How to Identify Safety & Mission Critical Software Hazard Causes for Munition System

Abstract

Ann Marie Neufelder, Softrel, LLC, 321-514-4659, sales@softrel.com

Problem statement

Traditionally software safety and software reliability are assessed independently and by different personnel. The unwritten assumption is that safety and reliability failures are inherently due to different root causes. Due to funding issues the reliability analysis are either not performed at all or performed to an insufficient level of rigor. Hence, the following problems can result:

Potential problems with software safety analysis:

- Catastrophic failure modes that aren't safety related may be ignored
- Faulty assumption that "big safety failures" are caused by "total failures of the software" when historically the big safety failures have been caused by
  - Small software defects that weren't handled properly or at all
  - Neglecting to consider the 4th dimension – how the software is used over time
  - Neglecting to consider the 5th dimension – how people and systems interact with it
- Faulty assumption that failures are due to one requirement or one line of code or one use case step – when historically this isn't often the case

Potential problems with software failure modes effects analysis:

- Conducted from black box point in view (when it should be from functional point of view)
- Faulty assumptions
  - The software engineers will write code that perfectly meets the requirements
  - The software engineers will exhaustively test their code
  - The software requirements aren't themselves flawed
- Are often underfunded *($ go to safety analyses)*

Potential problems with both analyses:

- Faulty assumption that the root causes for mission failures and safety failures are different
- The truth is that the root causes are identically the same – only the effect differs

Solution

This session will present an effective and proven solution for munitions systems as well as any other software intensive mission and safety critical system.

1) Analyze software root causes common to both safety and mission failures from the below sources.
   a) Section 4 of NATO - AOP-52 GUIDANCE ON SOFTWARE SAFETY DESIGN AND ASSESSMENT OF MUNITION-RELATED COMPUTING SYSTEMS, 2016
   b) Appendix E of JOINT SOFTWARE SYSTEMS SAFETY HANDBOOK (SSSH) - DOD JOINT SOFTWARE SYSTEMS SAFETY , 2010
2) Identify effects of the software root cause
3) Assess which effects pertain to safety, mission or both
4) Update assessment as applicable through design, coding, test procedure development
5) Mitigate based on combined ranking of criticality and controls