IEC 61508 – Why Tool Qualification

Hassan EL-Sayed , PhD CEng FInstMC
November 2024

UL Solutions | Empowering Trust in a Complex World

Our Members

Why tool qualification

- Faults and failures; what is the difference?
- Fault is a mistake or error, while failure is the inability to achieve a certain goal. So failure can be caused by a small fault in a plan, such as specification, design, testing, programming, coding, etc.
- **E/E/PE** elements are liable to fail due to degradation, corrosion, thermal stressing, environmental impact, software bug, defective component, lack of maintenance, wrong components values, etc.
- Random hardware failures are due to physical causes only apply to hardware components within a sub-system, element or complete system.
- Random HW failures can be classified as safe failures, dangerous or no effects failures. This leads to put the system in a fail safe mode or provide alarming or diagnostic.
- Systematic faults are produced by many sources during product lifecycle development or commissioning, can be created in any stage of the product lifecycle.

Why tool qualification

- Systematic faults are errors that occur due to flaws in the product development procedures, it can be in any stage of the product lifecycle, e.g. specification, design, programming, supporting tool, manufacturing, operation, maintenance, and decommissioning, examples such as:

✓ Weakness in the design, SRS not defined clearly, poor validation & verification process

✓ Lack of safety management procedures, lack of competency, no training provided

✓ Poor handling modification and impact analysis process, poor documentation and records

✓ Lack of confidence in the software development tools

✓ Lack of testing procedures in the application program – poor code verifiation

✓ Poor safety validation plan

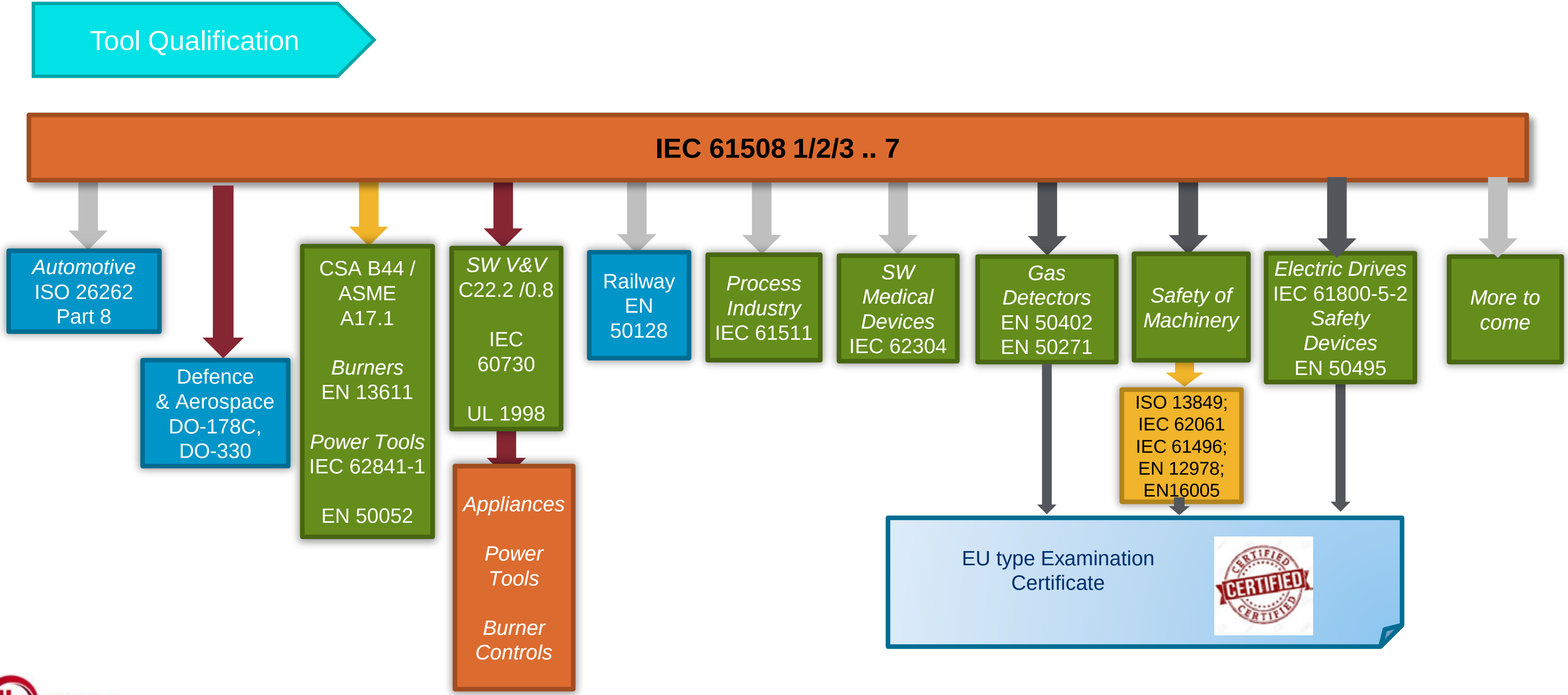
✓ Poor internal and external auditing to control quality and maintain documentation

✓ Human factor errors, such as incorrect data entry, improper use of equipment, ambiguous operational manual.

Why tool qualification

Cause	Systematic failure	Random HW failure
Aging / Wear out	x	x
Fatigue	x	x
Corrosion	x	
Design error	x	
EMC impact	x	
Environment impact	x	
Installation fault	x	
Internal HW fault		x
SW / Programming fault, coms	x	
Software supporting tools	x	
Incorrect test procedures	x	
Leakage current		x
Signal, Stuck - at		x
Signal drift, open circuit , short circuit		x
Unauthorized intervention	x	

Software and SIL designation



Software and SIL designation

- In random hardware analysis, FMEDA / FMEA methods used, so that Safety Integrity Level SIL of the safety function can be deterministically defined.
- Software cannot be measured quantitively, it is a qualitative assessment.
- A safety function targeting SIL 1, FMEDA can be used, but what evidence is there that the developed embedded software is suitable for SIL 1, for a type B product.
- Suppose, the developed embedded software shows no violation in the code, due to the use of a static analysis tool. But what about the tool that generates the target code, binary code?
- What evidence and confidence level shall be considered so that the supporting software development tools are developed and verified to support target SILs 1, 2 or 3.

Software and SIL designation / PIU Software

- Software can be modelled using, a cumulative distribution function: $F(t) = 1 - e^{-\lambda t}$

  Where λ is failure / time, then this can be expressed as by 1/ t_a is the time for SW to fail.

  $P(t_a \leq t) = 1 - e^{-t/t_a}$ where, t is the test run time

- Suppose, t_a is targeting SIL 1, of which must demonstrate no failures, then $t_a = 10^6$ hrs, for a given 95% confidence so that software will not fail in that selected time, then: say $t_a = 10^6$

  $$0.95 = 1 - e^{-t/10^6}$$

- $t = 3 \times 10^6$ hours. As t_a increases toward SIL 2, then t increases by an order of magnitude.
- See Part 7, annex D, table D1.
- For 99% confidence, we need at least 4.6M hrs to justify PIU argument for SIL 1.

Avg. freq. of dgrs failure, hr^{-1} (PFH) for high/cont demand	Hours of operation in total. 1-α = 0.95	SIL
10^{-9} to 10^{-8}	3×10^9	4
10^{-8} to 10^{-7}	3×10^8	3
10^{-7} to 10^{-6}	3×10^7	2
10^{-6} to 10^{-5}	3×10^6	1

Software Development Tools - Evidence

- **Software tools – proof of suitability for the target SIL is required.**
- **Tools Validation ,** IEC 61508, ISO 26262,

  Requirements based, testing of tool operational requirements which specify tool behaviour.
- **Increased confidence from use,** IEC 61508, ISO 26262,

  The software tool has an extensive history of successful previous use.
- **Tool error detection means,** EN 50128

  Built-in functionality to prevent or to detect tool errors like redundant code
- **Tool development according to a safety standard**

  Software tool is developed according to a safety standard to avoid systematic failures
- **Evaluation of tool development process, lifecycle review**

Off-line Support Tools, IEC 61508 Part 3 (ed.2), Cl. 7.4.4

- Selection of the off-line support tools are selected as a coherent part of the SW development activities.
- **Three classes of support tools:**
- **T1** generates no outputs which can directly or indirectly contribute to the executable code (including data) of the safety related system; Ex.
  - ✓ a text editor or a requirements or design support tool with no automatic code generation capabilities; configuration control tools.
- **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. Ex. includes:
  - ✓ a test harness generator; a test coverage measurement tool; a static analysis tool. T2 – Tools may fail to detect errors in the source code
- **T3** generates outputs which can directly or indirectly contribute to the executable code of the safety related system, ex. includes:
  - ✓ an optimising compiler. Tool may introduce errors in the generated code.

Tool Verification, Clause 7.4.4

- **7.4.4.1 to 7.4.4.6:** The selection of **the off-line for T1, T2 and T3** support tools **shall be justified,** for **each tool in class T3**, evidence shall be available that the **tool conforms to its specification or documentation.**
- **7.4.4.7 to 7.4.4.10:** Tool validation, records, validation passed / failure**;** shall be documented covering the following results, effective **measures to control failures** of the executable safety related system that result from faults that are attributable to the tool.
- **7.4.4.11 to 7.4.4.14:** the software uses only defined language features, contain features that facilitate the detection of design or programming mistakes, **the suitability of the automatic translator development shall be assessed.**
- **7.4.4.15 to 7.4.4.19:** Where off-line tools of classes **T2 and T3** generate items in the configuration baseline**, configuration management** shall ensure that information on the tools is recorded in the configuration baseline. Each new version of off-line support tool shall be qualified.

IEC 61508 Part 3, Tool Confidence level – Ed.2

Tool Classes	Description	verdict	example
T1			
T2			
T3			

Do tool errors in safety analysis tools have a direct or indirect impact on safety?
Hypothesis : YES
Detailed analysis is required!.

Example: IEC 61508: Tool class is selected according to tool description and listed.
Requires management tool, Modeling tool without code generation.
Selected tool class is T1

- No tool qualification is required
- No tool risk analysis is required.
- No mitigation actions in place for risks caused by tool errors even when they would have an direct or indirect impact on safety!

UL Solutions

IEC 61508 Part 3, Effect of Tool Errors Impact

- T1 tool is selected as part of the requirement of software configuration management tool.
- Deficiency of IEC 61508: tool qualification requirements for **T2 & T3 proportional to SIL are not provided**, and so supporting measures **are not described.**
- IEC 61508 does not ask for a **tool risk assessment analysis**, Therefore, no mitigation or action is in place for risks caused by tool errors even if they have direct or indirect impact.

Role of tool errors in a safety analysis tools, we have to consider:

- **Process risks**
  Risk analysis of the safety development lifecycle
- **Tool & integration risks**
  Risk analysis of automated or supporting safety analysis process
- **Tool operation scenarios**
  Look how the tool is used in the safety lifecycle. Does it introduce errors or failing to detect errors.

Annex A3, Support Tools and Programming Language

	Technique / Measure	SIL2	Method used in the Product X Software
1	Suitable programming language	HR	C is very suited to embedded real time applications like Product X
2	Strongly typed programming language	HR	C
3	Language subset	NR	NA (SIL 1& 2) . HR for SIL3 & 4
4b	Tools: increased confidence from use	HR	The IAR compiler and associated tools are highly regarded in the industry and have been used successfully by the developers on many projects.
5b	Translator: increased confidence from use	HR	As above

Problems with Part 3

- Little guidance on how to select the most appropriate set of tools and techniques for the development

- In some cases the tools and techniques appear to overlap, but both may be recommended and therefore it is not clear which tools should be used

- What is the minimum set of tools and techniques that would be considered sufficient to demonstrate compliance with the standard?

- *How can we produce a software safety case – a structured argument, supported by a body of evidence that provides a compelling, comprehensible and valid case that a system is safe in a given operating environment (Def Stan 00-56)*
- The guidance is just a list that does not show which set of techniques will be enough to justify safety.

- Improved tools that are now available make it reasonable to treat some recommended techniques as ‘highly recommended’

ISO 26262 Part 8

- Tool confidence level (TCL).
- Tool confidence Level TCL1 to TCL 3 is defined by tool impact (TI) and tool error detection TD level.
- TI – Tool Impact
- TD – Confidence level if tool error can be detected or prevented

		Tool error detection		
		TD1	TD2	TD3
Tool Impact	TI1	TCL1	TCL1	TCL1
	TI2	TCL1	*TCL2*	*TCL3*

Tool qualification methods

- Increased confidence from use
- Tool validation
- Evaluation of tool development
- Tool development according to a safety standard (tools vendors)

Qualification method	ASIL			
	A	B	C	D
Increased confidence from use	++	++	+	+
Evaluation of the development process	++	++	+	+
Validation of the software tool	+	+	++	++
Development in compliance with a safety standard	+	+	++	++

+ Recommended
++ Highly recommended

IEC 61508 Part 3, ed. 3 (Risk Based Approach)

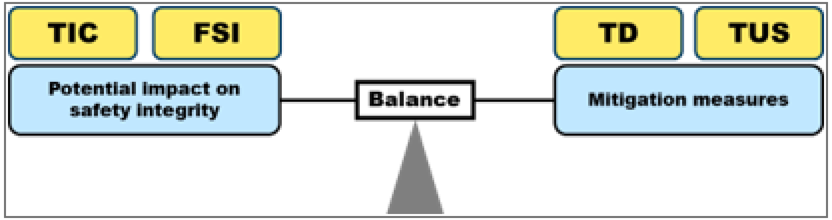


Figure H.1 – Tool Confidence Principle

TIC: the SW off-line support tool error impact class

FSI: The FS impact of the software off-line support tool

TD: Efficiency of detection / prevention means by the user of the software off-line support tool.

TUS: : the software off-line support tool usage for safety

IEC 61508 Part 3, ed. 3 (Risk Based Approach)

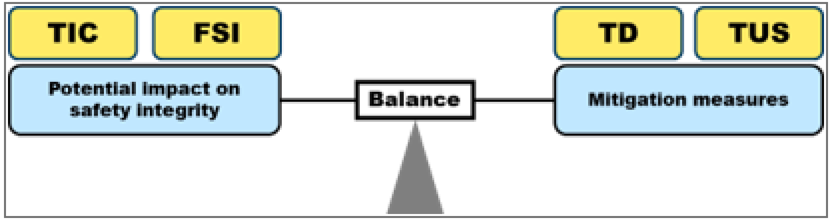


Figure H.1 – Tool Confidence Principle

Tool Impact class (TIC)

a) **TIC3**: Automated transformation or generation software off-line support tool which can introduce errors into a safety function.

b) **TIC2**: to detect errors in a safety function (or element SF). It can fail to detect them.

c) **TIC1:** it is considered that SW off-line support tool errors have negligible impact compared to human errors.

Note: Software off-line support tool with TIC1 are not subject to qualification

IEC 61508 Part 3, ed. 3 (Risk Based Approach)

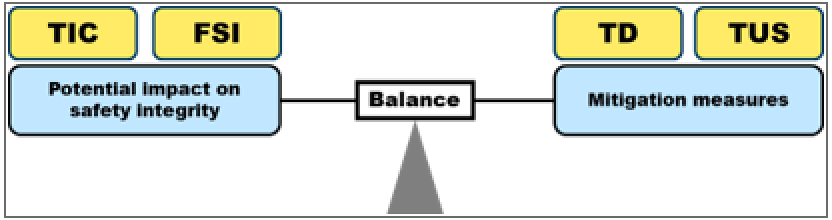


Figure H.1 – Tool Confidence Principle

TD definition:

TD1 : The errors that the tool can introduce or fail to detect, are detected on the tool's direct input/output.

TD2: The errors that the tool can introduce or fail to detect, are detected on data derived from the tool's input/output.

TD3: TD3 should be selected if TD1 or TD2 detection means are not applied.

IEC 61508 Part 3, ed. 3 (Risk Based Approach)

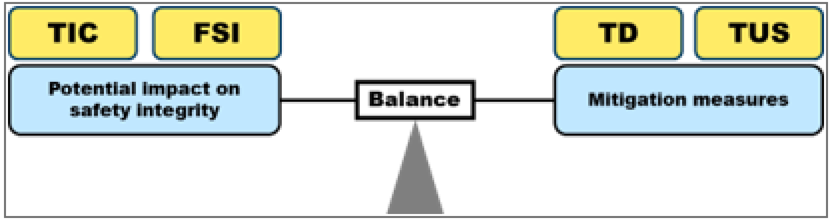


Figure H.1 – Tool Confidence Principle

The software off-line support tool usage confidence

Table 2 –TD/SIL/TIL Balancing Rules

TI	FSI	Minimum TUS for		
		TD1	TD2	TD3
2	1	-	1	1
	2	-	1	1
	3	1	2	2
	4	1	2	3
3	1	-	1	1
	2	-	1	2
	3	1	2	3
	4	1	3	4

A software off-line tool is said to be "qualified for TUS x" if it satisfies the SC requirements shown in Table 3, next page.

IEC 61508 Part 3, ed. 3 (Tool Qualification Example)

Tool name	Tool role	Tool version	Tool Impact if not mitigated	TIC	Error mitigation means	TD
Notepad++	Edit source code	5.6	incorrect source code	3	Review source code	3
gcc	Compile source code	9.3	incorrect object code	3	Test object code	3
IBM RTRT	Unit test	3.4	May miss errors in object code	2	Qualify tool	2
MyCodeGen	Code generator with verification of its output	7.8	May introduce error into source code	3	Review source code	3

#	Required confidence data	TUS-1	TUS-2	TUS-3	TUS-4
1	Tool usage plan and classification (H.3.1, H.3.2)	HR	HR	HR	HR
2	Tool user documentation (H.3.3)	HR	HR	HR	HR
3	Tool integration and assessment in usage context (H.3.4)	R	HR	HR	HR
4	Tool configuration management in usage context (H.3.5)	R	HR	HR	HR
5	Evidence of tool development for avoidance of systematic faults (H.4.1). -Tool specifications and verification	PR	R	HR	HR
6	Evidence of tool development for avoidance of systematic faults (H.4.1). -Architecture specification and verification -Module design and testing with coverage	PR	R	R	HR
7	Evidence of confidence from tool usage history (H.4.2)	R	HR	R	R
8	Tool problems management (H.4.3)	R	HR	HR	HR

Solu

Tools Qualification – Summary

- International standards usually define requirements for the development of safety related systems – procedures to classify, validate, certify or qualify tools.
- SW tool may have a negative impact on the safety related system (SR)
- Malfunctioning software tools can influence SR system by implementing errors or failing to detect errors .
- No single common approach used for tool validation, certification, and qualification among the different safety standards.
- Different techniques of importance to tool qualification applied to gain confidence in the tools used.
- Where applicable, software developers using tools are responsible for qualifying the software tools they are using.
- Tools vendors, can support these efforts by providing certification or qualification packages of their tool to support software development.

Questions?

Slot	Start Time	Paper	Workshop	Finish Time
-	15:05	Short Comfort Break (Oak Room)		15:25
10	15:25	*The Importance of Alarms for Functional Safety*	*SIL Calculations and use of IEC 61508 Part 6*	15:55
11	16:00	*Functional Safety and Communication Links*	*Discussion on Difference Between ISO 26262 and IEC 61508*	16:30