10 June 2022

MEMORANDUM FOR RECORD

To: Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E))
From: DoD Joint Weapon Safety Working Group (JWSWG)

Subj: Endorsement of White Paper Guidance to Perform Functional Hazard Analysis for Weapon Systems with Artificial Intelligence Capabilities, 19 May 2022

Ref: (a) DoD Instruction 5000.69 "DoD Joint Services Weapon and Laser System Safety Review Processes" of 9 Nov 2011
(b) Military Standard 882E "Department of Defense Standard Practice for System Safety", 11 May 2012

Encl: (1) White Paper Guidance to Perform Functional Hazard Analysis for Weapon Systems with Artificial Intelligence Capabilities, 19 May 2022

Reference (a) establishes policy and assigns responsibilities for the Department of Defense (DoD) Joint Services Weapon and Laser System Safety Review Processes. Reference (a) requires Joint Service safety reviews for weapon and laser systems that will be used by two or more DoD Components and establishes a DoD Joint Weapon Safety Working Group (JWSWG) that will coordinate and liaise with the DoD Laser System Safety Working Group (LSSWG) on joint safety review processes. Additionally, reference (a) authorizes publication of supporting guidance to provide specific information on the DoD Joint Services weapon and laser system safety processes.

In May 2022 a Joint Artificial Intelligence (AI) System Safety Working Group developed new guidance for performing a Functional Hazard Analysis (FHA) for Weapon Systems (WS) with AI capabilities. The traditional FHA, as described in reference (b), is a foundational System Safety Engineering (SSE) analysis in a System Safety Program and is one of the most important analyses that the system safety analyst will perform. This new guidance provided in enclosure (1) is a Joint effort between the US Army, US Navy, DoD SMEs and the Office Under Secretary of Defense (Research and Engineering) (OUSD(R&E)) addressing SSE best practices when developing an FHA for systems that incorporate AI supported functions. This document is being released to provide intermediate guidance to the DoD community with the understanding that it addresses a subset of potential AI technologies and challenges. This document will be updated as future guidance and methods are developed.

Subj: Endorsement of Guidance to Perform Functional Hazard Analysis for Weapon Systems with Artificial Intelligence Capabilities

The FHA for WS that incorporate AI capabilities guidance provided in enclosure (1) are endorsed by the undersigned as Co-Chairs of the DoD JWSWG. Based on this endorsement, the DoD JWSWG Co-Chairs recommend formal processing and issuance of enclosure (1).

Digitally signed by
HAWLEY.ERIC.J.1014133948
Date: 2022.06.06 09:32:43
-04'00'

Mr. Eric J. Hawley
(Navy), Co-Chair (Acting),
DoD Joint Weapon Safety
Working Group

Digitally signed by
HAMILTON.IAN.T.1257726874
Date: 2022.06.02 16:22:47 -04'00'

Mr. Ian T. Hamilton (SSTM)
(Army), Co-Chair, DoD Joint
Weapon Safety Working
Group

RADEMACHER.ST
EVEN.E.109970578
3

Digitally signed by
RADEMACHER.STEVEN.E.1099
705783
Date: 2022.06.02 16:35:42 -06'00'

Col Mark C. Murphy
(Air Force), Co-Chair, DoD
Joint Weapon Safety
Working Group

White Paper

GUIDANCE TO PERFORM

FUNCTIONAL HAZARD ANALYSIS

For

WEAPON SYSTEMS WITH ARTIFICIAL INTELLIGENCE CAPABILITIES

19 May 2022

Distribution Statement A:

Approved for public release; distribution is unlimited

March 2022.

Other requests shall be referred to NOSSA.

POC: Jerome P (Jay) Ball, jerome.p.ball.civ@us.navy.mil

Accreditations

Jerome (Jay) P Ball
Naval Ordnance Safety and Security Activity (NOSSA)
Deputy Executive Director/CHENG

Gunendran Sivapragasam
Technology System Safety Technical Lead
Technology Integration Safety Branch (R44)
NSWCDD Dam Neck Activity

Bruce Nagy
Research Engineer
NAWCWD, China Lake

Anh Belanger
System Safety/Software System Safety Lead
U.S. Army Aviation and Missile Command
Redstone Arsenal, AL

Wilfredo (Wil) Vega
System Safety Lead
USD (R&E)/DDRE (Advanced Capabilities)/
DD(Engineering) Engineering Policy & Systems/Specialty Engineering
Picatinny Arsenal, NJ

Benjamin Werner
Technical Lead, Quality Engineering and Systems Assurance Directorate
U.S. Army Combat Development Command (DEVCOM) Armaments Center
Picatinny Arsenal, NJ

Aaron Ortiz
System Safety Engineering Intern
Safety Engineering Office
Materiel Systems Organization (MSO)
U.S. Army Tank-automotive and Armaments Command (TACOM)

Jason Rupert
Senior Airworthiness Engineer
Modern Technology Solutions, Inc.
360 Quality Circle NW
Huntsville, AL

GUIDANCE TO PERFORM
FUNCTIONAL HAZARD ANALYSIS
For
WEAPON SYSTEMS WITH ARTIFICIAL INTELLIGENCE CAPABILITIES

FOREWORD: The Functional Hazard Analysis (FHA), per Task 208 in MIL-STD 882E, is a foundational System Safety Engineering (SSE) analysis in a System Safety Program (SSP) and is one of the most important analyses that the system safety analyst will perform. This document is a Joint effort between the US Army, US Navy and the Office Under Secretary of Defense Office for Research and Engineering addressing system safety engineering best practices when developing the FHA for systems that incorporate Artificial Intelligence (AI) supported functions. This document is being released to provide intermediate guidance to the DoD community with the understanding that it addresses a subset of potential AI technologies and challenges, however the guidance is none the less useful and needed. This document is intended to be updated as future guidance and methods are developed.

1. REFERENCES

a. MIL-STD-882E “Department of Defense Standard Practice System Safety”, 11 May 2012.

b. Joint Software System Safety Engineering Handbook, Ver 1, August 2010.

c. Joint Services – Software Safety Authorities Software System Safety Implementation Process and Tasks Supporting MIL-STD-882E Rev B, 14 March 2018.

2. DEFINITIONS

The following definitions are unique to this paper and have been defined through this joint effort:

a. Safety Control Entity – an independent and distinct external function that can actively intervene and interrupt a task or function execution with mechanisms to mitigate, control, or bring the system to a known safe state, in order to prevent a mishap occurrence. The function will have sufficient time and means to act. An operator, software, firmware, hardware, another AI function, etc can perform this function if the analysis supports independence and level of rigor requirements. Policy may limit what may act as a safety control entity in specific applications; this guidance does not alleviate any policy requirements.

b. Function - Intended behavior of a product based on a defined set of requirements regardless of implementation. [ARP 4754]

Associated with this concept, while not adopted specifically here but rather in principle, is the following: “FUNCTION Development Assurance Level: The level of rigor of development assurance tasks performed to Functions.” [ARP 4754] The concept is that Criticality levels are determined by Functions which are design agnostic, Level of Rigor tasks are applied based on what is implementing the Function.

c. Interlock – a barrier (design feature) to prevent the mishap from occurring. Interlocks can be viewed as an “AND” gate where at least two or more independent conditions have to be satisfied for a mishap to occur. An interlock must be developed at an appropriate level of robustness (level of rigor) commensurate with the function it is protecting, or “AND” gated with. Interlocks may be hardware, software, firmware, AI, or humans as specific policy allows.

d. Independence - 1. A concept that minimizes the likelihood of common mode errors and cascade failures between [system] functions or items, 2. Separation of responsibilities that assures the accomplishment of objective evaluation e.g. validation activities not performed solely by the developer of the requirement of a system or item.[ARP 4754]

NASA on the use of software to control hazards:

“While software controls can be, and are, used to prevent hazards, they must be implemented with care. Special attention needs to be placed on this software during the development process. When there are no hardware controls to back up the software, the software must undergo even more rigorous development and testing.” [NASA software handbook]

“When software is used to control a hazard, some care must be made to isolate it from the hazard cause it is controlling.” [NASA software handbook]

“Partitioning of the hazard control software is recommended. Otherwise, all of the software must be treated as safety-critical because of potential “contamination” from the non-critical code.” [NASA software handbook]

“If the hazard cause is erroneous software, then the hazard control software can reside on a separate computer processor from the one where the hazard/anomaly might occur.” [NASA software handbook]

These citations are software specific, but practically, the intent to characterize and improve quality of any particular interlock applies to all interlocks. Human factors analysis for operator in the loop, Mean Time Between Failure (MTBF) for hardware and Level of Rigor (LOR) for software, firmware, and AI are examples. Specific care must be given to ensure the interlock is not subject to common cause or common mode failures to justify independence. For interlocks to reduce control levels they must act serially in the functional failure set. If an item (software, firmware, AI) contains multiple functions, it will inherit the Criticality of the most significant level unless there is appropriate partitioning.

3. Functional Hazard Analysis (FHA) Process

This paper recommends a focus on Functions to determine Control and Criticality, which is used to determine what LOR needs to be applied to AI, software, and firmware functions in the FHA. In support of this, a brief overview of current systems engineering methodology and several examples are provided for context.

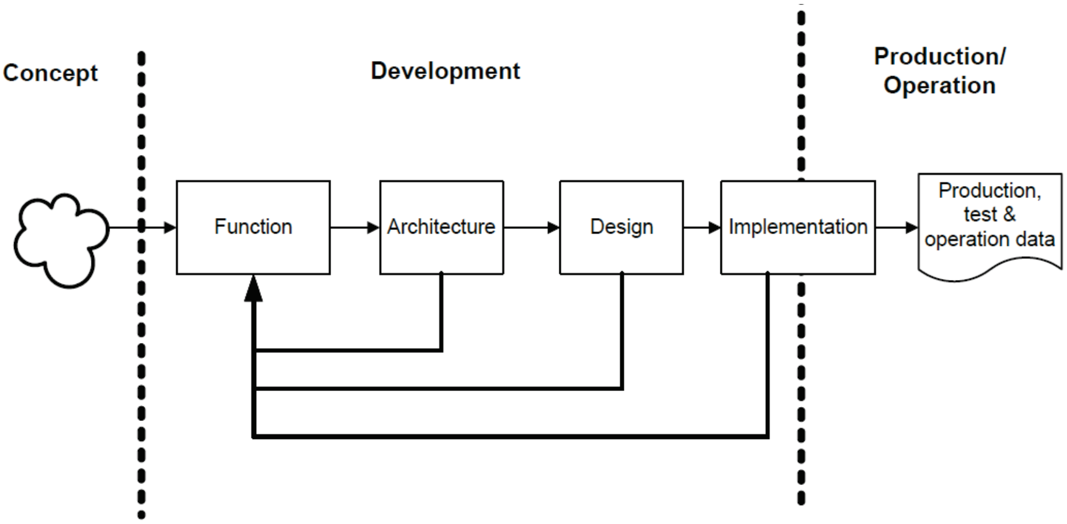
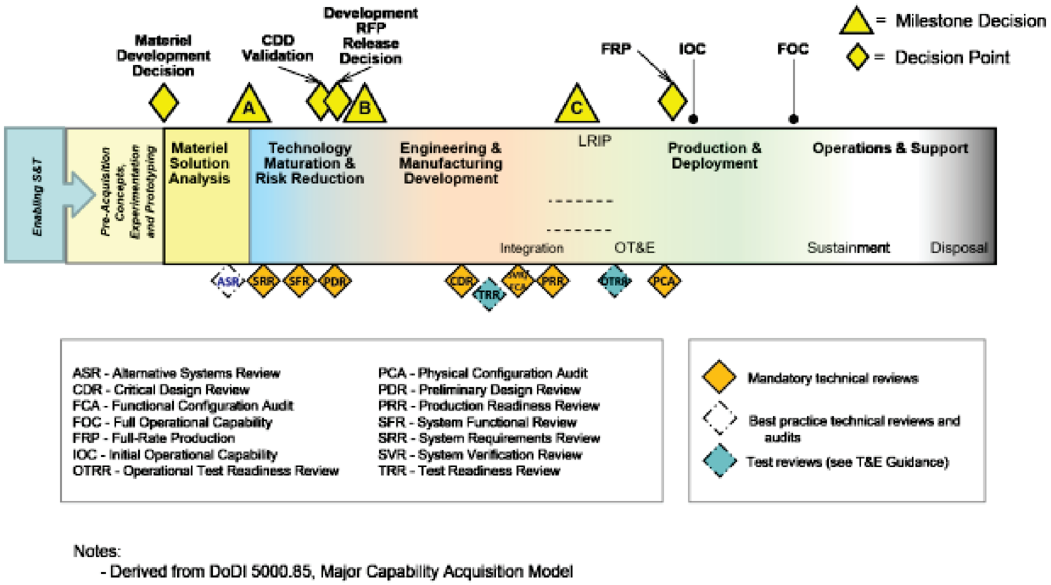


Figure 1: ARP 4754 Lifecycle Example

In this simplistic diagram of a system lifecycle the relationship between system functions and system architecture, design, and implementation is shown. This relationship is key to the execution of the FHA in the manner advocated here. The system functional allocation is used to influence architecture and design decisions; therefore, an early FHA is intended to be used to integrate safety into the architecture and to allow decision makers to balance the cost of LOR with the potential costs of design complexity and safety controls. The FHA here is focused on System Functions and will be agnostic of what technology implements the function.

This perspective aligns with current DoD implementation of System Engineering Technical Reviews (SETR) and the expected data provided during progressive milestones as shown in Figure 2.

Figure 3-1 provides the end-to-end perspective and the integration of SE technical reviews and audits across the system life cycle.


Materiel Development Decision
CDD Validation
Development RFP Release Decision
FRP
IOC
FOC
= Milestone Decision
= Decision Point
A
B
C
Enabling S&T
Pre-Acquisition Concepts, Experimentation and Prototyping
Materiel Solution Analysis
Technology Maturation & Risk Reduction
Engineering & Manufacturing Development
LRIP
Production & Deployment
Operations & Support
Integration
OT&E
Sustainment
Disposal
ASR
SRR
SFR
PDR
CDR
TRR
SVR FCA
PRR
OTRR
PCA
ASR - Alternative Systems Review
CDR - Critical Design Review
FCA - Functional Configuration Audit
FOC - Full Operational Capability
FRP - Full-Rate Production
IOC - Initial Operational Capability
OTRR - Operational Test Readiness Review
PCA - Physical Configuration Audit
PDR - Preliminary Design Review
PRR - Production Readiness Review
SFR - System Functional Review
SRR - System Requirements Review
SVR - System Verification Review
TRR - Test Readiness Review
Mandatory technical reviews
Best practice technical reviews and audits
Test reviews (see T&E Guidance)

Notes:
- Derived from DoDI 5000.85, Major Capability Acquisition Model

Figure 3-1. Major Capability Acquisition Life Cycle

Figure 2: DoD Systems Engineering Process [2022]

These processes are typically adapted to account for software as depicted Figure 3. Notice the offshoot into the software lifecycle after the System Functional Review (SFR), this is the reason why the FHA begins with System Functions first.

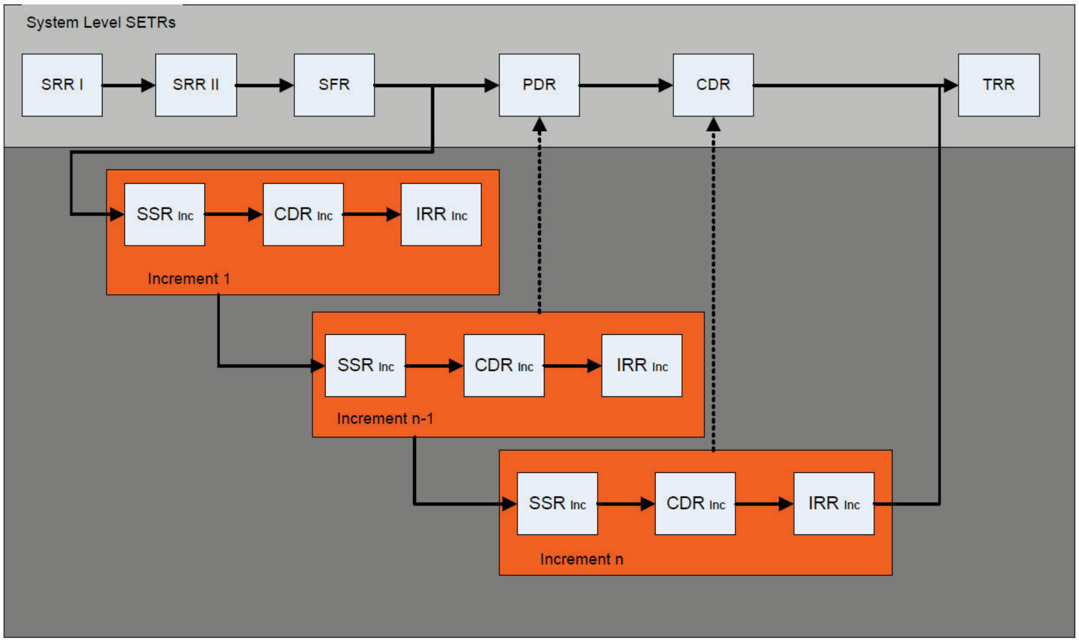
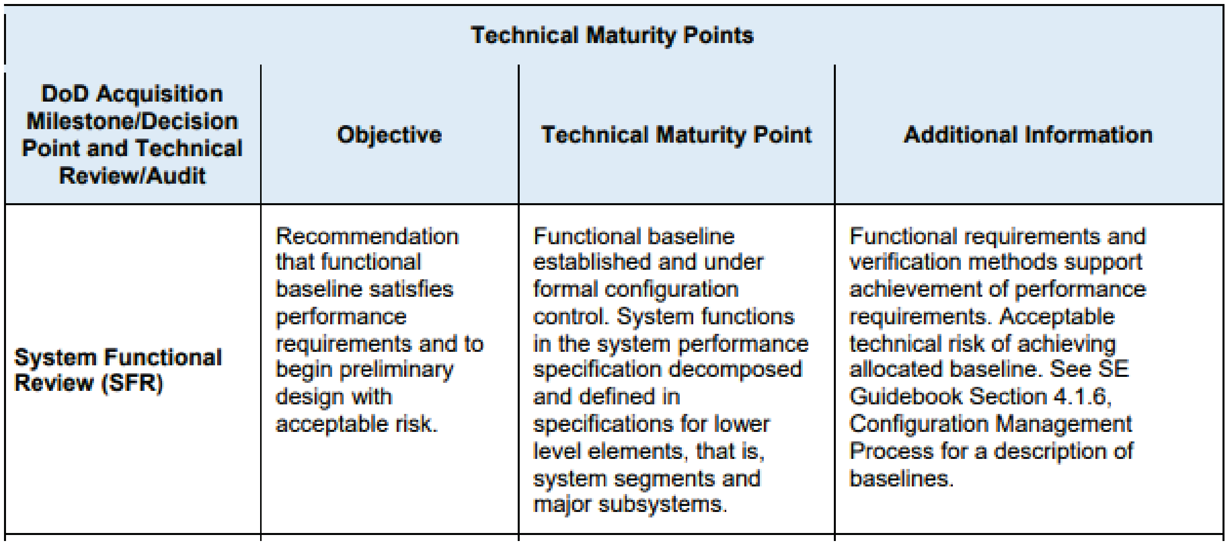


Figure 3: Software SETR

Key concepts here are associated with the SFR and its relationship to the software SETR process.



Technical Maturity Points			
DoD Acquisition Milestone/Decision Point and Technical Review/Audit	**Objective**	**Technical Maturity Point**	**Additional Information**
System Functional Review (SFR)	Recommendation that functional baseline satisfies performance requirements and to begin preliminary design with acceptable risk.	Functional baseline established and under formal configuration control. System functions in the system performance specification decomposed and defined in specifications for lower level elements, that is, system segments and major subsystems.	Functional requirements and verification methods support achievement of performance requirements. Acceptable technical risk of achieving allocated baseline. See SE Guidebook Section 4.1.6, Configuration Management Process for a description of baselines.

Figure 4: DoD SFR Requirements [2022]

The FHA begins with the System Functions per Figure 4, and determines the mishap severity of function failures and then function autonomy to assess the safety Criticality. Then following the System Engineering process of functional decomposition to element/component supporting functions, supporting functions inherit the System Function Criticality level. For integrated/interoperable Weapon Systems there must be consideration of the System of System Functions they support in determining Criticality. Following this process ensures the FHA is in accordance with the prescribed system safety processes in MIL-STD-882E. The intention is to influence early architecture decisions that impact the criticality of System Functions. This also allows mapping of those functions through the System Engineering lifecycle to allocate and focus analysis and development rigor where most appropriate.

It should be noted that Task 3 in Figure 5 below recommends changes to Control Category labels and definitions to accommodate AI.

From this point, when conducting the FHA for a system that contains AI enabled functions, the FHA process is built on the process described in the Joint Services – Software Safety Authorities Software System Safety Implementation Process and Tasks Supporting MIL-STD-882E, Reference 1c, and referred to here as Software Safety Implementation Guide (SSIG).

Figure 5 depicts the proposed FHA flow diagram which will accommodate systems that have AI enabled functions.

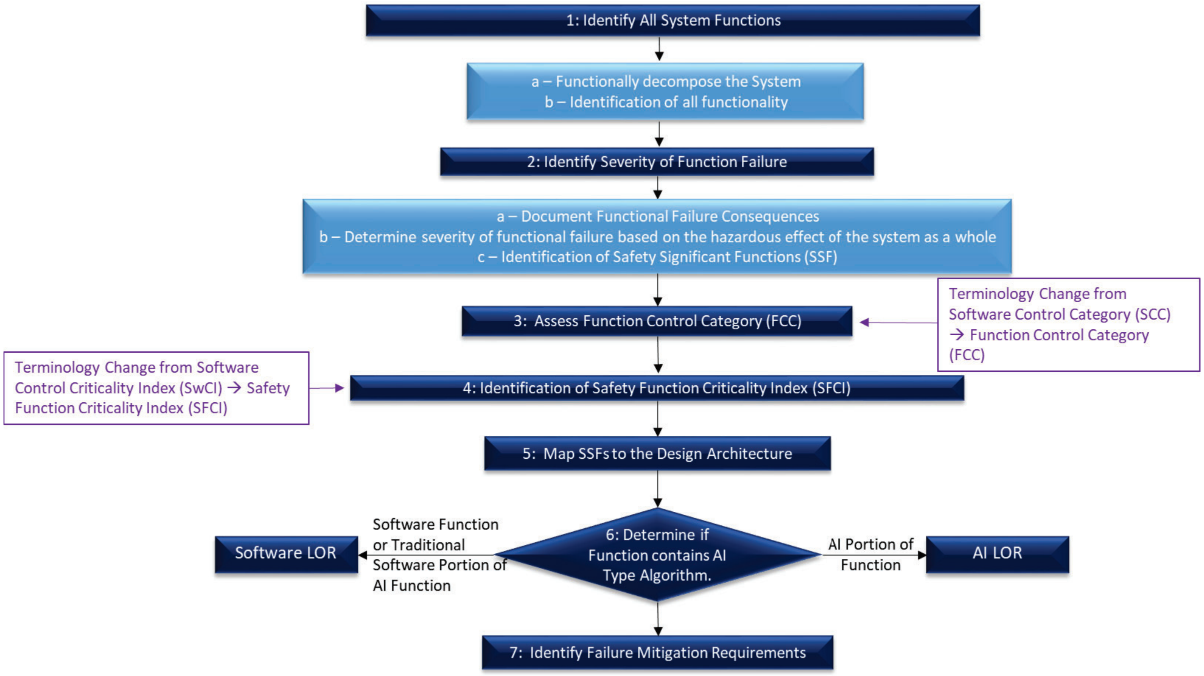

1: Identify All System Functions
a – Functionally decompose the System
b – Identification of all functionality
2: Identify Severity of Function Failure
a – Document Functional Failure Consequences
b – Determine severity of functional failure based on the hazardous effect of the system as a whole
c – Identification of Safety Significant Functions (SSF)
3: Assess Function Control Category (FCC)
Terminology Change from Software Control Category (SCC) → Function Control Category (FCC)
Terminology Change from Software Control Criticality Index (SwCI) → Safety Function Criticality Index (SFCI)
4: Identification of Safety Function Criticality Index (SFCI)
5: Map SSFs to the Design Architecture
Software LOR
Software Function or Traditional Software Portion of AI Function
6: Determine if Function contains AI Type Algorithm.
AI Portion of Function
AI LOR
7: Identify Failure Mitigation Requirements

Figure 5: Functional Hazard Analysis Process Steps

The tasks required to perform the FHA in this guidance are correlated to the FHA tasks in the SSIG as shown in Table 1.

Table 1: Functional Hazard Analysis Tasks Correlation

FHA Tasks in AI Guidance	FHA Tasks in the SSIG
1.a Functionally Decompose the System	4.1 Functionally Decompose the System
1.b Identification of all Functionality	4.2 Identification of all Functionality
2.a Document Functional Failure Consequences	4.3 Document Functional Failure Consequences
2.b Determine Severity of Functional Failure	4.4 Determine Severity of Functional Failure
2.c Identification of Safety Significant Function (SSF)	4.5 Identification of SSF
3 Assess SSFs against **Function Control Categories (FCC)**	4.8.1 Assess SSFs against **Software Control Categories (SCC)**
4 Combine SCC and Severity to assign **Safety Function Criticality Index (SFCI)**	4.8.3 Combine SCC and Severity to assign **Software Criticality Index (SwCI)**
5 Map SSFs to Design Architecture	4.7 Map SSFs to the Software Design Architecture
6 Determine if Function contains AI Type Algorithm	
7 Identify Failure Mitigation Requirements	4.9 Identify Failure Mitigation Requirements

Tasks 1.a, 1.b, 2a, 2.b, 2c, 5, and 7 are the same as the associated tasks from the SSIG. The following discussion focuses on tasks 3, 4, and 6 that are departures from MIL-STD-882E and the SSIG to integrate AI into the FHA process.

Task 3 - Assess Function Control Category:

The difference between Task 3, assessing Functional Control Category (FCC), and the associated tasks in the SSIG (Process Subtask 4.8.1) is updated labelling of Control Categories to be applicable generally to Functions including traditional software, firmware, and functions supported or enabled by AI (See table 2). Additionally, definitions of control levels were clarified to address how interlocks can be used to bring down the level of criticality.

Table 2: Functional Control Categories Definition

FUNCTIONAL CONTROL CATEGORIES		
Level	Name	Description
1	Autonomous (AT)	- Function exercises control authority over safety-significant hardware systems, subsystems or components without the possibility of predetermined safe detection and intervention by an independent safety control entity to preclude the occurrence of a mishap. - Function that displays safety-significant information that does not allow time for the operator (time is critical) to execute any action (e.g. independently validate display data) that would prevent or eliminate the occurrence of a mishap. - In the case of function failure, there is no functioning interlock that would prevent or eliminate the occurrence of a mishap.
2	Semi-Autonomous (SAT)	- Function exercises control authority over safety-significant hardware systems, subsystems or components, allowing time for predetermined safe detection and intervention by an independent safety control entity to preclude the occurrence of a mishap. - Function that displays safety-significant information, allowing the operator (with sufficient time) to execute an action for mitigation or control over a mishap. The operator must be trained to perform this action. - In the case of function failure, there is at least one functioning interlock that would prevent or eliminate the occurrence of a mishap.
3	Redundant Fault Tolerant (RFT)	- Function that issues commands over safety-significant hardware systems, subsystems, or components but requires a safety control entity to complete the command function. The system must provide the safety control entity sufficient notification of a failure or potential unsafe state. The system must additionally include one or more interlocks that would preclude the occurrence of a mishap. - Function that generates information or display of a safety-significant nature used by a safety control entity to make safety significant decisions. The system includes two or more interlocks that would preclude the occurrence of a mishap. - In the case of function failure, the system includes two or more independent interlocks that preclude the occurrence of a mishap.
4	Influential	- Function generates information of a safety-related nature used to make decisions by the operator but does not require operator action to avoid a mishap. - In the case of function failure, the system includes three or more independent interlocks that preclude the occurrence of a mishap.
5	No Safety Impact (NSI)	- Function does not possess command or control authority over safety-significant hardware systems, subsystems, or components and does not provide safety-significant information. Function does not provide safety-significant data or information that requires control entity interaction. Function does not transport or resolve communication of safety-significant data.

It should be noted that when identifying control categories, fulfilling any one of the definitions would be sufficient to determine which level a function falls into. For example, to be identified as Semi-Autonomous (SAT) the function must meet any one of the following definitions:

- Function exercises control authority over safety-significant hardware systems, subsystems or components, allowing time for predetermined safe detection and intervention by an independent safety control entity to preclude the occurrence of a mishap or hazard.

Or

- Function that displays safety-significant information, allowing the operator (with sufficient time) to execute an action for mitigation or control over a mishap or hazard. The operator must be trained to perform this action.

Or

- In the case of function failure, there is at least one functioning interlock that would prevent or eliminate the occurrence of a mishap or hazard.

<u>Interdependency Analysis (IA)</u>:

The IA is an analytical technique that may be used to identify the FCC. This analysis provides a structured approach to identify the interlocks that are present to prevent a mishap from occurring in the case of functional failure. IA is not required as a technique; however covering it provides the user a clearer understanding of the influence of design or architecture changes on the control level of the function. Table 3 demonstrates an example of IA.

Table 3: Interdependency Analysis Example

Safety Function Title	CSCI	CSC	CSU	Description	Safety Function Design Alternatives							
					Alternative A			FCC A	Alternative B			FCC B
					Hardware, Software or Human Interlocks				Hardware, Software or Human Interlocks			
e.g. Autonomous UxS Navigation Using AI	UxS Navigation	Obstacle Avoidance	DNN	AI Function - Autonomous Navigation using Deep Neural Net (DNN)	Operator monitors navigation in real-time and can intervene to prevent mishap			2	Operator monitors navigation in real-time and can intervene to prevent mishap	'Guard Rails' - Independent Software Module monitors / limits AI function output (blocks extreme outputs that could result in UxS crashing)		3

The Computer Software Configuration Item (CSCI), Computer Software Component (CSC) and Computer Software Unit (CSU) of the function are first identified so the FCC can be determined at the lowest level possible in the design where it occurs. This is especially important for safety critical functions as this allows the increased LOR requirements for safety critical functions to be applied to the smallest component of the function as possible. It should be noted that this assumes appropriate partitioning methods are employed that address both spatial and temporal isolation, for example as defined in the Avionics Application Standard Software Interface (ARINC 653).

Once the components of the function are understood, then the columns in green are filled by identifying interlocks that are planned within the design to prevent a mishap in the case the function in question fails. In the example, if the Deep Neural Network (DNN) fails, the operator will be able to intervene and thus, the presence of one interlock allows the FCC to be assigned a value of two.

The columns in Alternative B allows the safety analyst to evaluate the impact of adding interlocks on the FCC, reducing the LOR tasks that are required to be completed prior to assessment of risk level of the function. The example shows the addition of an independent software function that monitors and limits the outputs of the DNN to prevent the possibility of a crash due to DNN misbehavior. This results in two independent interlocks, allowing the FCC to be reduced to a value of three.

<u>Task 4 Identification of Safety Function Criticality Index (SFCI)</u>:

Once the FCC and the Severity of consequence is determined for a SSF the Criticality can be determined by the predefined and approved Software Safety Criticality Matrix (SSCM) in MIL-STD-882E Table V. It is recommended here that the title be changed to the Safety Function Criticality Matrix (SFCM) to accommodate AI functions and account for functions residing in firmware.   Additionally, the Software Criticality Index (SwCI) is renamed as the Safety Function Criticality Index (SFCI) for consistency.

	SEVERITY CATEGORY			
FUNCTION CONTROL CATEGORY	Catastrophic (1)	Critical (2)	Marginal (3)	Negligible (4)
1	SFCI 1	SFCI 1	SFCI 3	SFCI 4
2	SFCI 1	SFCI 2	SFCI 3	SFCI 4
3	SFCI 2	SFCI 3	SFCI 4	SFCI 4
4	SFCI 3	SFCI 4	SFCI 4	SFCI 4
5	SFCI 5	SFCI 5	SFCI 5	SFCI 5

Table 4: Safety Function Criticality Matrix

<u>Task 6 Determine if Function contains AI Type Algorithm</u>:

The safety analyst should now determine whether the function in question has AI technology in it.

The assessment process is to determine if an AI model covered by this guidance is present within a software function or module, or within firmware. To determine if the function is supported or enabled by an AI model the following criteria is recommended:

> Criteria 1 - The function uses data approximations to build/train its model, e.g. data approximations can come from simulations and synthetic data.
>
> Criteria 2 - Data samples are used to build/train its model and these data samples are a subset of the actual population size, e.g., training data samples from population to support machine learning, training data samples requiring clutter backgrounds.

One way to think about Criteria 1 is to ask: "Could another developer create a different set of statistics under the same conditions?" If no, then maybe this is not an AI function. If yes, then it meets the condition. As an example, if a statistical model of the trained function was developed, how accurate were the approximations used in creating the function. In other words, how close do these approximations fit the real-world physics regarding operational deployment? If the trained function is based on simulation results, then synthetic data that is not representative of real-world data will result in an inferior model. The goal is to have good quality and comprehensive training data that would result in a robust model.

One way to think about Criteria 2 is to ask: "What is the actual population size of the training set?" If the training set is equal to the actual population size, then maybe this is not an AI function. Consider the most basic ML algorithm, a regression line. If all the points that will ever occur for this function are on the scatter plot used to approximate the curve, why use a regression line? If all the ML algorithm inputs and outputs are known, why use ML and not traditional code? If traditional code can address the needs of the function, then traditional code should be prioritized in safety significant functions over AI.

Table 5 provides some examples of how the criteria can be used to identify AI functions. It should be noted that these are examples and a given AI type may align to a different criteria in a different development environment. It is important not take this chart as a reference for future results, it is imperative that SMEs assess each implementation on its own merits based on its development process.

Table 5: Examples of AI Function Identification

AI Type Examples of Specific Algorithms	Algorithm built based on using data approximations	Algorithm built based on using data samples from larger population	Final Score
Convolutional Neural Network	x (if synthetic data used for CNN)	x (training data samples)	2
Deep Neural Network	x (if synthetic data used for DNN)	x (training data samples)	2
Deep Neural Network with Reinforced Learning	x (Consider an RL that uses truth data to reinforce its decision sequence to use only a subset of the truth data)	NA	1
Deep Neural Network with Reinforced Learning	NA	x (Consider an RL that creates a synthetic version of its data population to be creating only a subset)	1
Recurrent Neural Network (Long Short Term Memory)	x (if synthetic data used for RNN)	x (training data samples)	2
Naïve Bayes	x (if modeling and sim data used to produce statistics for Naïve Bayes)	x (training data samples)	2
Expert Systems	x (Consider expert systems to be acquiring knowledge to approximate a SME's thought process)	. NA	1
Traditional Software – e.g. ship Pointing and Firing Cutout (only fire within pre-defined Azimuths and Elevations)	NA	NA	0

Satisfying either or both of the two criteria (Final Score = 1 or 2) means that the function contains an AI type meeting the criteria. While there may be AI types that do not fit these criteria, the defined Level of Rigor Tasks are based on AI types that do meet these criteria, as these are the most commonly used and likely to be deployed in the near future. The FHA guidance here is generally applicable to other AI types, however, if you believe you are deploying an AI type not covered by these criteria it is important to work with your appropriate tech authority or system safety oversight to determine appropriate means to apply Level of Rigor for your application. This guide is meant to be updated as lessons are learned.

Having determined the function is an AI type meeting these criteria the safety analyst should select the AI LOR for the appropriate criticality. If neither of the above criteria is satisfied (Final Score = 0) and the analyst is certain it is not another AI type, then the safety analyst should return to using the Software LOR contained in the SSIG

It is important to consider that the software/firmware used to implement the AI functions must be at the same Safety Function Criticality Index level as the AI function. For software refer to Mil-Std-882 and the SSIG. For example, if the AI function is identified as SFCI 1 then all software and firmware used to implement the AI function will also be identified as SFCI 1 (or SwCI 1 in the SSIG Appendix A). Tools used in development, e.g., model and sim, data curation scripts etc. should be qualified, validated, and approved at an appropriate level commensurate with the SFCI, see the SSIG and/or DO-330 for guidance.

Finally, tasks 5 and 7 remain unchanged from the SSIG.