It shall be noted that specialized compilers are usually necessary for the data and algorithm conversion.	See comments on 8.4.2.1.
	(8.4.7.3) All application data/algorithms and associated documentation for each specific application shall be subject to the software deployment requirements as specified in 9.1.	See comments on 8.4.2.1.
	(8.4.7.4) All application data/algorithms and associated documentation shall be subject to the software maintenance requirements specified in 9.2.	See comments on 8.4.2.1.
	(8.4.7.6) The Application Verification Report demonstrate the coverage and enforcement of the application conditions of the generic software and application tools.	See comments on 8.4.2.1.

<table>
<tr><td colspan="3">8.4.8
Development of Generic Software
Comments:
Release Note of the Generic Software and Application Tools of the Overall Software Testing/Final Validation phase of the generic software and application tools shall be subject to verification and validation. [6]</td></tr>
<tr><td></td><td>(8.4.8.8) The designers shall publish the Release Note of the generic software and application tools by the Overall Software Testing/Final Validation phase of the generic software and application tools. The contents of these documents shall be subject to verification and validation activities.
The following topics shall be addressed in the document “Application conditions of the generic software and application tools”:
• references to the user manuals of the generic software and application tools;
• any constraints on the application data/algorithms e.g. imposed architecture or coding rules to meet the safety integrity levels.</td><td>Application Information are written for product 1, for the Generic Product (GP) and the Generic Application (GA). These are inputs to the application engineering that provide application data. However, since the supplier has tool help (EBITool), where application engineer inputs plant data, many rules and restrictions are built into the tool. These built-in rules and constraints come partly from the products, partly from experience, but also from the Swedish Transport Administration’s regulations (TDok).</td></tr>
<tr><td colspan="3">9
Software deployment and maintenance</td></tr>
<tr><td colspan="3">9.1
Software deployment
Comments:
This part of the 2011 version of the standard is all new to this standard. New lifecycle documentations are Software release and deployment plan, Software deployment manual, Release notes, Deployment records, deployment verification report. [6]</td></tr>
<tr><td></td><td>(9.1.2) Input documents:
• All design, development and analysis documents relevant to the deployment.</td><td>Requirements with the word ’relevant’ will not include any extra work for the companies using these requirement, since they have been using documentations that they argued to be ’relevant’ even before the update of this standard.</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

	(9.1.3) Output documents: • Software Release and Deployment Plan • Software Deployment Manual • Release Notes • Deployment Records; • Deployment Verification Report	The information has always existed in the suppliers process, but not as strict as it is stated in 2011, e.g. writes installation protocols containing all the details, e.g. if all balises are updated the same. The supplier also collect and refer to all reviews related to the installation. The content of these lifecycle documentations has always existed, but as mentioned, it is documented a little differently. It is likely that there will be a clearer link between the suppliers work and CENELEC in the reporting, but this also depends on the Assessor's requirements. But as above, you do not need to have the CENELEC names on the documents, but you need to map more clearly what supplier document that corresponds to which CENELEC document.
	(9.1.4.1) The deployment shall be carried out under the responsibility of the project manager.	See comments on 9.1.3.
	(9.1.4.2) Before delivering a software release, the software baseline shall be recorded and kept traceable under configuration management control. Pre-existing software and software developed according to a previous version of this European Standard shall also be included.	See comments on 9.1.3.
	(9.1.4.3) The software release shall be reproducible throughout the baseline lifecycle.	See comments on 9.1.3.
	(9.1.4.4) A Release Note shall be written, under the responsibility of the Designer, on the basis of the input documents from 9.1.2.	See comments on 9.1.3.
	(9.1.4.6) A Software Deployment Manual shall be written on the basis of the input documents from 9.1.2.	See comments on 9.1.3.
	(9.1.4.7) The Software Deployment Manual shall define procedures in order to correctly identify and install a software release.	See comments on 9.1.3.

	(9.1.4.8) In case of incremental deployment (i.e., deployment of single components), it is highly recommended for SIL 3 and SIL 4, and recommended for SIL 1 and SIL 2, that the software is designed to include facilities which assure that activation of incompatible versions of software components is excluded.	See comments on 9.1.3.
	(9.1.4.9) Configuration management shall ensure that no harm results from the co-presence of different versions of the same software components where it cannot be avoided.	Configuration management (CM) has been in the process for at least 25 years ago. The supplier has ’CM-people’ in all projects i.e. Generic Product (GP), Generic Application (GA) and Specific Application (SA), including projects installation and deployment. No direct difference before and after 2011.
	(9.1.4.10) A rollback procedure (i.e., capability to return to the previous release) shall be available when installing a new software release.	Rollback Procedure has been used in the process, long time before 2011. Rollback is a must because you can experience unforeseen things when commissioned.
	(9.1.4.11) The software shall have embedded self-identification mechanisms, allowing its identification at the loading process and after loading into the target. The self-identification mechanism should indicate version information for the software and any configuration data as well as the product identity. *NOTE The data within the code, containing the information about the software release, is recommended to be protected through coding (see Table A.3 “Error Detecting Codes”).*	Embedded self-identification mechanisms have been around for a long time but are being developed and improved.
	(9.1.4.12) A Deployment Record shall be written on the basis of the input documents from 9.1.2.	See comments on 9.1.3.
	(9.1.4.13) A Deployment Record shall give evidence that intended software has been loaded, by inspection of the embedded self-identification mechanisms (see 9.1.4.11). This record shall be stored among the delivered system related documents like other verifications and is part of the commissioning and acceptance.	See comments on 9.1.3 and 9.1.4.11.

	(9.1.4.14) The deployed software shall be traceable to delivered installations. *NOTE This is of special importance when critical faults are discovered and need to be corrected in more than one installation.*	See comments on 9.1.3.
	(9.1.4.15) Diagnostic information shall be provided by the software, as part of fault monitoring.	Diagnostic information has been available before 2011 but is constantly being developed and improved.
	(9.1.4.16) A Deployment Verification Report shall be written, under the responsibility of the Verifier, on the basis of the input documents from 9.1.2.	See comments on 9.1.3.
	(9.1.4.17) Once the Software Deployment Manual has been established, verification shall address • that the Software Deployment Manual meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 9.1.4.7, • the internal consistency of the Software Deployment Manual.	See comments on 9.1.3.
	(9.1.4.18) Once the Deployment Record has been established, verification shall address • that the Deployment Record meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 9.1.4.13, • the internal consistency of the Deployment Record.	See comments on 9.1.3.

	(9.1.4.19) Once the Release Note has been established, verification shall address • that the Release Note meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 9.1.4.5, • the internal consistency of the Release Note. • The results shall be recorded in a Deployment Verification Report.	See comments on 9.1.3.
	(9.1.4.20) Measures shall be included in the software package to prevent or detect errors occurring during storage, transfer, transmission or duplication of executable code or data. The executable code is recommended to be coded (see Table A.3 “Error Detecting Codes”) as part of checking the integrity of the code in the load process.	'Error detecting codes' are used by the supplier for both product 1 and 2. It was used for these products as the 2001 version of this standard was followed as well (see table A.3, section 8.1), which means that this new requirement did not result in any extra work for the supplier in the update process.
9.2 Software Maintenance Comments: Before a maintenance work is started, it has to be decided, whether the maintenance work is 'major' or 'minor'. If this can not be decided by the company, the Assessor has make the decision. A new output document is 'Software Maintenance Verification Report'.		
	(9.2.3) Output documents: • Software Maintenance Plan • Software Change Records; • Software Maintenance Records • Software Maintenance Verification Report	The content of the Software Maintenance Verification Report is required but is documented in different documents with other names, e.g. verification of all changes in document and/or code. Tests are done on the entire system and component tests are done on modified parts.
	(9.2.4.5) A Software Maintenance Plan shall be written on the basis of the input documents from 9.2.2.	This lifecycle documentation is not new to this standard. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.

	(9.2.4.7) A Software Maintenance Record shall be written on the basis of the input documents from 9.2.2.	This lifecycle documentation is not new to this standard. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
	(9.2.4.9) A Software Change Record shall be written on the basis of the input documents from 9.2.2.	This lifecycle documentation is not new to this standard. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
	(9.2.4.11) A Software Maintenance Verification Report shall be written, under the responsibility of the Verifier, on the basis of the input documents from 9.2.2.	See comments on 9.2.3.
	(9.2.4.12) Once the Software Maintenance Plan has been established, verification shall address • that the Software Maintenance Plan meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 9.2.4.6, • the internal consistency of the Software Maintenance Plan.	This lifecycle documentation is not new to this standard. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
	(9.2.4.13) Once the Software Maintenance Record has been established, verification shall address • that the Software Maintenance Record meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 9.2.4.8, • the internal consistency of the Software Maintenance Record.	This lifecycle documentation is not new to this standard. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.

	(9.2.4.14) Once the Software Change Record has been established, verification shall address • that the Software Change Record meets the general requirements for readability and traceability in 5.3.2.7 to 5.3.2.10 and in 6.5.4.14 to 6.5.4.17 as well as the specific requirements in 9.2.4.10, • the internal consistency of the Software Change Record.	This lifecycle documentation is not new to this standard. The supplier are following the Document Control Summary (see section 8.1.2). This new requirement was not experienced to be an extensive work.
	(9.2.4.19) For each reported problem or enhancement a safety impact analysis shall be made.	Over the past 7-8 years, The supplier has made a safety impact analysis integrated into their bug reporting system (NCR management). In addition, an assessment of remaining NCRs is made prior to release.
	(9.2.4.20) For software under maintenance, mitigation actions proportionate to the identified risk shall be taken in order to ensure the overall integrity of the system whilst the reported problems are investigated and corrected.	-

8.3 Appendix C - Interviews

Interview with the supplier:

1. Where has it been a problem in the update of this European standard? Which main areas has been the most extensive ones?
2. Which parts in the process has been the most time consuming/expensive?
3. Could the update in standard been done in a better way? This for saving money and time.
4. Is there any parts of the 2011 version of this European standard that you think that you could have done in a better way? Thus, the standard does not suggest the safest/best way to work.
5. Is there any parts of the new version of this European standard that have been unclear? Parts where you have not known have to proceed. Thus, parts of the standard that should be changed?
6. Short term and long term costs. Where have it been expensive to update version of this European standard, but it will probably not be expensive in the long run? This as a comparison to when the 2001 version of the standard was used.
7. How has the work of updating the version of this standard proceeded?
8. Have the new formation of the organization been working as expected? Did you already have the resources needed within the organization? Or it was necessary for resources to be taken in from outside because of this part of this standard?
9. Were there any rolls that were harder than others to assign?
10. Have you been able to keep the role division?
11. Positive/Negative affects by the update of this standard?
12. Have you seen any positive affects by the extended traceability?
13. Has the new requirements on SIL 0 products affected you in terms of time and resources?
14. Do you experiencing that the requirements are clearer in the new version of this standard? Is some parts of the standard being misjudged? Or have you experienced that the standard is crystal clear?
15. How has it been to work with the two versions of the standard side by side throughout the transition?
16. How has the tools been affected by the update in standard? Has it been an extensive work to sort the tools in the update of the version in this standard?

Questions asked later on:

1. Can I look at an example of the documented argumentation that is approved by the assessor when it comes to the tables/parts of the standard that does not have to be used since the product is a generic product?
2. Who is responsible for the standard to be followed on the parts that you declines your product from? Is it a totally new product that has to follow the standard or will this part of the standard not be used since this part is declined?
3. Due to 4.8 shall motivations to not use an approved combination according to the tables, be registered in the Quality Assurance Plan. Has this been done for product 2 according to Table 21 - Test Coverages for Code? How did this motivation sounded?
4. Did you have documented proves for the competence of the staff before this update in standard or did you have to do that? How did you work with 5.2.2.3 (and 5.2.2.4), meaning routines have to be maintenance in order to handle the competence of the staff?

5. Did you have documented proof of staff skills before the standard update or was this a process to perform? How do you work with 5.2.2.3 (and 5.2.2.4), then to maintain procedures to manage staff skills?

6. Is requirements 6.5.4.16 used sometime in the process? Thought especially about product 2, where there have been older software to take into account.

7. Is all the information regarding the tools documented together in the same document? I refer to 6.7.2 - Tool specification, contains the requirements of 6.7.4.2, 6.7.4.3 and 6.7.4.4, or are these demands put on different documents?

8. Do you fulfill the requirement described in 6.3.4.6? Thus, the Software Validation report identify the steps necessary to demonstrate the adequacy of the Overall Software Test Specification? Was this something you did in the past also, or is this all new to you?

9. How do you do to meet the requirement in 6.3.4.16? The validator accepts some documentation for this approval, or how does the approval process before the operation look like? Was this something you did in the past also, or is this some new?

10. Does the assessor create the Software Assessment Plan?

11. Have you started from scratch with EN50128:2011 in any project? Or have you just changed version to existing projects? If you are in such cases could you argue away 6.5.4.1, which states that any plan must be issued at the beginning of the project?

12. Did you use the output documents described in requirements 6.6.3 when the 2001 version of this standard was used? Or was this part new to you? If so, was it demanding to start using?

13. Is there any action strategy of the case as a testament to the T3 tools are not sufficient? Thus there is the opportunity to check for any failures? See requirements 6.7.4.6. And how this is done?

14. Did you use additional methods for "Overall Software Testing" that does not stand with table A.7? If this is the case, is this documented? Are there documentations of why parts of A.7 is not being fully carried out?

15. How do you translate SIL 0? What is a SIL 0 product to you?

16. According to 6.1.4.1, tests performed by other parties such as the requirements manager, designs and implementers shall be documented. Where does this happen if tests are performed by either of these parties? Was this documentation also done before the version update?

17. Which other standards did you use as a dictionary when EN 50128 lacked in good examples?

18. Are you documenting anywhere that you were only using 'Structured Methodology' partly in table A.3?

19. Do you follow requirement 7.3.4.16 and 7.3.4.17 on prototypes? Did you use prototypes in all phases? Can you tell me how this works? What this done even before the version update?

20. Did you write a 'Software Interface Specification' before the version update? If not, was this an extensive work? Do you fulfill requirement 7.3.4.18 and 7.3.4.19 connected to this specification?

21. 'Software Integration test plan' has changed its name to 'Software Integration Test Specification'. Did your 'Software Integration test plan' the new requirements in 7.3.4.31 and 7.3.4.36? Or did you have to update this plan in order to follow these new requirements?

22. Did you write a 'Software Integration Verification Report' and an 'Overall Software Test Report' before the version update? Or are this new documentations and was them in that case time consuming to create?

23. Can you explain how requirement 7.7.4.3 works in your process? Does your Validator perform extra tests?

24. Did the Software Validation Report contain what is described in 7.7.4.10 even before this version update of the standard? If not, was there any part that was extensive to add? Is the techniques/methods used, included in this report? Was this included as the 2001 version of this standard was used?

25. Did you use Release Note before the version update? Can you explain the content of a release note?

26. Did you write an Application Preparation Plan before the version update? If not, was is an extensive work to add this plan to the process? Was/Is it included a risk analysis in this plan?

27. What parts of the Application Requirement Specification, Data/Algorithms Verification Report, Application Test Specification and Application Test Report are what you do not have included in your process? Is this because you work with a GP or because you have chosen to delete parts of these documents? Did you also write a 'Data/Algorithms Verification Report' before the standard update?

28. Do you write and did you write 'Application conditions of the generic software and application tools' as described in 8.4.8.8. If this document is written and has been added after the version update, was it a demanding job to create this document?

29. Have you had any release when the 2011 version of the standard has been used? When the previous version of the standard was used, were the output documents presented in requirement 9.1.3 written? Or are these new documentations new in your process? Will the release go differently now that the 2011 version of the standard is used?

30. How do you work with 'Configuration management'? I guess you've used this earlier (before the update of the standard), but have you further developed this system now when more is required by the release process?

31. Have you used 'Rollback procedure', 'Diagnostic information' and 'embedded self-identification mechanisms' when the 2001 version of the standard was used?

32. Have you done any maintenance work on your products when the 2011 version is being followed? If not, have you looked at the differences in requirements for this section and prepare, or do you take care of it when you get there?

33. Did you write a 'Software Maintenance Verification Report' before the default update as well?

34. Did you have a 'safety impact analysis' (requirement 9.2.4.19) when the 2001 version of the standard was used? Have you done that when the process are following the 2011 version, yet?

35. Did you perform risk-minimized software maintenance measures when the earlier version of this standard was followed?

8.4 Appendix D - Roles, responsibilities and competencies

8.4.1 Requirements Manager

Responsibilities:

1. shall be responsible for specifying the software requirements
2. shall own the Software Requirements Specification
3. shall establish and maintain traceability to and from system level requirements
4. shall ensure the specifications and software requirements are under change and configuration management including state, version and authorization status
5. shall ensure consistency and completeness in the Software Requirements Specification (with reference to user requirements and final environment of application)
6. shall develop and maintain the software requirement documents

Key competencies:

1. shall be competent in requirements engineering
2. shall be experienced in application's domain
3. shall be experienced in safety attributes of application's domain
4. shall understand the overall role of the system and the environment of application
5. shall understand analytical techniques and outcomes
6. shall understand applicable regulations
7. shall understand the requirements of EN 50128

8.4.2 Designer

Responsibilities:

1. shall transform specified software requirements into acceptable solutions
2. shall own the architecture and downstream solutions 3. shall define or select the design methods and supporting tools
3. shall apply appropriate design principles and standards
4. shall develop component specifications where appropriate
5. shall maintain traceability to and from the specified software requirements
6. shall develop and maintain the design documentation
7. shall ensure design documents are under change and configuration control

Key competencies:

1. shall be competent in engineering appropriate to the application area
2. shall be competent in safety design principles
3. shall be competent in design analysis & design test methodologies
4. shall be able to work within design constraints in a given environment
5. shall be competent in understanding the problem domain
6. shall understand all the constraints imposed by the hardware platform, the operating system and the interfacing systems
7. shall understand the relevant parts of EN 50128

8.4.3 Implementer

Responsibilities:

1. shall transform the design solutions into data/source code/other design representations
2. shall transform source code into executable code/other design representation
3. shall apply safety design principles
4. shall apply specified data preparation/coding standards
5. shall carry out analysis to verify the intermediate outcome
6. shall integrate software on the target machine
7. shall develop and maintain the implementation documents comprising the applied methods, data types, and listings
8. shall maintain traceability to and from design
9. shall maintain the generated or modified data/code under change and configuration control

Key competencies:

1. shall be competent in engineering appropriate to the application area
2. shall be competent in the implementation language and supporting tools
3. shall be capable of applying the specified coding standards and programming styles
4. shall understand all the constraints imposed by the hardware platform, the operating system and the interfacing systems
5. shall understand the relevant parts of EN 50128

8.4.4 Tester

Responsibilities:

1. shall ensure that test activities are planned
2. shall develop the test specification (objectives & cases)
3. shall ensure traceability of test objectives against the specified software requirements and of test cases against the specified test objectives
4. shall ensure that the planned tests are implemented and specified tests are carried out
5. shall identify deviations from expected results and record them in test reports
6. shall communicate deviations with relevant change management body for evaluation and decision
7. shall capture outcomes in reports
8. shall select the software test equipment

Key competencies:

1. shall be competent in the domain where testing is carried out e.g. software requirements, data, code etc.
2. shall be competent in various test and verification approaches/methodologies and be able to identify the most suitable method in a given context
3. shall be capable of deriving test cases from given specifications
4. shall have analytical thinking ability and good observation skills
5. shall understand the relevant parts of EN 50128

8.4.5 Verifier

Responsibilities:

1. shall develop a Software Verification Plan (which may include quality issues) stating what needs verification and what type of process (e.g. review, analysis etc.) and test is required as evidence
2. shall check the adequacy (completeness, consistency, correctness, relevance and traceability) of the documented evidence from review, integration and testing with the specified verification objectives
3. shall identify anomalies, evaluate these in risk (impact) terms, record and communicate these to relevant change management body for evaluation and decision
4. shall manage the verification process (review, integration and testing) and ensure independence of activities as required
5. shall develop and maintain records on the verification activities
6. shall develop a Verification Report stating the outcome of the verification activities

Key competencies:

1. shall be competent in the domain where verification is carried out e.g. software requirements, data, code etc.
2. shall be competent in various verification approaches/methodologies and be able to identify the most suitable method or combination of methods in a given context
3. shall be capable of deriving the types of verification from given specifications
4. shall have analytical thinking ability and good observation skills
5. shall understand the relevant parts of EN 50128

8.4.6 Integrator

Responsibilities:

1. shall manage the integration process using the software baselines
2. shall develop the Software and Software/Hardware Integration Test Specification for software components based on the Designer's component specifications and architecture stating what the necessary input components, the sequence of integration activities and the resultant integrated components are
3. shall develop and maintain records on the integration activities
4. shall identify integration anomalies, record and communicate these to relevant change management body for evaluation and decision
5. shall develop a component and overall system integration report stating the outcome of the integration

Key competencies:

1. shall be competent in the domain where component integration is carried out e.g. relevant programming languages, software interfaces, operating systems, data, platforms, code etc.
2. shall be competent in various integration approaches/methodologies and be able to identify the most suitable method or combination of methods in a given context
3. shall be competent in understanding the design and functionality required at various intermediate levels

4. shall be capable of deriving the types of integration test from a set of integrated functions
5. shall have analytical thinking ability and good observation skills tending towards the system level perspective
6. shall understand the relevant parts of EN 50128

8.4.7 Validator

Responsibilities:

1. shall develop a system understanding of the software within the intended environment of application
2. shall develop a validation plan and specify the essential tasks and activities for software validation and agree this plan with the assessor
3. shall review the software requirements against the intended environment/use
4. shall review the software against the software requirements to ensure all of these are fulfilled
5. shall evaluate the conformity of the software process and the developed software against the requirements of this European Standard including the assigned SIL
6. shall review the correctness, consistency and adequacy of the verification and testing
7. shall check the correctness, consistency and adequacy of test cases and executed tests
8. shall ensure all validation plan activities are carried out
9. shall review and classify all deviations in terms of risk (impact), records and submits to the body responsible for Change Management and decision making
10. shall give a recommendation on the suitability of the software for intended use and indicate any application constraints as appropriate
11. shall capture deviations from the validation plan
12. shall carry out audits, inspections or reviews on the overall project (as instantiations of the generic development process) as appropriate in various phases of development
13. shall review and analyse the validation reports relating to previous applications as appropriate
14. shall review that developed solutions are traceable to the software requirements
15. shall ensure the related hazard logs and remaining non-conformities are reviewed and all hazards closed out in an appropriate manner through elimination or risks control/transfer measures
16. shall develop a validation report
17. shall give agreement/disagreement for the release of the software

Key competencies:

1. shall be competent in the domain where validation is carried out
2. shall be experienced in safety attributes of application's domain
3. shall be competent in various validation approaches/methodologies and be able to identify the most suitable method or combination of methods in a given context
4. shall be capable of deriving the types of validation evidence required from given specifications bearing in mind the intended application
5. shall be capable of combining different sources and types of evidence and synthesise an overall view about fitness for purpose or constraints and limitations of the application

6. shall have analytical thinking ability and good observation skills
7. shall have overall software understanding and perspective including understanding the application environment
8. shall understand the requirements of EN 50128

8.4.8 Assessor

Responsibilities:

1. shall develop a system understanding of the software within the intended environment of application
2. shall develop an assessment plan and communicate this with the safety authority and the client organisation (contracting body of the assessor)
3. shall evaluate the conformity of the software process and the developed software against the requirements of this European Standard including the assigned SIL
4. shall evaluate the competency of project staff and organisation for the software development
5. shall evaluate the verification and validation activities and the supporting evidence
6. shall evaluate the quality management systems adopted for the software development
7. shall evaluate the configuration and change management system and the evidence of its use and application
8. shall identify and evaluate in terms of risk (impact) any deviations from the software requirements in the assessment report
9. shall ensure that the assessment plan is implemented
10. shall carry out safety audits and inspections on the overall development process as appropriate at various phases of development
11. shall give a professional view on the fitness of the developed software for its intended use detailing any constraints, application conditions and observations for risk control as appropriate
12. shall develop an assessment report and maintain records on the assessment process

Key competencies:

1. shall be competent in the domain/technologies where assessment is carried out
2. shall have acceptance/licence from a recognised safety authority
3. shall have / strive to continually gain sufficient levels of experience in the safety principles and the application of the principles within the application domain
4. shall be competent to check that a suitable method or combination of methods in a given context have been applied
5. shall be competent in understanding the relevant safety, human resource, technical and quality management processes in fulfilling the requirements of EN 50128
6. shall be competent in assessment approaches/methodologies
7. shall have analytical thinking ability and good observation skills
8. shall be capable of combining different sources and types of evidence and synthesise an overall view about fitness for purpose or constraints and limitations on application
9. shall have overall software understanding and perspective including understanding the application environment
10. shall be able to judge the adequacy of all development processes (like quality management, configuration management, validation and verification processes)
11. shall understand the requirements of EN 50128

8.4.9 Project Manager

Responsibilities:

1. shall ensure that the quality management system and independency of roles according to 5.1 are in place for the project and progress is checked against the plans
2. shall allocate sufficient number of competent resources in the project to carry out the essential tasks including safety activities, bearing in mind the requisite independence of roles
3. shall ensure that a suitable validator has been appointed for the project as defined in EN 50128
4. shall be responsible for the delivery and deployment of the software and ensure that safety requirements from the stakeholders are also fulfilled and delivered
5. shall allow sufficient time for the proper implementation and fulfilment of safety tasks
6. shall endorse partial and complete safety deliverables from the development process
7. shall ensure that sufficient records and traceability is maintained in safety related decision making

Key competencies:

1. shall understand quality, competencies, organisational and management requirements of EN 50128
2. shall understand the requirements of the safety process
3. shall be able to weigh different options and understand the impact on safety performance of a decision or selected options
4. shall understand the requirements of the software development process
5. shall understand the requirements of other relevant standards

8.4.10 Configuration Manager

Responsibilities:

1. shall be responsible for the software configuration management plan
2. shall own the configuration management system
3. shall establish that all software components are clearly identified and independently versioned inside the configuration management system
4. shall prepare Release Notes which includes incompatible versions of software components

Key competencies:

1. shall be competent in software configuration management
2. shall understand the requirements of EN 50128

8.5 Appendix E - Documents

Proposal of the German National Committee:

SC9XA/DE0002/NCP
October 2013

EUROPEAN COMMITTEE FOR ELECTROTECHNICAL STANDARDIZATION

TECHNICAL COMMITTEE 9XA: Communication, signalling and processing systems

Proposal of the German National Committee for an extension of the validity of EN 50128:2001 for 3 years to be discussed at the CLC/SC9XA meeting to be hold on November 27th 2013 in Porto/Portugal

---

Background

During the development of future EN 50126-x, it has been detected that the standard EN 50128:2001 is listed in TSI “CCS” and “HS Rolling Stock” as dated reference.

EN 50128:2001 will be withdrawn in April 2014 and replaced by EN 50128:2011.

This will cause serious problems during the approval procedure of all railway applications which are covered by TSI “CCS” and “HS Rolling Stock”, as EN 50128 from 2001 and from 2011 have to be adhered at the same time.

This requires two different development and manufacturing processes which cannot be combined.

Additionally the application of EN 50128:2011 and the future implementation of the delayed EN50126-5 would result in process modification that have to be applied for only a limited period of time (until EN 50126-5 will be issued), if EN 50128:2011 were the only mandatory standard

Proposal:

The German National Committee proposes:

- SC9XA to ask BT for extension of the use of EN 50128:2001 by prolonging the dow of EN 50128:2011 for at least 3 years
- A joint initiative of TC 9 and its subcommittees to urgently ask ERA to ensure that during the development of TSIs, no conflicts to ENs due to dated references occur.
