centum digital

Preparing Safety-critical software for its certification in Naval industry

QA&Test Embedded Embedded 2021

About Me

Marta Saña Espinosa is a Telecommunications Engineer with 21 years' experience in SW development projects, always performing tasks directly related to Quality Assurance. She has performed, designed and managed V&V campaigns in different industries (rail, aerospace, naval, defense, telecom, HR...) and different technologies applications, including critical software certification using DO-178 and MIL-STD-882E standards, involving verification and validation of all the products produced during the different phases of the software development lifecycle.

1. Possible aplicable standars for Naval critical software certification

2. Safety evaluation steps for software according to MIL-STD-882E

3. DO-178C standard application

4. Software lifecycle processes according to DO-178C

5. DO-178C and MIL-STD-882E (JSSSEH) documents traceability

6. Final conclusions: Objectives and activities for each level of rigor (LOR)

AGENDA

APPLICABLE STANDARS IN NAVAL INDUSTRY

Preparing Safety-critical software for its certification in Naval industry

Applicable Standards for critical software certification in Naval Industry

- There is not any specific, official and agreed standard for critical software development and certification in Naval industry
- Applicability study for different standards:
    - Generic standards:
        - MIL-STD-882E: DoD Standard Practice for System Safety
        - JSSSEH: Joint Software Systems Safety Engineering Handbook
    - Industry-specific standards:
        - AOP-52: NATO Guidance on Software Safety Design and Assessment of Munition-Related Computing Systems
        - DO-178C: Software Considerations in Airborne Systems and Equipment Certification
- GOAL OF THIS STUDY: Describe a coherent and complete process that sets the necessary guidelines for a possible certification

Applicable Standards for critical software certification in Naval Industry

- CONCLUSION about MIL-STD-882E and JSSSEH standard, and even AOP-52:
  - They are very focused in safety definition process at system level, as well as risk analysis process
  - They define the activity types that should be caried out through the system lifecycle, but the specific activities to be carried out for each level of rigor, have to be agreed between system developer and client.
- This flexibility allows the use of any software lifecycle
- So, which Centum and its client decided was to use MIL-STD-882E standard and it complementary guide, JSSSEH, as the framework standard, and to use DO-178C aeronautical standard as an starting point for specific activities definition.

SAFETY EVALUATION STEPS FOR SOFTWARE ACCORDING TO MIL-STD-882E

Preparing Safety-critical software for its certification in Naval industry

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

Step 01:

Analysis/ Safety Evaluation

- ***Step 1:*** This process starts with the hazards identification and risk evaluation of the system, in order to assign them a severity category, according the the table of next slide.

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

SEVERITY CATEGORIES		
Description	**Severity Category**	**Mishap Result Criteria**
Catastrophic	**1**	Could result in one or more of the following: death, permanent total disability, irreversible significant environmental impact, or monetary loss equal to or exceeding $10M.
Critical	**2**	Could result in one or more of the following: permanent partial disability, injuries or occupational illness that may result in hospitalization of at least three personnel, reversible significant environmental impact, or monetary loss equal to or exceeding $1M but less than $10M.
Marginal	**3**	Could result in one or more of the following: injury or occupational illness resulting in one or more lost work day(s), reversible moderate environmental impact, or monetary loss equal to or exceeding $100K but less than $1M.
Negligible	**4**	Could result in one or more of the following: injury or occupational illness not resulting in a lost work day, minimal environmental impact, or monetary loss less than $100K.

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

Step 02:
Obtain the Level of Rigor of each of the Safety-Significant software function

- ***Step 2:*** After identifying which software functions are safety-significant, the following steps are followed to determine their Level of Rigor (LOR):
  - ***Step 2.1:*** Firstly, a Functional Failure Analysis (FAA) is performed, in order to determine the control category of each safety-significant software function, which is determine mostly by the amount of human intervention in the software control (more details in the next slide)

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

SOFTWARE CONTROL CATEGORIES		
Level	**Name**	**Description**
1	**Autonomous**	The software exercises autonomous control over safety-critical systems, subsystems or hardware components without the possibility of previous safety detection or intervention by a control body to prevent a hazard from occurring. Software failure at the moment at which it is required leads directly to a hazard occurring.
2	**Semi-autonomous**	The software exercises control over safety-critical systems, subsystems or hardware components, allowing time for the pre-set detection and intervention of independent safety systems in order to mitigate or control the hazard. The software shows safety information which requires immediate action of the operator in order to mitigate or control a hazard. Software failures (including delays in execution) would make it impossible to prevent this hazard occurring.
3	**Redundant failure tolerant**	The software issues orders on safety-critical systems, subsystems or hardware components which require a control entity to complete the control function. The detection and intervention (reaction) systems include redundant and independent fault-tolerant mechanisms for each hazard condition defined. The software generates safety-critical information which is used to take critical decisions. It includes redundant and independent fault-tolerant detection and monitoring mechanisms for each hazard condition defined.
4	**Influential**	The software generates safety information used by the operator to take decisions, but does not require the action of an operator in order to avoid an accident.
5	**Without Safety Impact**	The software does not control safety-critical systems, subsystems or hardware components, and does not provide safety-critical information.

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

Step 02:
Obtain the Level of Rigor of each of the Safety-Significant software function

- ***Step 2.2:*** Secondly, taking into account both the control category and severity category for each of the safety-significant software functions, the Software Criticality Index (SwCI) or Level of Rigor (LOR) is determined using the table of the next slide.

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

SOFTWARE SAFETY CRITICALITY MATRIX				
	SEVERITY CATEGORY			
SOFTWARE CONTROL CATEGORY	Catastrophic (1)	Critical (2)	Marginal (3)	Negligible (4)
Autonomous (1)	SwCI 1	SwCI 1	SwCI 3	SwCI 4
Semi-Autonomous (2)	SwCI 1	SwCI 2	SwCI 3	SwCI 4
Redundant Failure-Tolerant (3)	SwCI 2	SwCI 3	SwCI 4	SwCI 4
Influential (4)	SwCI 3	SwCI 4	SwCI 4	SwCI 4
Without Safety Impact (5)	SwCI 5	SwCI 5	SwCI 5	SwCI 5

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

Step 03:
Provide the development team with the tasks to be performed in accordance with the Level of Risk (LOR)

- ***Paso 3:*** Once the Level of Rigor has been established for each Safety-Significant function, it is necessary to provide the software development team with the tasks required for the coding, verification and validation of the software, for each of the aforementioned levels. The table in the next slide sets out the high-level activities that standard MIL-STD-882E establishes for each Level of Rigor (LOR).

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

SwCI	Activities by Level of Rigor
SwCI 1	The program must perform an analysis of requirements, architecture, design and source code, and also undertake specific in-depth safety tests.
SwCI 2	The program must perform an analysis of requirements, architecture and design, and also undertake specific in-depth safety tests.
SwCI 3	The program must perform an analysis of requirements and architecture, and also undertake specific in-depth safety tests.
SwCI 4	The program must undertake specific in-depth safety tests.
SwCI 5	Once the evaluation performed by Safety Engineering results in No Safety, then no analysis or specific safety verification is required.

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

Step 04:
Implementation of the software tasks, in accordance with the Level of Rigor (LOR) and the review of the execution thereof to calculate the residual risk

- ***Step 4:*** The forth step of this process has as its objective the implementation of the software development, verification and validation tasks, in accordance with the Level of Rigor (LOR) and the review of the execution thereof to calculate the residual risk:
  - If the tasks are completed correctly, the results will be used to re-evaluate the risks of the system to which the hazard identified belongs.
  - If the LOR tests are not completed, a risk level will be assigned to the hazard, based on the table in the next slide.

Steps of the Safety Lifecycle (MIL-STD-882E) from the software point of view

Relationship between SwCI, Level of Risk, LOR and Risk		
SwCI	**Level of Risk**	**LOR Software Tasks and Assessment/Acceptance of the Risk**
SwCI 1	**High**	If the LOR tasks for SwCI 1 are not specified or are incomplete, the contributions to the system risk will be documented as HIGH, and will be presented to the Program Manager (PM) for him to reach a decision. The PM must document the decision whether to spend the resources necessary to implement the LOR tasks for SwCI 1, or prepare a formal risk assessment for acceptance of a HIGH risk.
SwCI 2	**Serious**	If the LOR tasks for SwCI 2 are not specified or are incomplete, the contributions to the system risk will be documented as SERIOUS, and will be presented to the PM for him to reach a decision. The PM must document the decision whether to spend the resources necessary to implement the LOR tasks for SwCI 2, or prepare a formal risk assessment for acceptance of a SERIOUS risk.
SwCI 3	**Medium**	If the LOR tasks for SwCI 3 are not specified or are incomplete, the contributions to the system risk will be documented as MEDIUM, and will be presented to the PM for him to reach a decision. The PM must document the decision whether to spend the resources necessary to implement the LOR tasks for SwCI 3, or prepare a formal risk assessment for acceptance of a MEDIUM risk.
SwCI 4	**Low**	If the LOR tasks for SwCI 4 are not specified or are incomplete, the contributions to the system risk will be documented as LOW, and will be presented to the PM for him to reach a decision. The PM must document the decision whether to spend the resources necessary to implement the LOR tasks for SwCI 4, or prepare a formal risk assessment for acceptance of a LOW risk.
SwCI 5	**No Safety**	No specific tests or analysis is required for safety

DO-178C STANDARD APPLICATION

Preparing Safety-critical software for its certification in Naval industry

DO-178C Standard **Application**

- MIL-STD-882E standard define which type of activities are carried out throughout the software development lifecycle, but leave it open to an agreement between system developer and client the specific activities that has to be carried out for each LOR.
- As DO-178C standard, which is applied in critical software development in aeronautical industry, defines in a very specific way, which activities need to be performed for each of the levels, we can support us in DO-178 to define the software development activities in our case.
- Therefore, once LOR is defined for each of the software components, we will use DO-178C as a starting point to determine which activities will be performed.

DO-178C Standard **Application**

- The failure conditions in DO-178C are categorised into 5 safety levels (Design Assurance Level, DAL) from A to E, according to the impact of a software malfunction:

Category	Description	DAL
Catastrophic	Failure conditions that prevent safe flight and landing.	A
Hazardous	Failure conditions that reduce the capacity of the crew to address adverse operating conditions, cause a considerable reduction in safety margins, cause serious or fatal injuries to a small number of passengers.	B
Major	Failure conditions that increase the workload of the crew, cause a significant reduction in safety margins, discomfort or possible injuries to passengers.	C
Minor	Slight reduction in the safety margins or functional capacities, slight effects on the crew, inconvenience to passengers.	D
Without safety consequences	No effect on the operation of the aircraft or the crew workload	E

DO-178C Standard Application

- Once these levels are defined, in order to be able to use DO-178C activities definition, an approximate traceability between LOR levels defined by MIL-STD-882E and DAL levels defined by DO-178C has to be established:

DO-178C		MIL-STD-882E	
DAL	**Category**	**LOR SwCI**	**Level of Risk**
A	Catastrophic	SwCI 1	High
B	Hazardous	SwCI 2	Serious
C	Major	SwCI 3	Medium
D	Minor	SwCI 4	Low
E	Without safety consequences	SwCI 5	No Safety

- It has to be taken into account that this traceability is approximate, so the level definition in each of the standard is not exactly equivalent, so these diferrences have to be taken into acoount when defining the activities to be performed to each LOR.

SOFTWARE LYFECYCLE PROCESSES ACCORDING TO DO-178C

Preparing Safety-critical software for its certification in Naval industry

Software life-cycle processes According to DO-178C

- DO-178C standard divides the software life-cycle into six processes:
  - Planning.
  - Development.
  - Verification (including both verification per se and validation).
  - Configuration Management.
  - Quality Assurance.
  - Certification Liaison.

Software life-cycle processes According to DO-178C

Planning

- Planning process entails the development of specific plans and standards for the project, and their fulfilment along the project lifecycle. DO-178C establishes for this process a total of five plans and three standards:
  - Plans:
    - PSAC: Plan for Software Aspect of Certification.
    - SDP: Software Development Plan.
    - SVP: Software Verification Plan.
    - SCMP: Software Configuration Management Plan.
    - SQAP: Software Quality Assurance Plan.
  - Standards:
    - SRS: Software Requirements Standards.
    - SDS: Software Design Standards.
    - SCS: Software Code Standards.
- Although almost all organisations follow this suggestion of 5 plans and 3 standards, as long as all the required content is available, it is possible to group plans and/or standards in the way it fits more with the organisation needs.

Software life-cycle processes According to DO-178C

Software Development

- This process is subdivided into four sub-processes:
  - ***Software Requirements Process:*** The objective of this sub-process is to develop high-level software requirements, which will be evaluated in accordance with the Safety process.
  - ***Software Design Process:*** The purpose of the sub-process is to divide the high-level requirements into design level requirements which can be traced directly with the source code.
  - ***Software Coding Process:*** This begins once the low-level requirements are transformed into code. This code must be traceable with the requirements from the design process.
  - ***Integration Process:*** Lastly, the integration sub-process integrates this source code into its environment in real time, along with its hardware components.

Software life-cycle processes According to DO-178C

Verification & Validation

- The objectives of verification and validation are satisfied by means of a combination of reviews, analyses, test cases and procedures development, and the subsequent execution of these test procedures and cases.
- The reviews and analyses provide an evaluation of the precision, completeness and verifiability of the software requirements, software architecture and source code.
- The development of test cases and procedures likewise provides an evaluation of internal consistency and completeness of requirements.
- The execution of the test cases and procedures provides a demonstration of the fulfillment of the requirements.

Software life-cycle processes According to DO-178C

Configuration Management

- The objective of this process is to establish effective and safe configuration control for all elements produced over the life-cycle of the software.
- There are four main groups of activities that are completed within the configuration management process:
  - Configuration Management: all the elements within configuration control must have a means of identification and referencing on an unequivocal basis.
  - Change control: Tracked items must have an established change control method. Changes must be traceable and reversible.
  - Baseline Establishment: a baseline or point of origin must be defined for each item in change configuration.
- Software Archiving: this provides an additional degree of safety beyond track changes.
- Standard DO-178C defines two levels of rigor or configuration control categories, CC1 and CC2, for the handling of items in configuration management. The activities in configuration control category CC2 are a subset of those included within category CC1, as the latter is more restrictive.

Software life-cycle processes According to DO-178C

Configuration Management

SCM Process Activity	CC1	CC2
Configuration Identification	*	*
Baselines	*	
Traceability	*	*
Problem Reporting	*	
Change Control – integrity and identification	*	*
Change control – tracking	*	
Change Review	*	
Configuration Status Accounting	*	
Retrieval	*	*
Protection against Unauthorised Changes	*	*
Media Selection, Refreshing, Duplication	*	
Release	*	
Data Retention	*	*

Software life-cycle processes According to DO-178C

Quality Assurance

- The purpose of this process is to ensure that the software life-cycle process is producing, has produced and will produce a software element that complies with the certification criteria.
- This is performed by means of a review of the transitions between the processes in order to see whether the inputs and outputs generated fits with the expectations.
- Meanwhile, since the changes in the plans are inevitable, they must be registered and reviewed in order to ensure that process integrity has not been tampered.

DO-178C AND MIL-STD-882E (JSSSEH) DOCUMENTS TRACEABILITY

Preparing Safety-critical software for its certification in Naval industry

Deliverable Documents in DO-178C and MIL-STD-882E (JSSSEH)

- MIL-STD-882E standard, and its JSSSEH guide, explicitly specify which plans have to be delivered, but in terms of the low-level documentation, again it leaves it open to an agreement between system developer and client, that needs to be included and agreed in the contract, and must be expressed in a CDRL (Contract Deliverable Requirement List)..
- This flexibility, again, allows us to assume the low-level documental structure of DO-178C.
- So, in terms of plans, apart from the standards and plans stated in DO-178C, the following plans have to be added, as there are explicitly demanded in MIL-STD-882E and its JSSSEH guide:
    - Software Installation Plan (SIP)
    - Software Transition Plan (STP)

Preparing Safety-critical software for its certification in Naval industry

Deliverable Documents in DO-178C and MIL-STD-882E (JSSSEH)

- The following table compiles the list of deliverables of standards DO-178C and MIL-STD-882E (JSSSEH), in addition to the traceability between the two:

Life-cycle data element (DO-178C)	Life-cycle data element (JSSSEH)
Plan for Software Aspects of Certification (PSAC)	Reliability Engineering Plan
Software Development Plan (SDP)	Software Development Plan
Software Verification Plan (SVP)	Test and Evaluation Master Plan Software Test Plan
Software Configuration Management Plan (SCMP)	Software Configuration Management Plan
Software Quality Assurance Plan (SQAP)	Quality Assurance Plan
Software Requirements Standards (SRS)	-
Software Design Standards (SDS)	-
-	Software Installation Plan (SIP)
-	Software Transition Plan (STP)
Software Code Standards (SCS)	-
Software Requirement Data (SRD)	-
Software Design Description (SDD)	-
Source Code (SC)	-
Executable Object Code (OC)	-
Software Verification and Validation Cases and Procedures (SVCP)	-
Software Verification and Validation Result (SVR)	-
Software Life Cycle Environment Configuration Index (SECI)	-
Software Configuration Index (SCI)	-
Problem Reports (PR)	-
Software Configuration Management Records (SCMR)	-
Software Quality Assurance Records (SQAR)	-
Software Accomplishment Summary (SAS)	-
Trace Data (TrD)	-
Parameter Data Item File (PDIF)	-
-	Contract Deliverable Requirements List (CDRL)

FINAL CONCLUSIONS: OBJECTIVES AND ACTIVITIES FOR EACH LEVEL OF RIGOR (LOR)

Final Conclusions: Objectives and Activities For Each Level of Rigor (LOR)

LOR 5

- No Additional activity is required apart from the engineering process that usually follows the system developer.

Final Conclusions: Objectives and Activities For Each Level of Rigor (LOR)

LOR 4

- The objectives to fulfil are the following:

Process	Objectives Summary	Documents / Deliverables
Planning	- All plans apply to this level	All plans
Development	- High-level requirements developed - Software architecture developed - Executable Object Code developed	Software Requirement Data (SRD) Software Design Description (SDD) Executable Object Code (OC)
V&V	- Review and analysis of high-level safety requirements - Normal and robustness tests of high-level safety requirements - Requirements coverage of high-level safety requirements - Tests to ensure compatibility with target equipment	Software verification and validation cases and procedures (SVCP) Software verification and Validation results (SVR)
Configuration Management	- Configuration Management	Software life cycle environment configuration index (SECI) Problem reports (PR) Software configuration management records (SCMR)
Quality Assurance	- Quality Assurance (fulfilment of plans and compliance review of them)	Software quality assurance records (SQAR)
Certification Liaison	- Accomplishment summary and configuration index	Software configuration index (SCI) Software accomplishment summary (SAS)

Final Conclusions: Objectives and Activities For Each Level of Rigor (LOR)

LOR 3

- LOR 4 objectives, plus:

Process	Objectives Summary	Documents / Deliverables
Planning	- Development standards (3 standards)	Software requirements standards (SRS) Software design standards (SDS) Software code standards (SCS)
Development	- Low-level requirements developed - Trace data developed - Source Code developed	Software design description (SDD) Trace data (TR) Source code (SC)
V&V	- Review and analysis of high level requirements (and their conformity with standards) - Low-level requirements review (precision, conformity with standards, high-level requirements coverage) - Review and Analysis of the software architecture (consistency, compatibility with high level requirements) - Review and Analysis of source code (statement coverage) - Low-level requirements normal and robustness tests - Test procedures review - Test Results Review	Software verification and validation cases and procedures (SVCP) Software verification and Validation results (SVR)
Quality Assurance	- Additional Quality Assurance (plans and standards review, standards fulfilment, y transition criteria)	Software quality assurance records (SQAR)

Final Conclusions: Objectives and Activities For Each Level of Rigor (LOR)

LOR 2

- LOR 3 objectives, plus:

Process	Objectives Summary	Documents / Deliverables
V&V	- Additional Review and analysis of high level requirements (compatibility with target equipment) - Additional Low-level requirements review (compatibility with target equipment, verifiability) - Additional Review and Analysis of the software architecture (compatibility with target equipment, verifiability) - Additional Review and Analysis of source code (verifiability, decision coverage)	Software verification and validation cases and procedures (SVCP) Software verification and Validation results (SVR)

Final Conclusions: Objectives and Activities For Each Level of Rigor (LOR)

LOR 1

- LOR 2 objectives, plus:

Process	Objectives Summary	Documents / Deliverables
V&V	- Additional Review and Analysis of source code (modified condition/decision coverage) - Verification of traceability between source code and object code	Software verification and validation cases and procedures (SVCP) Software verification and Validation results (SVR) Trace Data (TR)

THANKS!

+34 911 84 03 96
info@centum.com
www.centum.com