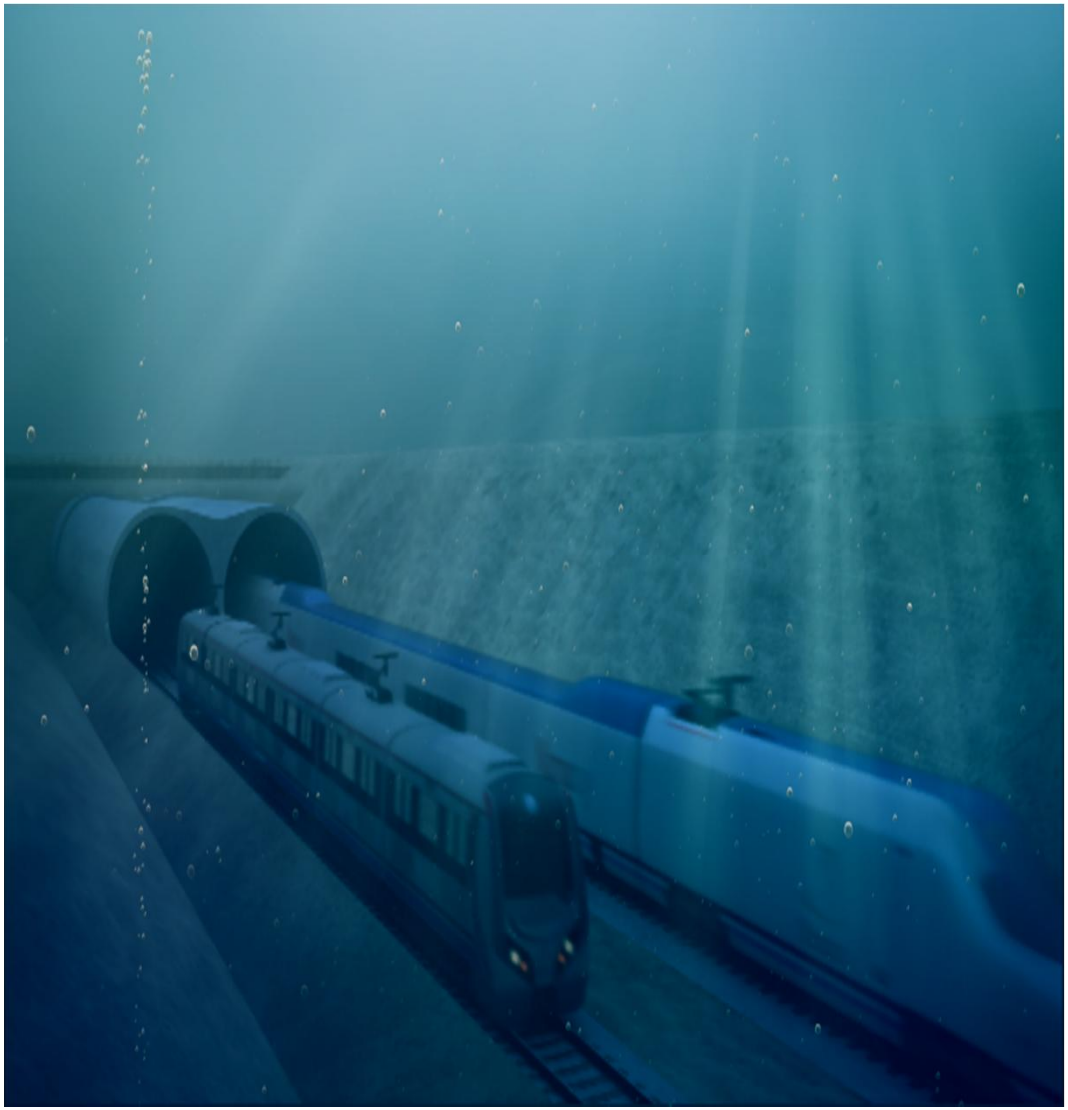
SIEMENS

Ada Europe 2015 / 24th June 2015

Software Development of Safety-Critical Railway Systems

Summary



1. Main Standards

2. Basic Concepts
   - Definition
   - System Lifecycle

3. EN50128
   - Introduction
   - Development Lifecycle
   - Organization
   - Features of ADA
   - How does Ada help?

SIEMENS

Chapter 1

MAIN STANDARDS

Main Standards

CENELEC 50126:1999: Railway Applications – Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS)

- Defines a process, based on the system life cycle and tasks within it, for managing RAMS.

CENELEC 50128:2001 or 50128:2011: Railway Applications – Communications, signalling and processing systems - Software for Railway Control and Protection Systems

- Specifies procedures and technical requirements for the development of programmable electronic systems for use in railway control and protection applications.

CENELEC 50129:2003: Railway Applications – Communications, signalling and processing systems - Safety Related Electronic Systems for Signalling

- Defines the conditions that shall be satisfied in order that a safety-related electronic railway system can be accepted as adequately safe for its intended application.
- Specifies procedures and information for identifying the credible failure modes of hardware components.

SIEMENS

Chapter 2

BASIC CONCEPTS

Basic Concepts: Definitions (I)

Reliability: probability that an item can perform a required function under given conditions for a given time interval (*t*1, *t*2).

Availability: ability of a product to be in a state to perform a required function under given conditions at a given instant of time or over a given time interval assuming that the required external resources are provided.

Maintainability: probability that a given active maintenance action, for an item under given conditions of use can be carried out within a stated time interval when the maintenance is performed under stated conditions and using stated procedures and resources.

Safety: freedom from unacceptable risk of harm.

Basic Concepts: Definitions (II)

Hazard: physical situation with a potential for human injury and/or damage to environment.

Risk: probable rate of occurrence of a hazard causing harm and the degree of severity of the harm.

Frequency of occurrence of a hazardous event		Severity			
		Insignificant	Marginal	Critical	Catastrophic
Frequency	Frequent	Undesirable	Intolerable	Intolerable	Intolerable
	Probable	Tolerable	Undesirable	Intolerable	Intolerable
	Occasional	Tolerable	Undesirable	Undesirable	Intolerable
	Remote	Negligible	Tolerable	Tolerable	Undesirable
	Improbable	Negligible	Negligible	Tolerable	Tolerable
	Incredebible	Negligible	Negligible	Negligible	Negligible

Tolerable Hazard Rate (THR): A rate which guarantees that the resulting risk does not exceed a target individual risk.

Basic Concepts: Definitions (III)

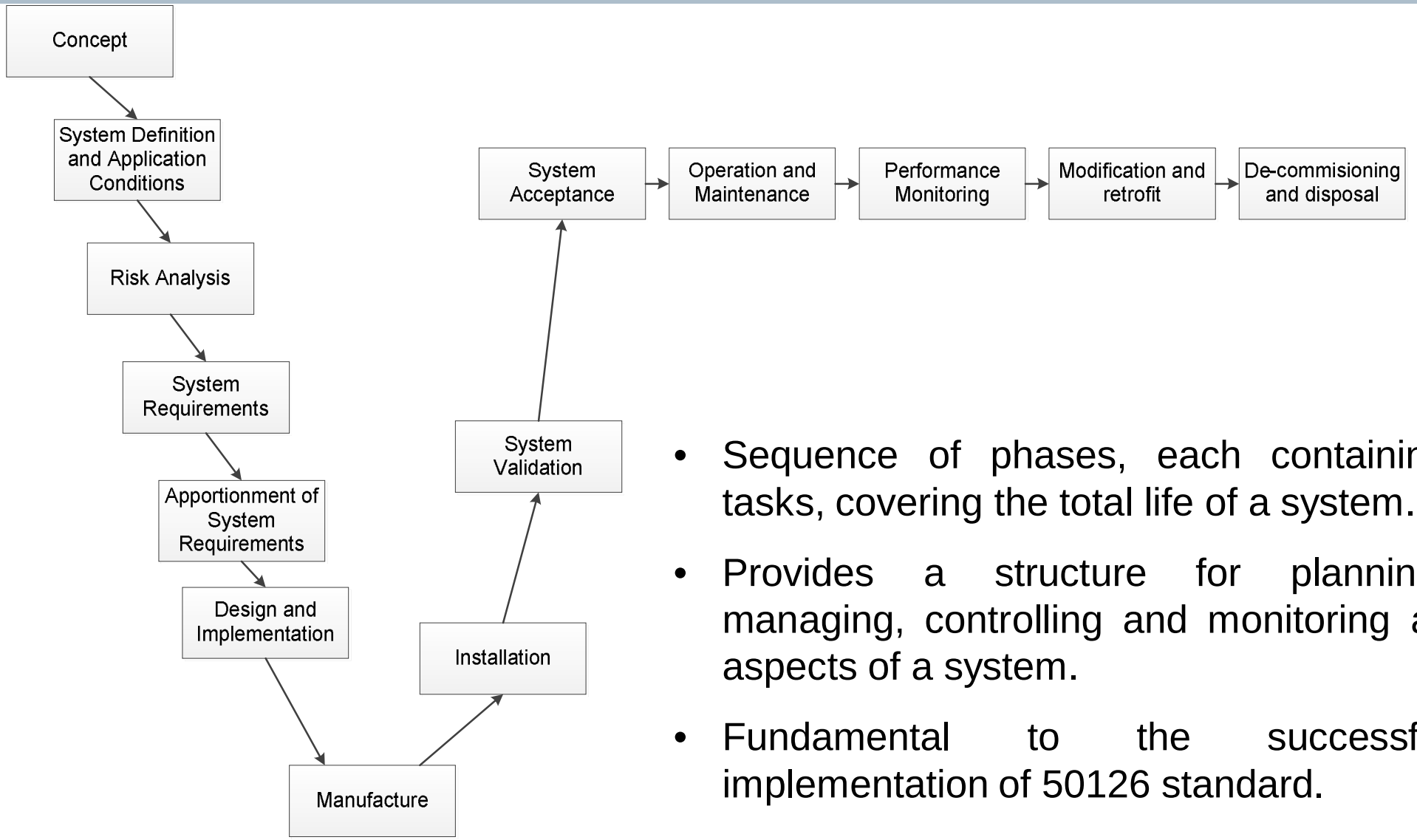
Safety Integrity Level (SIL): classification number which determines the techniques and measures that have to be applied in order to reduce residual software faults to an appropriate level.

THR per hour and per function	Safety Integrity Level
$10^{-9} \leq THR < 10^{-8}$	4
$10^{-8} \leq THR < 10^{-7}$	3
$10^{-7} \leq THR < 10^{-6}$	2
$10^{-6} \leq THR < 10^{-5}$	1

Validation: activity of demonstration, by analysis and test, that the product meets, in all respects, its specified requirements.

Verification: activity of determination, by analysis and test, that the output of each phase of the life-cycle fulfils the requirements of the previous phase.

Basic Concept: System Lifecycle (I)



- Sequence of phases, each containing tasks, covering the total life of a system.
- Provides a structure for planning, managing, controlling and monitoring all aspects of a system.
- Fundamental to the successful implementation of 50126 standard.

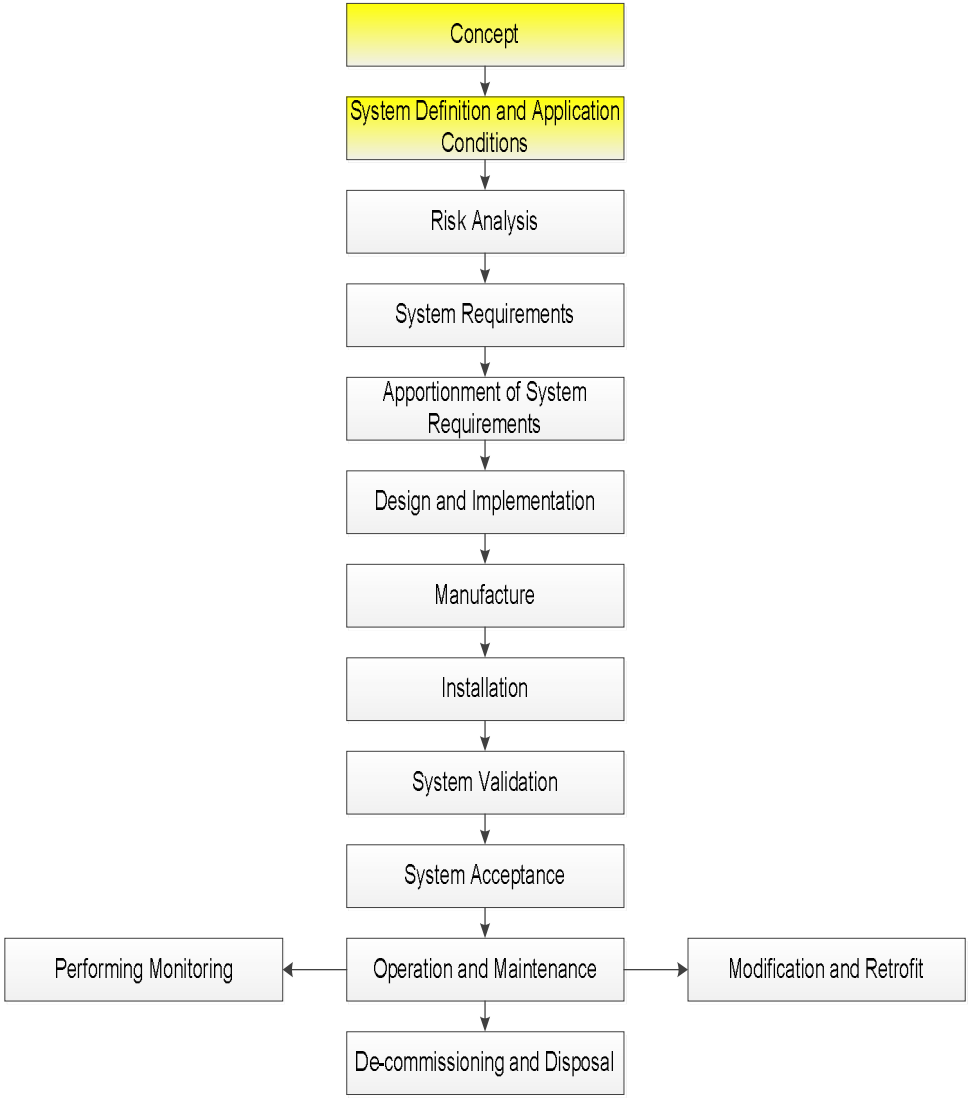
Basic Concept: System Life Cycle (I)

1. Concept

- Develop an initial understanding of the system.

2. System definition and application conditions

- Establish system mission profile.
- Define the boundary of the system.
- Establish the application conditions.
- Define the scope of system hazard analysis.



Basic Concept: System Life Cycle (II)

3. Risk Analysis

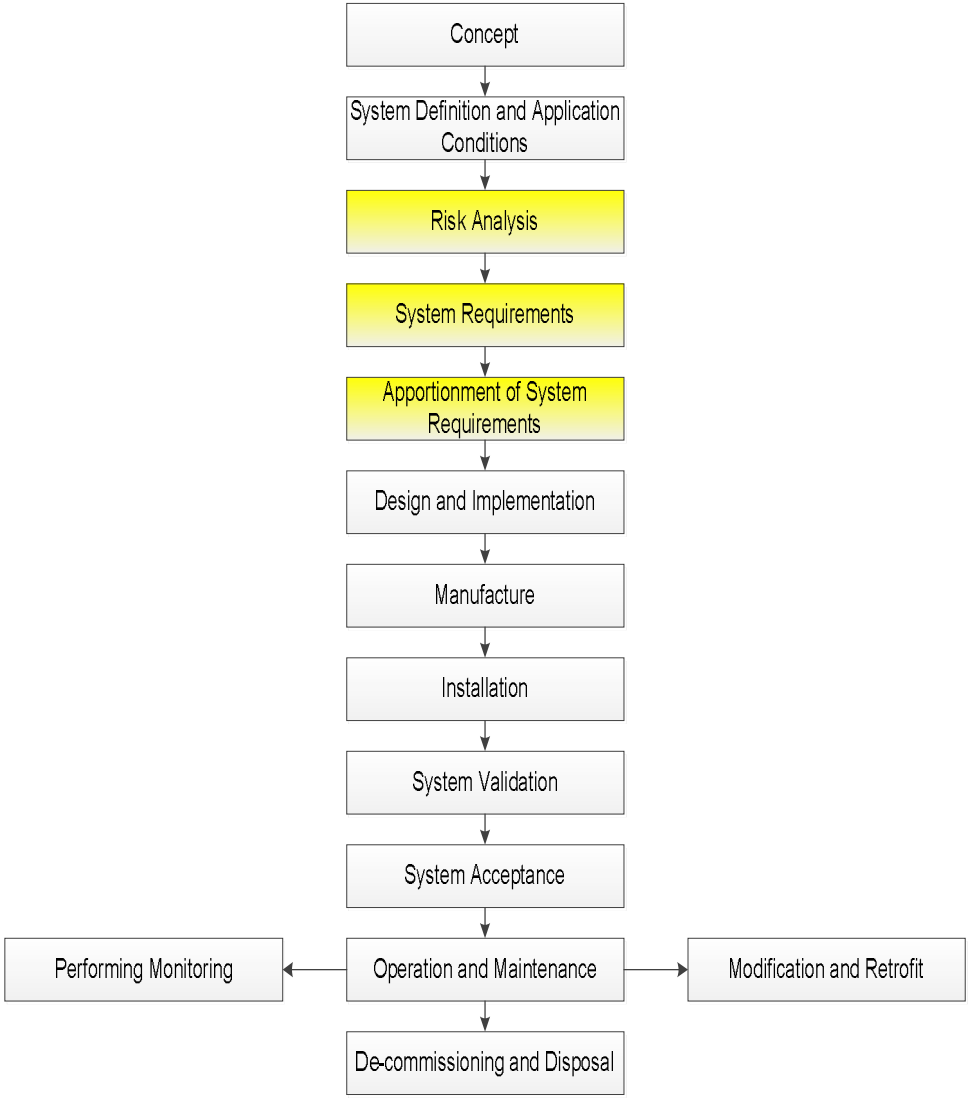
- Undertake project related risk analysis.

4. System requirements

- Specify system and its acceptance criteria.

5. Apportionment of system requirements

- Specify sub-system and component requirements and their acceptance criteria.



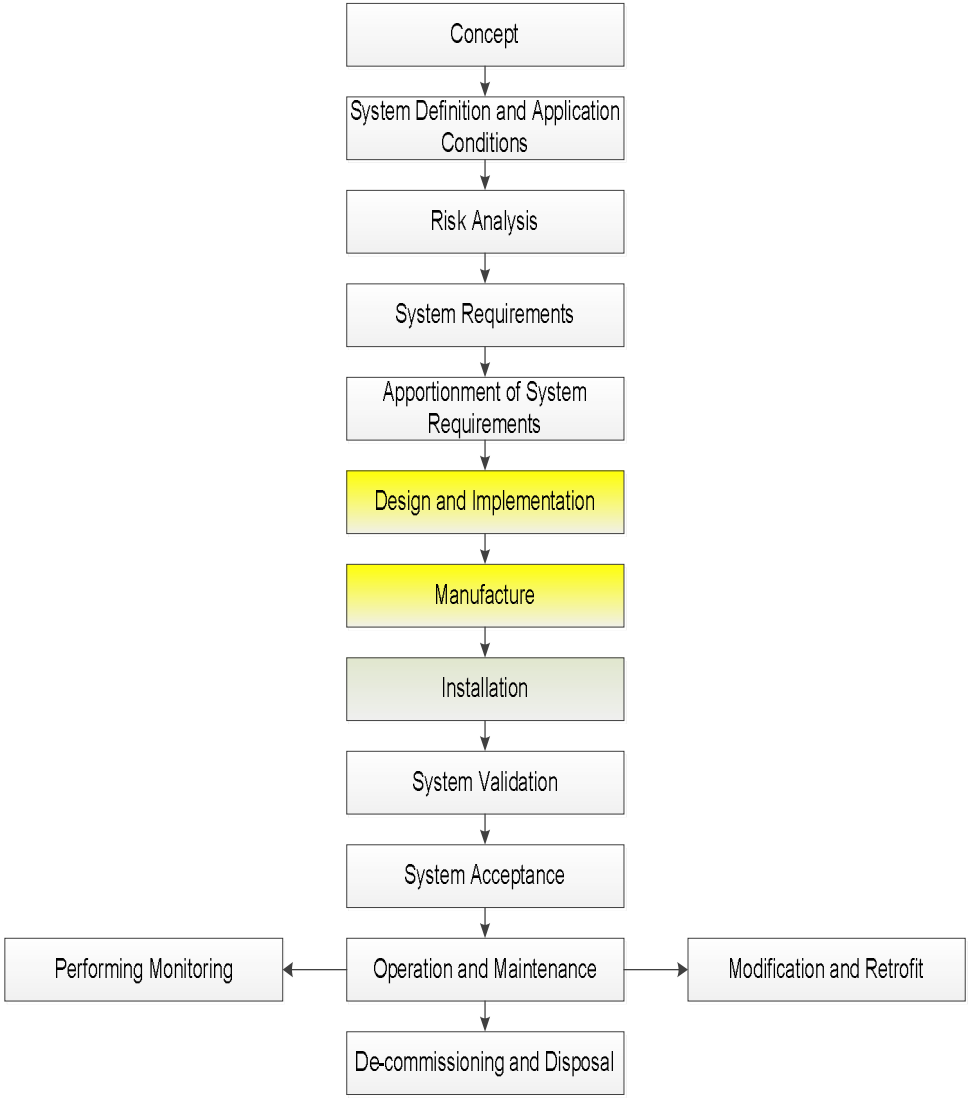
Basic Concept: System Life Cycle (III)

6. Design and implementation

- Perform design analysis and testing conforming to the requirements.
- Perform design conforming to the requirements.
- Perform implementation and validation conforming to the requirements.

7. Manufacturing

- Implement a manufacturing process.
- Manufacture and test sub-assembly of components.



Basic Concept: System Life Cycle (IV)

8. Installation

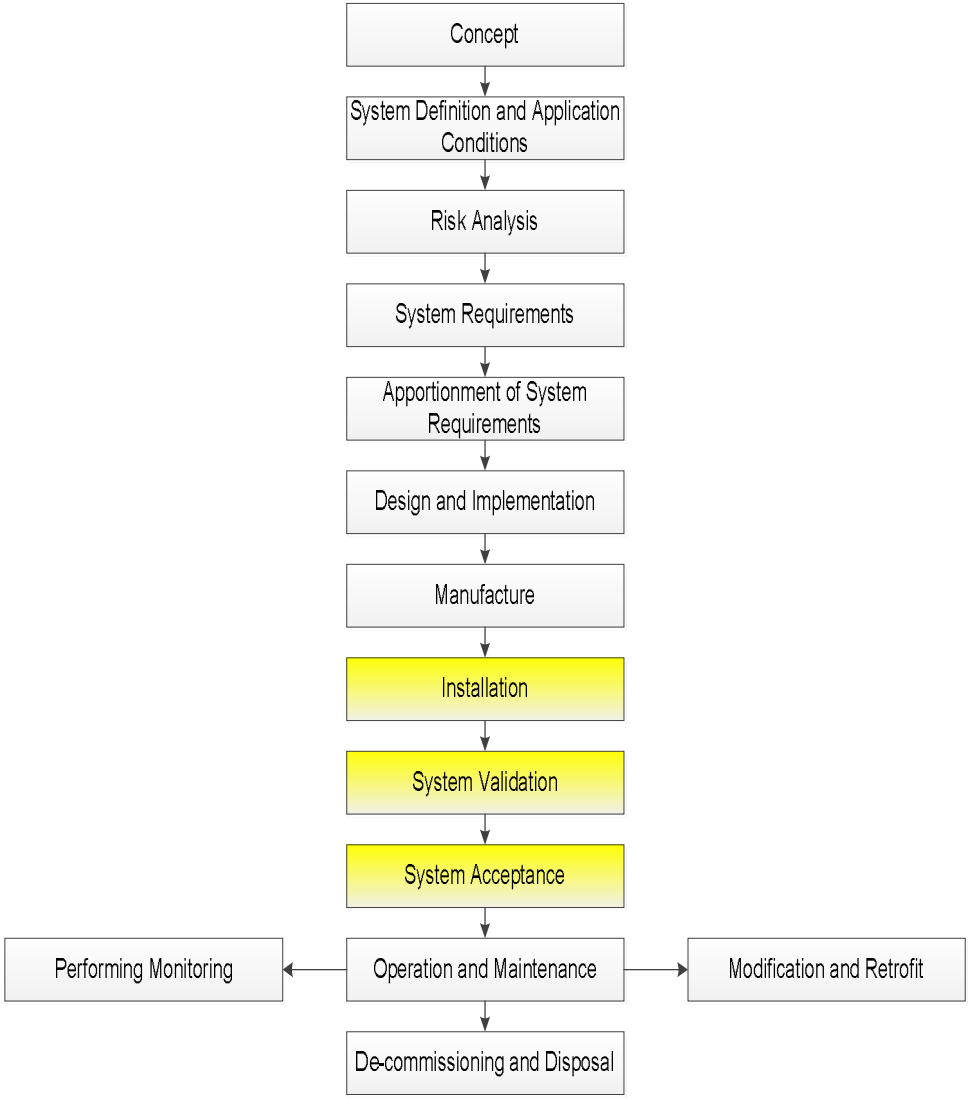
- Assemble and install the system.

9. System validation

- Validate the system and external risk reduction measures.
- Commission the system and external risk reduction measures.
- Prepare, and if appropriate accept, the Application Specific Safety Case for the system.

10. System acceptance

- Accept the system for entry into service.



Basic Concept: System Life Cycle (V)

11. Operation and maintenance

- Operate (within specified limits), maintain the system within RAMS requirements.

12. Performance monitoring

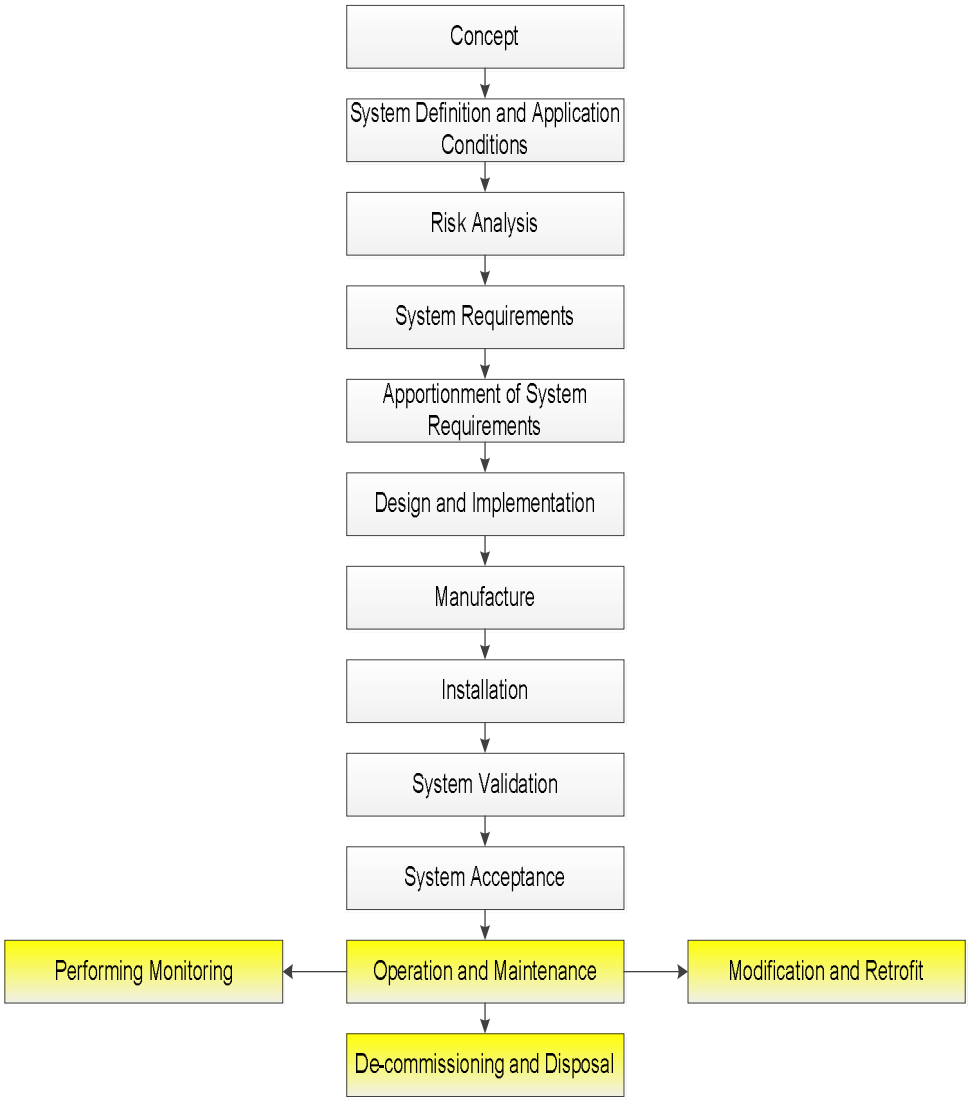
- Maintain confidence in the RAMS performance of the system.

13. Modification and retrofit

- Control system modification and retrofit tasks to maintain system RAMS requirements.

14. Decommissioning and disposal

- Control system decommissioning and disposal.



SIEMENS

Chapter 3

EN50128

EN50128: Introduction

Objective

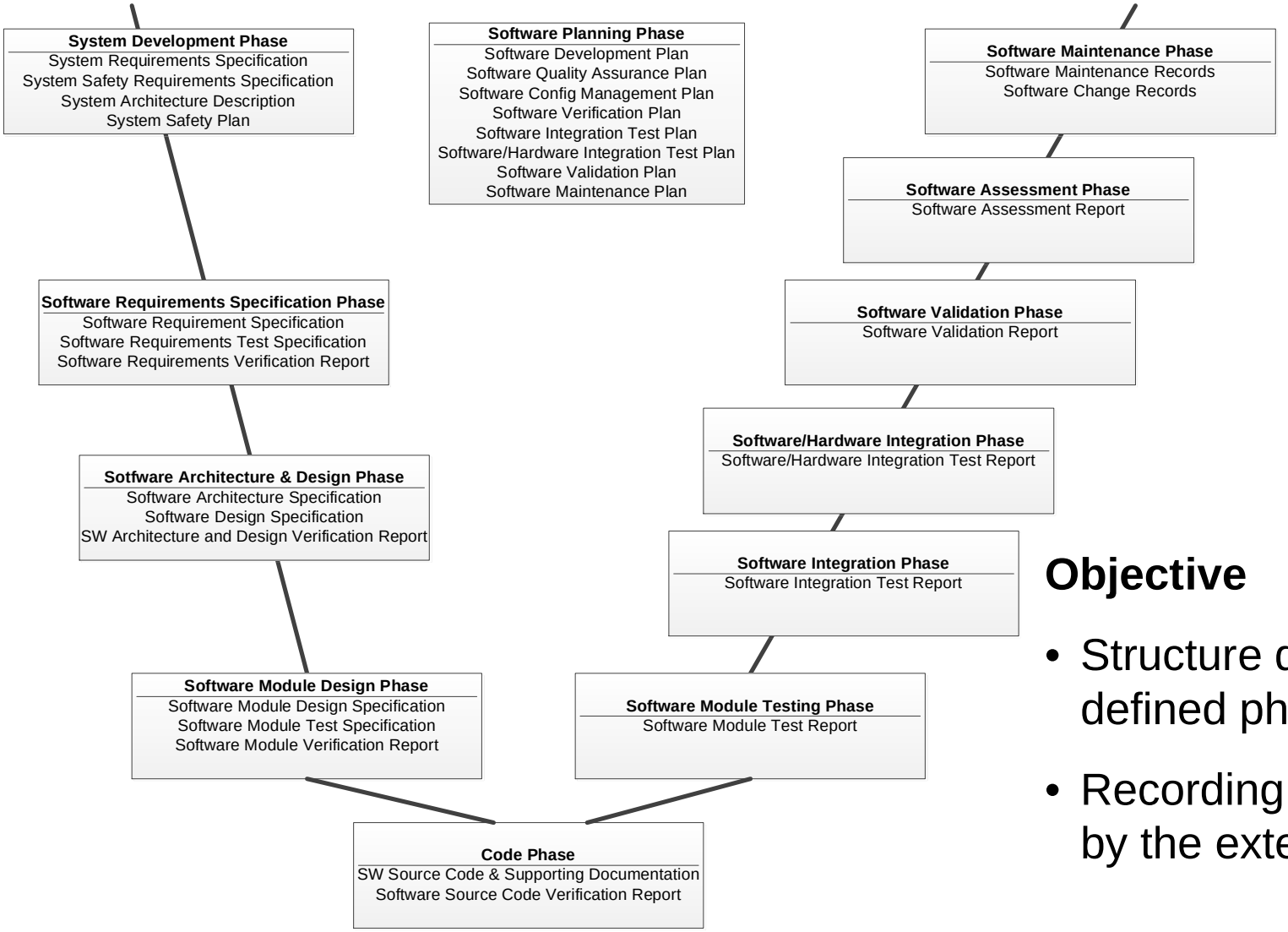
Specifying procedures and technical requirements for the development of programmable electronic systems for use in railway control and protection applications

Scope

Applicable exclusively to software and the interaction between software and the system of which it is part:

- Application programming
- Operating system
- Support tools
- Firmware.

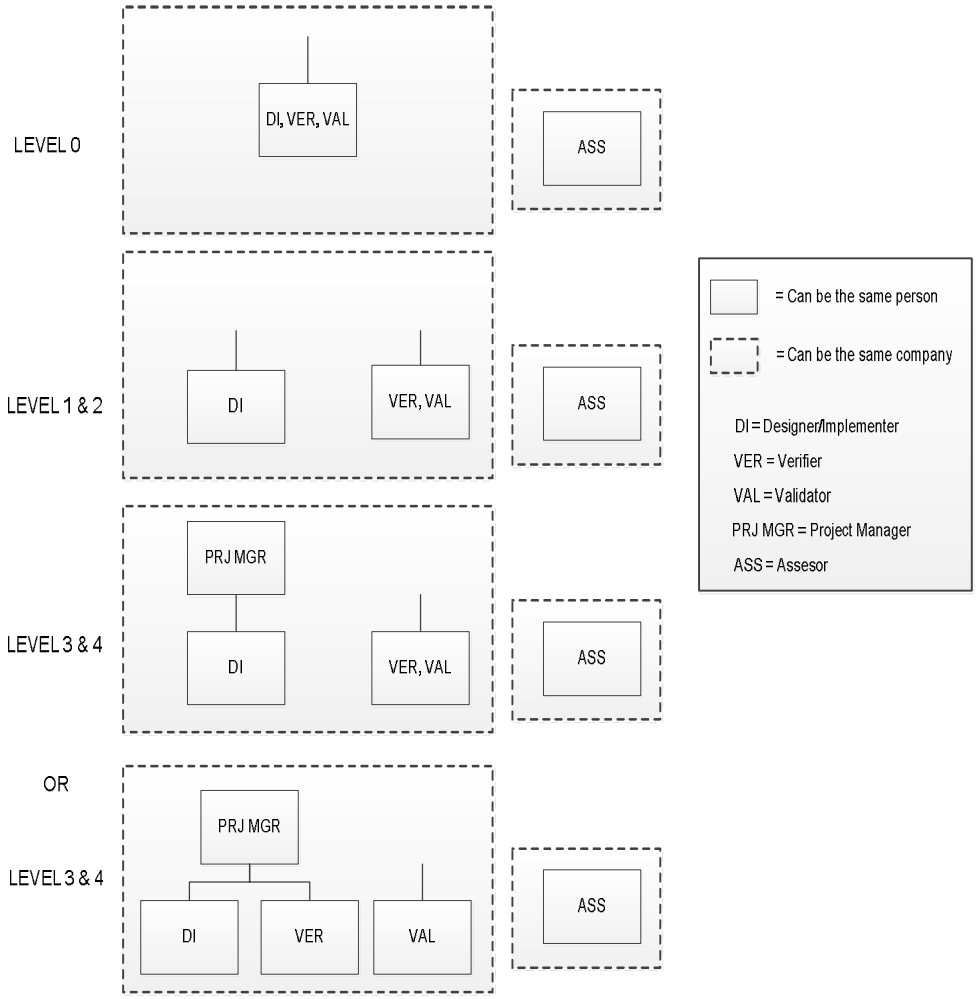
EN50128: Development Lifecycle and Documentation



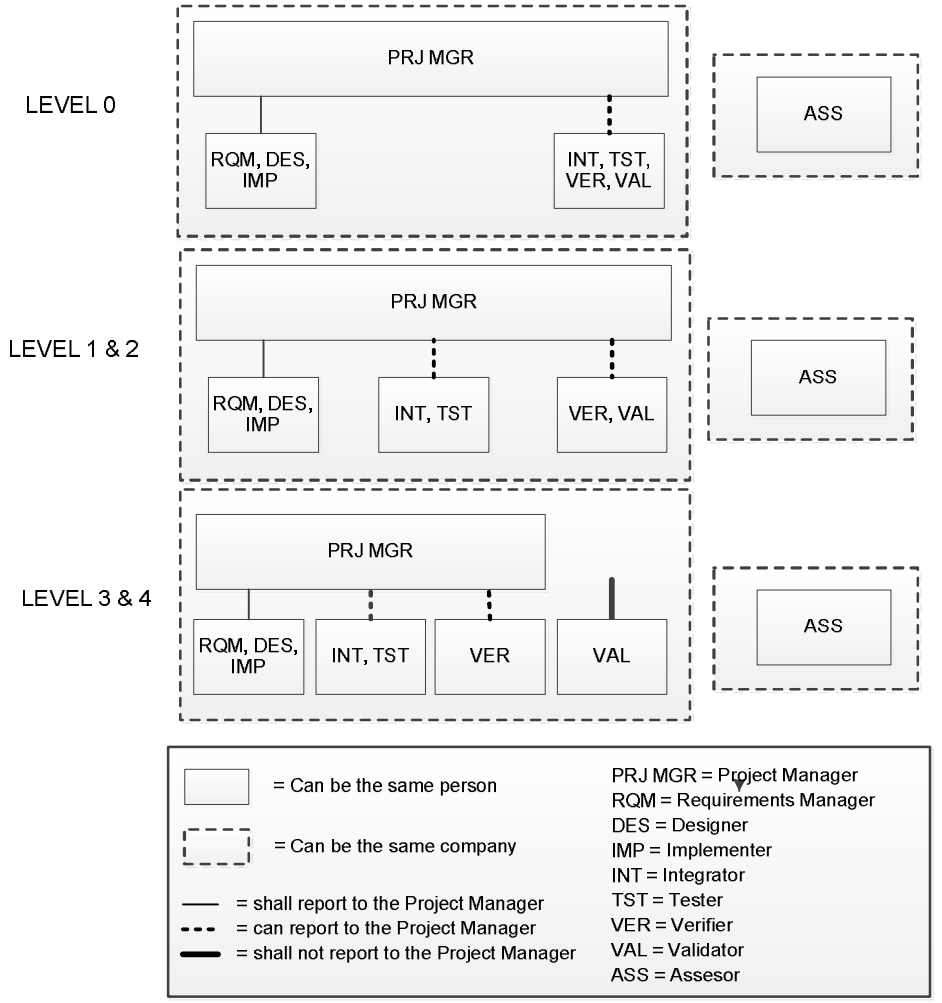
Objective

- Structure development into defined phases and activities.
- Recording evidences required by the external assessor.

EN50128: Organisation



Organisation: 50128:2001



Organisation: 50128:2011

EN50128: Features of ADA (I)

Structured

- Programming paradigm aimed at improving the clarity, quality, and development time of a computer program by making extensive use of subroutines, block structures and for and while loops in contrast to using simple tests and jumps such as the “goto” statement.

Statically Typed

- Process of verifying the type safety of a program based on analysis of the source code. If a program passes a static type-checker, then the program is guaranteed to satisfy some set of type-safety properties for all possible inputs.

Strongly Typed

- Classify values and expression in types allowing interacting when they are the same type.

EN50128: Features of ADA (II)

Imperative

- Focused on describing *how* a program operates. It expresses commands to take action.

Wide-Spectrum

- Designed to be simultaneously a low-level (no abstraction from a computer's instruction set architecture) and a high-level language (strong abstraction from the details of the computer).

Object-oriented

- Based on the concept of "objects", which are data structures that contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods.

EN50128: How does ADA help (50128:2001)? (I)

1. ADA

- ADA is recommended.
- Selection subset of language.
- Definition of Coding Standard.
- Definition of Style Guide.

TECHNIQUE/MEASURE	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1. ADA	R	HR	HR	R	R
2. MODULA-2	R	HR	HR	R	R
3. PASCAL	R	HR	HR	R	R
4. Fortran 77	R	R	R	R	R
5. 'C' o C++ (unrestricted)	R	-	-	NR	NR
6. Subset of C or C++ with coding standards	R	R	R	R	R
7. PL/M	R	R	R	NR	NR
8. BASIC	R	NR	NR	NR	NR
9. Assembler	R	R	R	-	-
10. Ladder Diagrams	R	R	R	R	R
11. Functional Blocks	R	R	R	R	R
12. Statement List	R	R	R	R	R

Requirements:

1. For software safety integrity levels 3 and 4, when a subset of languages 1, 2, 3 and 4 is used the recommendation changes to 'HR'.
2. For certain applications, languages 7 and 8 may be the only ones available. For software safety integrity levels 3 and 4 where a 'HR' option is not available, it is strongly recommended that there is a subset of these languages and a precise set of coding standards in order to raise the recommendation to 'R'.
3. For information on assessing the suitability of a programming language, see entry in the bibliography for 'Suitable Programming Language', B.62.
4. If a specific language is not in the table, it is not automatically excluded. However, it should conform to B.62.

NR=Not Recommended; H=Recommended; HR=High Recommended; M=Mandatory

EN50128: How does ADA help (50128:2001)? (II)

4. Modular Approach

- Object Oriented Language.
- Structured Language.

6. Analyzable Programs

- High Level Computer Programming Language.

7. Strongly Typed Programming Language

- Extremely Strong Typing Language.

8. Structured Programming

- Structured Language.

10. Language subset

- Coding Standard indicates subset.

11. Validated Translator

- Demonstrate with Ada Conformity Assessment Test Suite (ACATS).

TECHNIQUE/MEASURE	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1. Formal Methods, including CCS, CSP, HOL, LOTOS, OBJ, Temporal Logic, VDM, Z and B, for instance.	-	R	R	HR	HR
2. Semi-Formal Methods	R	HR	HR	HR	HR
3. Structured Methodology, including JSD, MASCOT, SADT, SDL, SSADM and Yourdon, for instance.	R	HR	HR	HR	HR
4. Modular Approach	HR	M	M	M	M
5. Design and Coding Standards	HR	HR	HR	M	M
6. Analysable Programs	HR	HR	HR	HR	HR
7. Strongly Typed Programming Language	R	HR	HR	HR	HR
8. Structured Programming	R	HR	HR	HR	HR
9. Programming Language	R	HR	HR	HR	HR
10. Language Subset	-	-	-	HR	HR
11. Validated Translator	R	HR	HR	HR	HR
12. Translator Proven in Use	HR	HR	HR	HR	HR
13. Library of Trusted/Verified Modules and Components	R	R	R	R	R
14. Functional and Black-box Testing	HR	HR	HR	M	M
15. Performance Testing	-	HR	HR	HR	HR
16. Interface Testing	HR	HR	HR	HR	HR
17. Data Recording and Analysis	HR	HR	HR	M	M
18. Furry Logic	-	-	-	-	-
19. Object Oriented Programming	-	R	R	R	R

Requirements:
1. A suitable set of techniques shall be chosen according to the software safety integrity level.
2. For software safety integrity levels 3 and 4, the approved set of techniques shall include one from techniques 1, 2 and 3, together with techniques 11 or 12. The remaining techniques shall still be treated according to their recommendation.

NR=Not Recommended; H=Recommended; HR=High Recommended; M=Mandatory

EN50128: How does ADA help (50128:2001)? (III)

3. Absence Dynamic Objects
4. Absence Dynamic Variables
5. Limited Use of Pointers

- No generic, untyped pointers or implicitly declare any pointer.
- Coding Standard.

TECHNIQUE/MEASURE	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1. Existence of Coding Standards	HR	HR	HR	HR	HR
2. Coding Style Guide	HR	HR	HR	HR	HR
3. Absence of Dynamic Objects	-	R	R	HR	HR
4. Absence of Dynamic Variables	-	R	R	HR	HR
5. Limited Use of Pointers	-	R	R	R	R
6. Limited Use of Recursion	-	R	R	HR	HR
7. Absence of Unconditional Branches	-	HR	HR	HR	HR

Requirement:
It's admitted that techniques 3 and 4 may be present as part of a validated compiler or translator.

NR=Not Recommended; H=Recommended; HR=High Recommended; M=Mandatory

EN50128: How does ADA help (50128:2001)? (IV)

2. Information Hiding/Encapsulation

- Package Specification.
- Extremely Strong Typing.

3. Parameter Number Limit

4. One Entry/One Exit Point in Subroutines and Functions

- Coding Standard.

5. Fully Defined Interfaces

- Package Specification.

TECHNIQUE/MEASURE	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1. Software Module Size Limit	HR	HR	HR	HR	HR
2. Information Hiding/Encapsulation	R	HR	HR	HR	HR
3. Parameter Number Limit	R	R	R	R	R
4. One Entry/One Exit Point in Subroutines and Functions	R	HR	HR	HR	HR
5. Fully Defined Interfaces	HR	HR	HR	M	M
Requirement: A suitable set of techniques shall be chosen according to the software safety integrity level. NR=Not Recommended; H=Recommended; HR=High Recommended; M=Mandatory					

EN50128: How does ADA help (50128:2011)? (I)

1. ADA

- ADA is recommended.
- Selection subset of language.
- Definition of Coding Standard.
- Definition of Style Guide.

TECHNIQUE/MEASURE	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1. ADA	R	HR	HR	HR	HR
2. MODULA-2	R	HR	HR	HR	HR
3. PASCAL	R	HR	HR	HR	HR
4. C or C++	R	R	R	R	R
5. PL/M	R	R	R	NR	NR
6. BASIC	R	NR	NR	NR	NR
7. Assembler	R	R	R	NR	NR
8. C#	R	R	R	R	R
9. Java	R	R	R	R	R
10. Statement List	R	R	R	R	R

Requirements:

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.

NR=Not Recommended; H=Recommended; HR=High Recommended; M=Mandatory

EN50128: How does ADA help (50128:2011)? (II)

4. Modular Approach

- Object Oriented Language.
- Structured Language.

7. Analyzable Programs

- High Level Computer Programming Language.

8. Strongly Typed Programming Language

- Extremely Strong Typing Language.

9. Structured Programming

- Structured Language.

11. Language subset

- Coding Standard indicates subset.

12. Object Oriented Programming

- Object Oriented Language.

13. Procedural Programming

- ADA use the concept of procedure call.

TECHNIQUE/MEASURE	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1. Formal Methods	-	R	R	HR	HR
2. Modelling	R	HR	HR	HR	HR
3. Structured Methodology	R	HR	HR	HR	HR
4. Modular Approach	HR	M	M	M	M
5. Components	HR	HR	HR	HR	HR
6. Design and Coding Standards	HR	HR	HR	M	M
7. Analysable Programs	HR	HR	HR	HR	HR
8. Strongly Typed Programming Language	R	HR	HR	HR	HR
9. Structured Programming	R	HR	HR	HR	HR
10. Programming Language	R	HR	HR	HR	HR
11. Language Subset	-	-	-	HR	HR
12. Object Oriented Programming	R	R	R	R	R
13. Procedural Programming	R	HR	HR	HR	HR
14. Metaprogramming	R	R	R	R	R

Requirements:
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.

NR=Not Recommended; H=Recommended; HR=High Recommended; M=Mandatory

EN50128: How does ADA help (50128:2011)? (III)

3. Absence Dynamic Objects
4. Absence Dynamic Variables
5. Limited Use of Pointers

- No generic, untyped pointers or implicitly declare any pointer.
- Coding Standard.

7. No Unconditional Jumps
9. Entry/Exit Point Strategy
10. Limited Number of subroutine parameters
11. Limited Use of Global Data

- Coding Standard.

TECHNIQUE/MEASURE	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1. Coding Standards	HR	HR	HR	HR	HR
2. Coding Style Guide	HR	HR	HR	HR	HR
3. No Dynamic Objects	-	R	R	HR	HR
4. No Dynamic Variables	-	R	R	HR	HR
5. Limited Use of Pointers	-	R	R	R	R
6. Limited Use of Recursion	-	R	R	HR	HR
7. No Unconditional Jumps	-	HR	HR	HR	HR
8. Limited size and complexity of Functions, Subroutines and Methods	HR	HR	HR	HR	HR
9. Entry/Exit Point strategy for Functions, Subroutines and Methods	R	HR	HR	HR	HR
10. Limited number of subroutine parameters	R	R	R	R	R
11. Limited use of Global Variables	HR	HR	HR	M	M
Requirement: It's admitted that techniques 3 and 4 may be present as part of a validated compiler or translator. NR=Not Recommended; H=Recommended; HR=High Recommended; M=Mandatory					

EN50128: How does ADA help (50128:2011)? (IV)

1. Information Hiding

- Package Specification.

2. Information Encapsulation

- Package Specification.
- Extremely Strong Typing.

3. Parameter Number Limit

- Coding Standard.

4. Fully Defined Interfaces

- Package Specification.

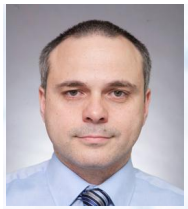
TECHNIQUE/MEASURE	SIL SW 0	SIL SW 1	SIL SW 2	SIL SW 3	SIL SW 4
1. Information Hiding	HR	HR	HR	HR	HR
2. Information Encapsulation	R	HR	HR	HR	HR
3. Parameter Number Limit	R	R	R	R	R
4. Fully Defined Interfaces	HR	HR	HR	M	M

Requirement:
1) 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.

NR=Not Recommended; H=Recommended; HR=High Recommended; M=Mandatory

Thank you for your attention

Contact page



Javier Rodríguez de los Santos
Mass Transit ATP Manager

Siemens Rail Automation S.A.U.
Spain /R&D Department

C/ Ronda de Europa, 5
28760 Tres Cantos Madrid

Phone: +91 678 98 80
E-mail:javier.rodriguez@siemens.com

www.siemens.es