CENELEC EN 50128

Sonawala Raj Prakashbhai, 1340138
High Integrity Systems (M.Sc.)
Frankfurt University of Applied Science
Frankfurt am Main, Germany
raj.sonawala@stud.fra-uas.de

***Abstract*—The standard CENELEC EN 50128 is used for software development in railway applications, ensuring that software is safe and protected. This standard is useful to provide the most important aspects of the software in railways for instance availability, reliability, maintainability.Initially, this standard was introduced in 2001 since then it is used for software development for railway-related applications. For fulfilling developments, maintenance and provision a list of necessaries is provided in railway control and monitoring systems. [1]. In this paper, I have provided crucial information on CENELEC EN 50128 and followed the software development life cycle.**

***Index Terms*—CENELEC EN 50128, Model Checking, Safety, European Committee standard, Safety Software Development Process**

I. Introduction

Fig 1 represents the scope of the various CENELEC standards and their purpose. There are mainly three standards available for the safety purpose of software and hardware. The CENELEC EN 50128 is the standard used for the development aspects of software in the rail sector. It provides procedures, principles, and measures for the software to [1] be considered secure in the rail sector.CENELEC basically is the European Committee for Electrotechnical Standardization issued first version of standard EN 50128 in the year 2001 . Later in the year 2011, This committee published the second version of it [1].

Software is one of the most important parts of every system. In history, there are many events had occurred which lead to injury, death, destruction, loss of vital equipment, and damage to the environment. To prevent such hazardous situations to occur it is recommended to follow the standards.EN acronym for European Norms. The CENELEC standards are taken as technical standards and all countries across Europe are mandatory to follow them and anticipated to provide them the place of national standards without any modification [1].

Fig 2 describes the structure of standard CENELEC EN 50128 according to the new version. The software quality assurance is composed by considering important aspects, for instance, quality management, the verification, validation, assessment. As described in image clause 7 that represents the development of generic software. Clause 8 concern more focus on application data and describe managing the data preparation process.

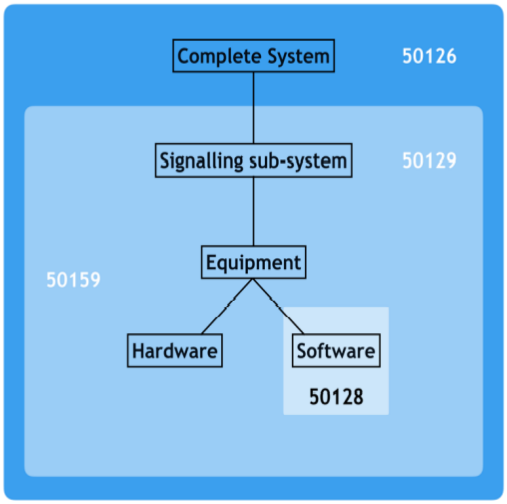


Fig. 1. Main standards applicable to railway system [3]

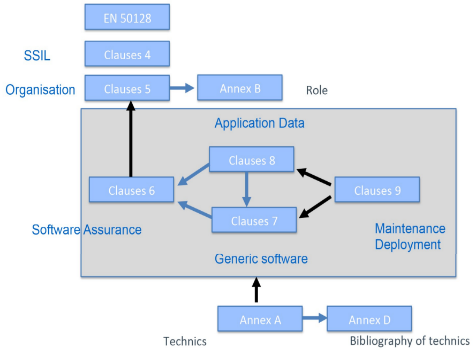


Fig. 2. CENELEC EN 50128:2011 [4]

II. Principles of the Standard

There are the following basic ten principles were used in standard EN 50128.

1) Top-down design approach
2) modularity
3) Verification of every phase of the development cycle
4) Verified Software components and Software component libraries

5) Scalable documentation and traceability
6) Auditable documents
7) Validation
8) Appraisal
9) Configuration management
10) Appropriate consideration of issues of organization and staff competence [1]

III. Safety Integrity Level (SILs)

SIL stands for Safety Integrity Level. A SIL is use to calculate of safety system performance, in terms of probability of failure on demand (PFD). [2] SIL is often scaled between 0 and 4. With value 4 SIL products have the most important impact on the protection while 0 value posses' software or product doesn't have any safety impact and thanks to that no requirements. within the new edition EN 50128, the bottom level of safety impact is considered as SIL 0 light. [3] Requirements are added to SIL 0 level, due to change in explanations. this alteration has made SIL 0 broadly misjudged, the meaning has changed within the remake, however, the name remained identical [2].

According to New Version of EN 50128:

- SIL 4: the disastrous impact
- SIL 3: community impact
- SIL 2: major emphasis on the protection of installation and production, or risk employees get injured.
- SIL 1: minor emphasis on the protection of the installation and production
- SIL 0: SIL1: Light and minimum level of safety effect [2]

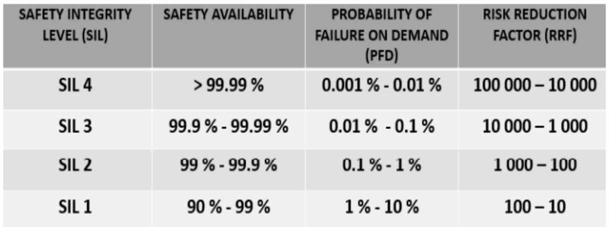
The selection of the safety level should be based on the risk related to the software being implemented in the system. For instance, if the risk is very high then, a high SIL level Safety requirement should be applied, on another hand, if a SIL level is low then low-level Safety requirements should be recommended to apply.

IV. How to Perform a SIL study

Directly measuring the SIL is not possible. A SIL study need to be carry out to calculate the SIL of the system and sub-system. There are some diverse ways to measure the SIL of the software. For the software application related to the standard EN 50128, this SIL study helps determined by checking the following things:

- Type of device
- Hardware architecture
- Voting logic
- Proof test intervals
- Required to meet the target Risk Reduction Factor (RRF) [2]

Steps required to perform SIL study:



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

Fig. 3. The interval of PFD and RRF for each SIL [2]

1) Divide and subdivide the SIF (Safety Instrumented Functions) into its components and architecture.
2) Measure the PFD (probability of failure on demand) of each component. In performing this task, the engineer needs a rate of failure from past data. There are many diverse formulas exist that can be used to perform the calculation of PFD [2]. For instance, rate= 1/MTBF = R/T where R is the number of failures and T is total time can be used to calculate PFD.
3) Combine all component PFD to figure out the SIF PFD, which is straightforwardly done by adding all the PFD of all components of the products.
4) Translate the entire PFD to RRF (Risk Reduction factor) and compare this to the expected SIL [2].

If the RRF had not fall into the expected figure in SIL, the 'weakest point' needs to be found among all the components of the product. 'Weakest point' means that component that contains the largest probability of failure. To bring the SIL of the software in the correct figure, it is crucial to fix the weakest point. [2]

V. Software development life cycle

It is important to follow the most scalable software development model to measure the complexity of the entire software and the strategy behind the development of the software can be estimated.

IN standard EN 50128, it is recommended to follow the V-model as an application development life-cycle. The V-model is the extension of the waterfall model which contains seven life cycle phases: requirements, architecture design, Design and Implementation, software and hardware integration, Validation Phase, Assessment Phase, Maintenance Phase. Additionally, there are two activities ongoing during lifecycle phases, verification, and validation of the product quality. [2]

Based on the scheme illustrated in fig 4 the "tailoring" of the EN 50128 life-cycle explanation.

1) Requirement Specification phase: In this stage of the V-model, The developer must make identification of system-level requirements for the software. Standard EN-50128 states that requirements should contain the subsequent aspects: absolute, consistency, accuracy, and simplicity.

2) Architecture Phase: The objective of this stage is to keep reference as a software requirements and would give rise to the architecture of software and focus on acquired these requirements and assess the significance of hardware and software integration for safety. Additionally, the other crucial objective of this stage is to examine and other options for achieving the state of SIL (Safety Integrity Level).

3) Design and Implementation phase: The main motive of the stage is to design and manufacture the software that contain the required SIL (Safety Integrity Levels) and select an appropriate tools that can help during verification, validation, assessment, and maintenance.

4) Software/Hardware Integration phase: Interaction between hardware and software is most important in this software development lifecycle and this phase of the development majorly focuses on how software and hardware meet each other efficient way and perform the required task.

5) Validation phase: In this phase, functionality is to specify performance quality, safety and reliability, and requirements for security and achieving standard SIL.

6) Assessment Phase: The main objective of this Phase is to get an idea that with software development lifecycle processes and at the end final outcome products meet the above Safety Integrity Levels. [2]

7) Maintenance Phase: The goal is to judge products, to ensure their correctness and accuracy concerning the products and standards provided are suitable to the stated SIL. [2]

Summarising :

The initial stage of the V model in CENELEC EN 50128 for instance System Definition and Application condition-stage concerns creating the system, subparts of systems, and required functions with Safety integrity level based on given characterization.further in the lifecycle developer needs to recognize requirements of software with the priority of achieving the defined functions with stated SIL. In the next stage, the appointment of system requirements stage assignment of working position and duties between team members by taking their experience and area of expertise as a reference. In Design and Implementation phase move the team towards the manufacturing process of the software with that, developer must need to prioritize his work in development of safety measures. On the right side,further in lifecylce stage consists of installation, gathering of all sub-systems, and perform operations. In installation phase, it involves the integration of the software with the hardware. the Verification stage assessment need to be perform on the installed software, so developer get knowledge that developed software works as anticipated. The acceptance stage verifies that developed software if fulfilling all requirements with assigned SIL level [2].

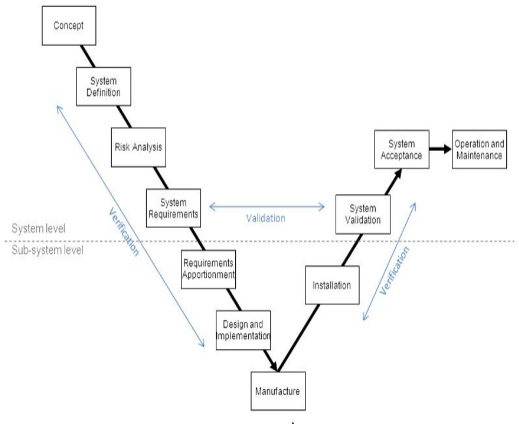


Fig. 4. The V-model in EN CELELEC [7]

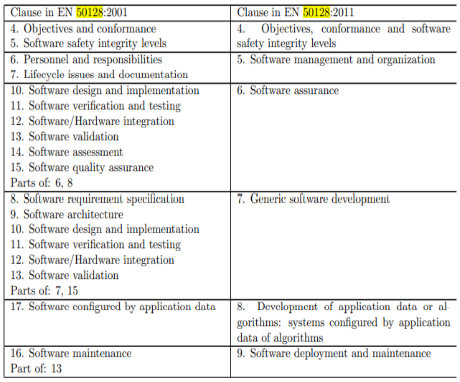
VI. Accidents occured because of software failures

A. Cambrian Line incident

In Morning Year 2017, October 17, four trains were traveled through the Cambrian Coast line during travel temporary speed restriction data was not being sent to the trains by ERTMS(European Rail Traffic Management System). Due to the absence of TSRs data, trains' speeds were increased level crossed 50 mph, significantly exceeding the temporary speed limits of 19 mph needed to give the required warning time for level crossing users. It was found in the investigation, during an automated signaling computer restart the previous evening the temporary speed restriction data was not uploaded but a display screen inaccurately showed the restrictions data was being loaded for transmission to trains.EN 50128 standard was used in the development of software in the Cambrian Line train. [5]

Major of reasons for Cambrian Line incident:

1) Due to the corrupted database, the GEST sub-system had entered in fault resulting temporary speed restriction data not being uploaded to RBC (Radio Block Centre) following software rollover.
2) Signallers had not received any indication of system failure.
3) Volatile memory was used to store temporary speed restrictions in the RBC. As a result during rollover temporary speed restrictions data was lost.
4) The design could not establish required level of safety integrity for validation of temporary speed restriction data provided to RBC.
5) GEST server software was unable to recognize and maintain the corruption of its database.
6) The safety related software requirements for the GEST software were inadequately defined. [5]



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

Fig. 5. Comparison of Available clauses in Old and New Version [2]

B. Spanish Rail Disaster

In the year 2013, July 24 the Alvia High speed- the train was traveling from Madrid to Ferrol, in the north-west of Spain, got crashed at high speed on a turn about 4 km(2.5 mi) outside of railway station at Santiago de Compostela. In this catastrophic, 143 passengers were injured and 79 died. In further investigation, it was found that the train had supposed to lower the speed before a turn comes around 80 km/h instead of that train was traveling at the speed of 200 km/h. In addition, According to the investigation report the driver was distracted with repeated mobile phone calls after the tunnel when the curve came in the way of the train the driver realized that he forget to lower down the speed of the train. And because of that, the train entered the curve at high speed, Eventually, the whole train went off the track. Due to that, this big accident happened.

From the observation of the Spanish Rail Disaster, It can be said that it mainly occurred because of human error. However, by removing the role of humans and including automatic speed controlling systems and Automatic brake systems in System requirements, this incident could be easily avoided. [5]

VII. New version EN 50128:2011

The old version of standard EN 50128:2001 was expired in the year 2014 and New version was launched in the year 2011. From the date 2017-04-25, the old version was started to get replaced by a new version. The new version contains a much higher number of requirements in comparison to an old version, Due to that, the new version's complexity and advantages also simultaneously get increased. The new version contains many restrictions, but it provided a better and clear structure of the software development lifecycle. [2]

Fig 5, it is visible that the new version only contains some main clauses and other clauses which existed in the old version have been completely getting changed in the new version. During the year 2011 to 2017, this period was the existence of two versions. However, all new projects started between these years follow the 2011 version. The new version contains requirements related to SIL and roles and responsibilities made much harder compared to an old version. [2]

VIII. Conclusion

This standard has been made mandatory across Europe, European Railway Authorities had decreased the complexity on the software side in railways. From the past Accidents that occurred basis on that, it can be said that Safety Standard EN 50128 is crucial and need to be followed strictly during software development in railways. Additionally, It is also observable that the new version of EN 50128:2011 is much more expensive to implement compared to EN 50128:2001. However, in many scenarios, the new standard was found much more efficient compared to the old version.

References

[1] "CENELEC Wikepedia," https://de.wikipedia.org/wiki/EN_50128, [Online; accessed Dec-2021]

[2] Åsa Nordström.The effect of the update of the European standard EN 50128.Juni 2017

[3] Jean-Louis Boulanger.ADACORE TECHNOLOGIES FOR CENELEC EN 50128:2011,1.1,October 2018

[4] Boulanger, J.-L., CENELEC 50128 and IEC 62279 standards. John Wiley Sons, 2015.

[5] "Loss of safety critical signalling data on the Cambrian Coast line" https://assets.publishing.service.gov.uk/media/5df8fa1be5274a08de86827d/R172019_191219_Cambrian_Coast_line.pdf

[6] "Disaster complexity and the Santiago de Compostela train derailment" www.ucm.es/data/cont/docs/1091-2016-01-17-2016%20SHULTZ%20GARCIA-VERY%20GESTEIRA%20ET%20AL%20SANTIAGO%20DE%20COMPOSTELA.pdf

[7] Alok katiyar.CENELC Standards for Signalling Applications https://slideplayer.com/slide/8750784/