20028
Joint Software Systems Safety Engineering Handbook Implementation Guide

Robert E. Smith, CSP
Booz Allen Hamilton
20th Annual NDIA Systems Engineering Conference
Springfield, VA

25 October 2017
System Safety, to include Software Safety, is required for acquisition programs IAW DoDI 5000.02 and MIL-STD-882E.

Detailed guidance for software safety is provided in the Joint Software Systems Safety Engineering Handbook (JSSSEH) Version 1.0 published 27 August 2010 as referenced in MIL-STD-882E.

- Comprehensive handbook, although lengthy at 344 pgs
- Acquisition Programs unfamiliar with software safety find it difficult to extract software safety techniques and processes in order to satisfy MIL-STD-882E Software Level of Rigor (LOR) requirements
- Programs typically re-state the LOR table from MIL-STD-882E, Table V in their Safety Plans and do not identify and specify the artifacts and Objective Quality Evidence (OQE) to be produced for all LOR tasks
- Could result in not performing a comprehensive software safety program and therefore not fully characterizing software’s contribution to system risk

Joint Boards recognized this concern and developed a JSSSEH Implementation Guide on 1 April 2016 to further assist programs, and was endorsed by the Joint Services Weapon Safety Review (JSWSR) Boards on 29 June 2016.

Software Safety Requirements

- **Software Safety is required for acquisition programs**
  - DoDI 5000.02, Enclosure 3, Para 11 – SOFTWARE “…The SEP should address the following: software unique risks; inclusion of software in technical reviews; identification, tracking, and reporting of metrics for software technical performance, process, progress, and quality; **software safety** and security considerations; and software development resources.”
  - DoDI 5000.02, Enclosure 3, Para 16 - ENVIRONMENT, SAFETY, AND OCCUPATIONAL HEALTH (ESOH) “The Program Manager will integrate ESOH risk management into the overall systems engineering process for all engineering activities throughout the system’s life cycle. As part of risk reduction, the Program Manager will eliminate ESOH hazards where possible, and manage ESOH risks where hazards cannot be eliminated. **The Program Manager will use the methodology in MIL-STD-882E...**”
  - MIL-STD-882E, Section 4.4 Software contribution to system risk. “The assessment of risk for software, and consequently software-controlled or software-intensive systems, cannot rely solely on the risk severity and probability….. Therefore, **another approach shall be used for the assessment of software’s contributions to system risk** that considers the potential risk severity and the degree of control that software exercises over the hardware.”
Common Approaches to Software Safety

- MIL-STD-882E references the JSSSEH and Section 4.4.2 includes a note to “Consult the Joint Software Systems Safety Engineering Handbook and AOP 52 for additional guidance on how to conduct required software analyses.”
- The JSSSEH is a lengthy document making it difficult for programs not familiar with software safety activities to extract detailed LOR tasks and tailor for particular program needs
- Programs often default to only referencing or reusing the LOR table from MIL-STD-882E (i.e., Table V) as their software safety approach in their System Safety Management Plans (SSMPs) and/or System Safety Program Plans (SSPPs)
- May result in not performing the specific LOR tasks that comprise a comprehensive software safety program, resulting in failure to assess software’s contribution to system risk(s)
MIL-STD-882E, Table V, Software Safety Criticality Matrix

<table>
<thead>
<tr>
<th>SOFTWARE CONTROL CATEGORY</th>
<th>SEVERITY CATEGORY</th>
<th>Level of Rigor Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Catastrophic (1)</td>
<td>Program shall perform analysis of requirements, architecture, design, and code; and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>conduct in-depth safety-specific testing.</td>
</tr>
<tr>
<td>1</td>
<td>SwCI 1</td>
<td>Program shall perform analysis of requirements, architecture, design, and code; and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>conduct in-depth safety-specific testing.</td>
</tr>
<tr>
<td>2</td>
<td>SwCI 1</td>
<td>Program shall perform analysis of requirements and architecture; and conduct in-depth</td>
</tr>
<tr>
<td></td>
<td></td>
<td>safety-specific testing.</td>
</tr>
<tr>
<td>3</td>
<td>SwCI 2</td>
<td>Program shall conduct safety-specific testing.</td>
</tr>
<tr>
<td>4</td>
<td>SwCI 3</td>
<td>Once assessed by safety engineering as Not Safety, then no safety specific analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or verification is required.</td>
</tr>
</tbody>
</table>
MIL-STD-882E, Table V, Level of Rigor Tasks

<table>
<thead>
<tr>
<th>SwCI</th>
<th>Level of Rigor Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>SwCI 1</td>
<td>Program shall perform analysis of requirements, architecture, design, and code; and conduct in-depth safety-specific testing.</td>
</tr>
<tr>
<td>SwCI 2</td>
<td>Program shall perform analysis of requirements, architecture, and design; and conduct in-depth safety-specific testing.</td>
</tr>
<tr>
<td>SwCI 3</td>
<td>Program shall perform analysis of requirements and architecture; and conduct in-depth safety-specific testing.</td>
</tr>
<tr>
<td>SwCI 4</td>
<td>Program shall conduct safety-specific testing.</td>
</tr>
<tr>
<td>SwCI 5</td>
<td>Once assessed by safety engineering as Not Safety, then no safety specific analysis or verification is required.</td>
</tr>
</tbody>
</table>

• Note that the LOR tasks table contains no details on the specific tasks, artifacts and Objective Quality Evidence (OQE) to be produced for LOR (e.g., requirements analysis, architecture analysis, design analysis, safety-specific testing, and code analysis)
• The JSSSEH includes these details, but not in a specific location
• Challenge is getting Acquirers (Customer) and Developers (software developers) to specify how they will turn the objectives of MIL-STD-882E and the JSSSEH “guidance” into actual Software System Safety Engineering (SSSE) Requirements
Implementation Guide Overview

- Developed by the Joint Services – Software Safety Authorities (JS-SSA) Sub-Working Group in support of the JSWSR Boards on 1 April 2016 - endorsed by the JSWSR Boards on 29 June 2016
  - Short name – “Implementation Guide”
- Provides implementation guidance for Software System Safety program requirements specified in MIL-STD-882E and guidance detailed in the JSSSEH
- Updated in 2017 to address identified errors, Service comments and create more direct alignment with the Tasks in MIL-STD-882E
- Released as “Revision A” on 17 October 2017
Implementation Guide Outline and Methodology

- The implementable process task requirements are presented as a decomposition of parent and children activities, similar to a Work Breakdown Structure (WBS)
- Parent tasks are graphically represented depicting inputs to the tasks and the products that the task would typically produce
- Tasks identified as MIL-STD-882 requirements are coded in the graphics using an extreme bold border of the task box
- Task decomposition is to the level necessary for a basic understanding of the process, the tasks that implement the process, and the products the tasks would likely produce
- The requirements derived that apply to each task are specified and cross referenced to both the applicable MIL-STD-882E requirements and JSSSEH sections and paragraphs that provide guidance on meeting the requirements
14 Process Tasks identified in the Implementation Guide
- Process Task 1.0: Prepare the System Safety Management Plan (SSMP)
- Process Task 2.0: Prepare System Safety Program Plan (SSPP)
- Process Task 3.0: Preliminary Hazard Analysis
- Process Task 4.0: Functional Hazard Analysis (FHA)
- Process Task 5.0: LOR Allocations to Safety-Significant Functions
- Process Task 6.0: Preliminary Safety Requirements Analysis (SRA)
- Process Task 7.0: Perform In-Depth Hazard Analysis
- Process Task 8.0: Perform Detailed Safety Requirements Analysis
- Process Task 9.0: Perform Safety Requirements Traceability
- Process Task 10.0: Perform Code-Level Safety Analysis
- Process Task 11.0: Perform Software Test Planning
- Process Task 12.0: Monitor Safety-Significant Software Testing
- Process Task 13.0: Perform Residual Safety Risk Assessment
- Process Task 14.0: Participate in Life-Cycle Management and Support

Each Process Task has Process Subtasks to amplify details and/or additional steps associated with each Task
Process Tasks (2017 Guide)

• **13** Process Tasks identified in the Implementation Guide
  – Process Task 1.0: Prepare the System Safety Management Plan (SSMP)
  – Process Task 2.0: Prepare System Safety Program Plan (SSPP)
  – Process Task 3.0: Preliminary Hazard Analysis
  – Process Task 4.0: Functional Hazard Analysis (FHA)*
  – Process Task 5.0: Initiate Safety Requirements Hazard Analysis (SRHA)*
  – Process Task 6.0: Perform System and Subsystem Hazard Analyses*
  – Process Task 7.0: Finalize SRHA*
  – Process Task 8.0: Perform Final Safety Requirements Traceability*
  – Process Task 9.0: Perform Code-Level Safety Analysis
  – Process Task 10.0: Perform Software Test Planning
  – Process Task 11.0: Monitor Safety-Significant Software Testing
  – Process Task 12.0: Perform Safety Risk Assessment*
  – Process Task 13.0: Participate in Life-Cycle Management and Support

• Each Process Task has Process Subtasks to amplify details and/or additional steps associated with each Task

* Changes in 2017: Titles of tasks revised and previous Task 5.0 combined into Task 4.0, and SRA is now System Requirements Hazard Analysis (SRHA)
3.4. Process Task 4.0: Functional Hazard Analysis (FHA)

References* to specific sections in JSSSEH and MIL-STD-882E

Process flow diagram* provided

Process Task / Subtask referenced for each step

* NOTE - References still the same in 2017 Guide. Flow diagrams altered as appropriate.
3.4.1. Process Subtask 4.1: Functionally Decompose the System

The information contained in the FHA reflects the same level of maturity as the design architecture. This is expected, and reinforces that the FHA must be kept current through all phases of the development lifecycle, to include functional, physical, and contractual changes made under configuration control. Frequency of updates to the FHA should be specified within the SOW and contract. However, SSSE should update the software inputs to the FHA IAW the SW development process and schedule. The format of the FHA should reflect that which will provide the analysis “answers” required by the analyst and criteria of the contract.

The first step of the analysis is to decompose the system. If the system is mature enough, this first step may be a physical decomposition of the system. If the system has not yet been allocated to specific pieces of hardware, this decomposition will be functional. The system must be analyzed functionally from the perspective of both “what the system is documented to do functionally”, and “what you think the system can do functionally”. The former is an assessment of documented functionality from the functional specifications and the latter is assessed by analyzing the functionality of the physical components of the system. The analysis of the physical attributes of the system is likely to provide insight to “hidden” or undocumented functionality. This is especially true for systems heavily using COTS components.

### Functional Hazard Analysis

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.4: Example FHA Format

Figure 2.4 provides an example of a FHA format that will provide the analyst with the most basic of information required by the analysis. If the analyst (or the Acquirer) requires more than this simple example format can provide, add the appropriate columns to the format to identify and track the information required. The decomposition of the system is documented in Column one. System decomposition can be done in a WBS-like structure which may aid in structure, flow, traceability and assignment of responsibilities. For instance, on large, complex programs such as an Aircraft (Refer: Figure 2.2) the hazard “Loss of Engine” may be completely under the control of the Engine Integrated Product Team (IPT). The Engine IPT is more likely to support safety if the FHA can readily show the IPT which parts it is responsible for.

3.4.2. Process Subtask 4.2: Identification of All Functionality

Column two of the example FHA format in Figure 2.4 depicts where the system functionality is documented. For the initial FHA, the functionality may be “higher level” functions that haven’t yet been decomposed to lower level functionality. For an initial FHA, this is sufficient for this level of analysis maturity as lower-level functionality will likely take on the same criticality as their parent higher-level functionality.
# Appendix A – LOR Task Table [Partial]

## Legend:
- **PR**: Prerequisite Requirement – Required regardless of LOR or required in order to assess and determine LOR
- **R**: Required for assigned LOR
- **AD**: As directed by Customer/Contract
- **IV&V**: Independent Verification and Validation
- **N/A**: Not Applicable for this program or LOR

## Required System Safety Tasks to Support Software System Safety Per MIL-STD-882E

<table>
<thead>
<tr>
<th>Level of Rigor (LOR) Activity</th>
<th>Primary Responsibility</th>
<th>Support Responsibility</th>
<th>Level of Rigor</th>
<th>Representative Artifacts Produced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required System Safety Tasks to Support Software System Safety Per MIL-STD-882E</td>
<td>Developer System Safety Manager, Developer Software Safety</td>
<td>Developer Program Manager, Developer Hardware and Software Design Engineer, Developer Software Design Architect, Developer Configuration Management</td>
<td>PR</td>
<td>System safety program plan (SSPP) and software system safety program plan (SSWPP), SOx, SDx, Acquirer Approved SSPP/SSWPP</td>
</tr>
<tr>
<td>Section 3.0 Process and Process Tasks for Software System Safety</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Section 3.1, Prepare the SSPP: Subsections 3.1.1-3.1.11 (Best Practice)</td>
<td>Acquirer System Safety Manager, Developer System Safety Manager, Developer Software Design Architect</td>
<td>Developer System Safety, Acquirer SDxWG Review and Approval</td>
<td>PR</td>
<td>Documented Program-Specific Terms and Definitions, MIL-STD-882E definitions and terms are required, latest approved by appropriate authorities, Acquirer Approved SSPP</td>
</tr>
<tr>
<td>Section 3.1.2, Prepare the SSPP</td>
<td>Developer Software Safety, Developer Software Development</td>
<td>Acquirer SDxWG Review and Approval</td>
<td>PR</td>
<td>SDxW, CxW, SSPP, SSWPP, Acquirer Approved SSPP/SSWPP</td>
</tr>
<tr>
<td>Section 3.1.3: Develop safety entry/exit criteria for each program phase of the software development life cycle to include concept refinement, requirements, preliminary and detailed design, coding, Test &amp; V&amp;V, software release and support.</td>
<td>Developer Software Safety, Developer Software Development and Test Configuration Management</td>
<td>Acquirer SDxWG Review and Approval</td>
<td>PR</td>
<td></td>
</tr>
<tr>
<td>Section 3.1.4: Develop (or update) the Software Control Category (SCC) Definition to be used on the program</td>
<td>Developer Software Safety, Developer Software Development</td>
<td>Acquirer SDxWG Review and Approval</td>
<td>PR</td>
<td></td>
</tr>
</tbody>
</table>
LOR 1 Example [Partial]

- Table indicates required ("R") LOR activities for LOR 1, 2, 3, and 4
- E.g., Design Practice (DP)-11: Analyze all safety functional threads...
  - Required only for LOR 1
  - One of many LOR 1 activities required ("R") for LOR 1
  - Appendix A specifies the LOR activity, primary and support activities, applicable LOR, and artifact(s) to be produced
Change Management

- JS-SSA meets twice annually
- **Approved path for changes:**
  - Any user can submit comments
  - Comments collected from 4th QTR FY until end of 2nd QTR FY (comments, corrections, additions, deletions, etc.)
  - **Submit comments to JS-SSA Chair**
  - Proposed changes adjudicated between the Service JS-SSA Implementation Guide (IG) IPT
  - Changes approved by the JS-SSA Sub-group will then be integrated into the Implementation Guide and a new revision released in time for the Fall meeting (or end of year)
- **100+ proposed changes submitted during the FY2017 review period**
- Proposed changes were adjudicated via email and in a face-to-face meeting April 2017
- Draft JS-SSA IG update distributed to Working Group and approved in August 2017
- Release of 2017 Guide Update (Rev A) on 17 October 2017
2017 Summary of Changes

• Numerous changes between 2016 Guide and 2017 Guide

• Two “Critical” changes to the Implementation Guide
  – Less emphasis and more controls on tailoring of LOR table by contractors (Section 2.0)
    ▪ Changed from: “The LOR table should be tailored for any given program as agreed to by the Acquirer and Developer.”
    ▪ To: “The LOR table should be assessed for tailored implementation for any given program, and tailoring is permitted as long as the tailored LOR tasks are approved by both the Acquirer and Developer.”
  – Allows risks to be carried over, if appropriate, from one contractual activity to another following a reassessment (Section 3.2.4.2)
    ▪ Changed from: “Risk accepted in one contractual activity should never be carried over as the baseline for the next contractual activity.”
    ▪ To: “Risk acceptance performed in one contractual activity should be reassessed for the next contractual activity.”
2017 Summary of Changes (cont.)

- **Four “Significant” changes to the Implementation Guide**
  - SSMP tasks added to the LOR table in Appendix A as “Acquirer” activities
  - Removed requirement that Contractors must comply with future versions of DODI 5000.02 and MIL-STD-882, just the versions under contract
  - Clarified purpose of document as defining the processes and tasks needed to implement a MIL-STD-882E compliant SSSE program
  - Made the current Process Task 5.0 “LOR Allocations to Safety-Significant Functions” a subtask of draft Process Task 4.0 “FHA”

- **Majority of remaining changes are relatively minor and designed to resolve known inconsistencies and improve alignment with MIL-STD-882E**
  - Primarily changes to the process flow figures and associated paragraphs detailing the subtasks for the analyses/reports (PHA, SRA, etc.) to better define tasks and processes
  - Many editorial and administrative corrections

- **Changed “Hazard Risk Index” to “Risk Assessment Code”**

- **Changes to the LOR table in Appendix A**
2017 Summary of Changes – Appendix A

- Seven new Baseline LOR SSE-related activities detailing Acquirer (i.e., “ACQ-#”) responsibilities
- Some activity descriptions updated and enhanced, but overall, no other new activities added

<table>
<thead>
<tr>
<th>Level-Of-Rigor Activity / Task Type</th>
<th>2016 IG</th>
<th>2017 IG</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquirer (ACQ-#.#)</td>
<td>0</td>
<td>7</td>
<td>+7</td>
</tr>
<tr>
<td>System Safety Engineering (SSE-#.#)</td>
<td>22</td>
<td>22</td>
<td>-</td>
</tr>
<tr>
<td>Requirements Phase (RP-#)</td>
<td>11</td>
<td>11</td>
<td>-</td>
</tr>
<tr>
<td>Design Phase (DP-#)</td>
<td>13</td>
<td>13</td>
<td>-</td>
</tr>
<tr>
<td>Implementation (Coding) Phase (IP-#)</td>
<td>15</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>Test Phase (DP-#)</td>
<td>23</td>
<td>23</td>
<td>-</td>
</tr>
<tr>
<td>Life Cycle Support Phase (LC-#)</td>
<td>12</td>
<td>12</td>
<td>-</td>
</tr>
<tr>
<td>TOTAL ACTIVITIES / TASKS</td>
<td>96</td>
<td>103</td>
<td>+7</td>
</tr>
</tbody>
</table>
Several activities now required to be performed at lower LOR to align with MIL-STD-882E Table V LOR requirements

<table>
<thead>
<tr>
<th>Level-Of-Rigor</th>
<th>2016 IG</th>
<th>2017 IG</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>42</td>
<td>49</td>
<td>+7</td>
</tr>
<tr>
<td>1</td>
<td>54</td>
<td>54</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>47</td>
<td>49</td>
<td>+2</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
<td>38</td>
<td>+3</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>27</td>
<td>+7</td>
</tr>
<tr>
<td>TOTAL (LOR 1 + Baseline)</td>
<td>96</td>
<td>103</td>
<td>+7</td>
</tr>
</tbody>
</table>
**Conclusion**

- Software Safety is required for acquisition programs IAW DoDI 5000.02 and MIL-STD-882E
- Additional guidance for software safety is provided in the JSSSEH Version 1.0 published 27 August 2010 as referenced in MIL-STD-882E
- Joint Boards developed a JSSSEH Implementation Guide on 1 April 2016 to further assist program perform software safety, and was endorsed by the JSWSR Boards on 29 June 2016
- 2017 Implementation Guide Update (Rev A) release on 17 October 2017
- Implementation Guide will be updated annually, as required

*Implementation Guide assists in performing a comprehensive software safety program to fully characterize software’s contribution to system risk*
Resources (location of documents)

- DAU Acquisition Community Connection Site, ESOH Community
  - https://acc.dau.mil/ESOH
  - or
  - look under the “Resources” section

- DoD Joint Software System Safety Engineering Handbook, 2010

Questions?

Robert E. Smith
Lead Associate

Booz Allen Hamilton
1550 Crystal Dr, Suite 1100
Arlington, VA 22202
Tel (703) 412-7661
smith_bob@bah.com