MISSION READY SOFTWARE

INTUITIVE®

Rapid Reliability Risk Assessment Utilizing Software Tools and MBSE in RAM Engineering

11/06/25

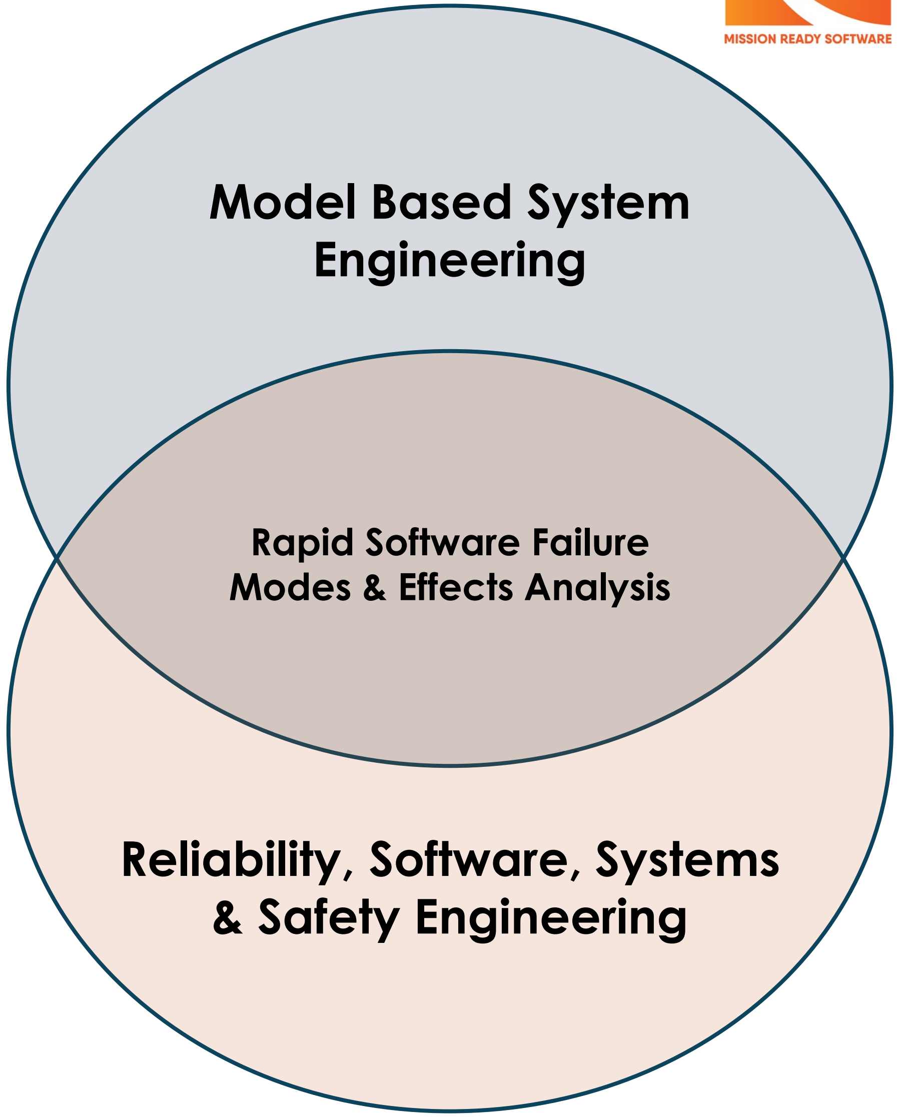
About the speakers and agenda

- Presenters:
  - Matthew McCarrell is a Model-Based Systems Engineer (MBSE) at *INTUITIVE* within their MBSE Team
  - Ann Marie Neufelder is the Founder of Mission Ready Software and an expert in Software FMEAs

- Agenda
  - Introduction
  - Overview
  - Background on FMEAs and MBSE
  - State Management Failure Mode Example
  - Error Handling Failure Mode Example
  - How the script works
  - Summary

Overview

- Software FMEAs are critical in reducing the likelihood of an undesired outcome. But only if they are executed before the code is complete.
- Ineffective analysis strategies have focused on the least likely failure modes and root causes and are too detailed to be completed in time to affect design.
- Effective software FMEAs start with what has historically gone wrong in the past – the common defect enumerations (CDEs).
- Some required inputs for the software FMEA development are captured within an MBSE model.
- Extracting that data in an automated fashion expedites the analysis and mitigation back into the model.
- A script can analyze MBSE model views to generate CDEs


Model Based System Engineering
Rapid Software Failure Modes & Effects Analysis
Reliability, Software, Systems & Safety Engineering

17 Common Mistakes when conducting a software FMEA

Organizational mistakes

- None of the software FMEA analysts have a background in software
- The analysis is not constructed by a cross-functional team
- Conducting the SFMEA too late (most of these failure modes are too expensive to fix once the code is written)
- **Conducting the SFMEA without the proper software deliverables such as the SRS, SDD, IRS, etc.**
- Failing to track the failure modes and/or make any corrective actions to the requirements, design, code, use case, users manual as a result of the SFMEA
- **Failing to tailor the software FMEA to the highest risk areas and most relevant failure modes**

Faulty Assumptions

- **Assumption that all failures originate in a single line of code or specification**
- **Assumption that software works**
- **Assumption that software specifications are correct and complete**
- Assumption that all failure modes will be found and fixed in testing
- Assumption that all failure modes are impossible or negligible in severity

MBSE/FMEA integration addresses 9 of the 17 mistakes people make when conducting a software FMEA (bolded items)

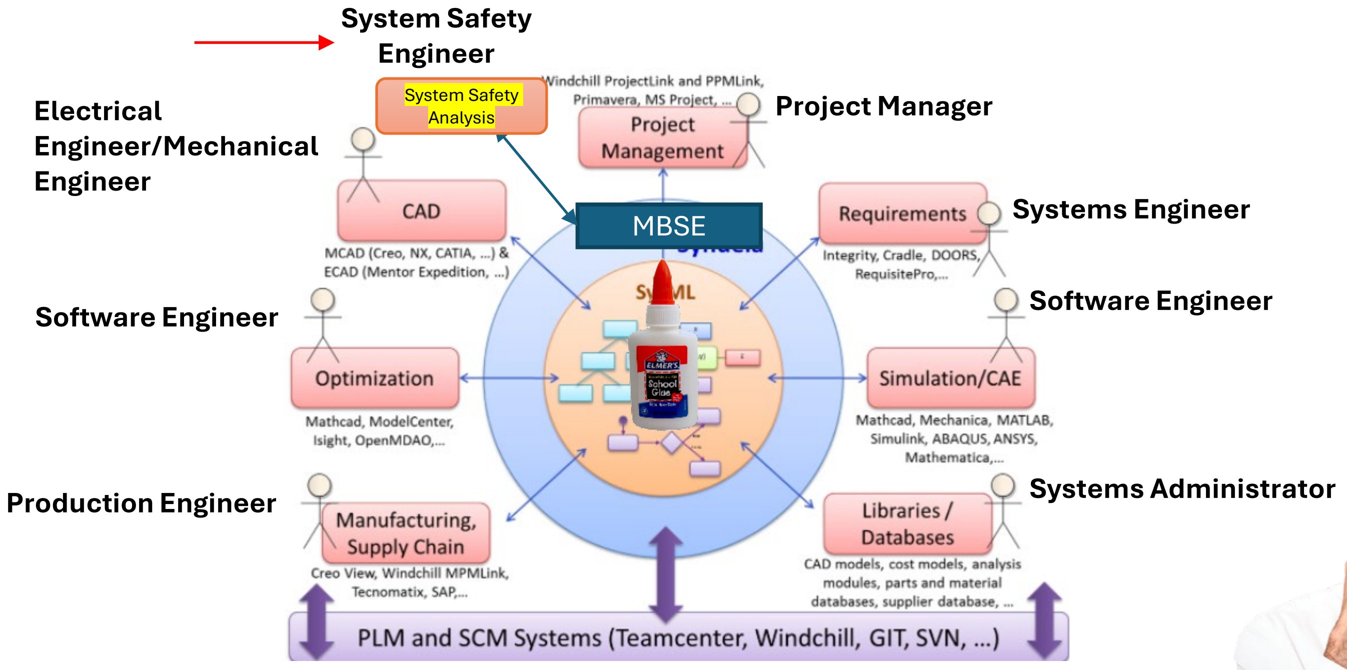
FMEA Execution mistakes

- **Focusing on total failure of the software - failing to consider small things that lead to big things going wrong**
- **Black box versus functional approach – analyze what the software does and not what it is**
- Ignoring the 6 dimensions that lead to software failures - the system, the users who use the system, the battlefield environment, and the mission
- **Conducting the SFMEA at too high (system requirements) or too low (lines of code) a level or architecture**
- **Mixing functional failure modes with process failure modes (i.e. fault timing means the software design not the software schedule)**
- Incorrectly assigning a failure rate or likelihood

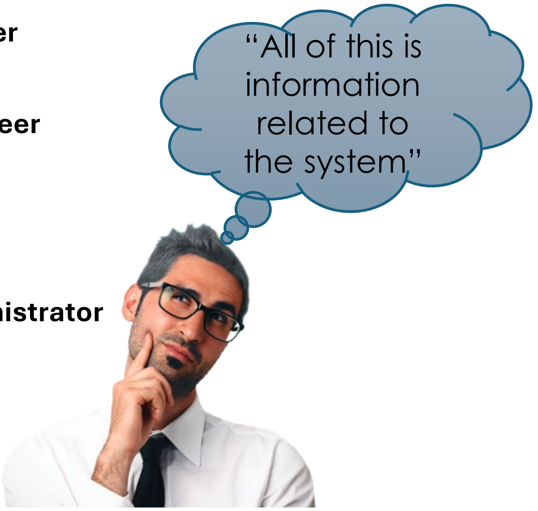
What is MBSE?

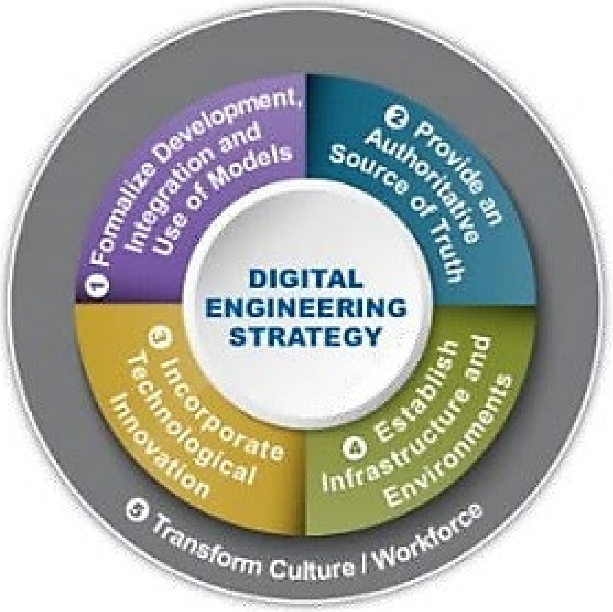
Digital Engineering (DE) is the use of digital artifacts, digital environment, and digital tools in the performance of engineering functions.

Model-Based Engineering (MBE) is an approach to engineering that uses models as an integral part of the technical baseline that includes the requirements, analysis, design, and verification of a capability, system or product throughout the acquisition lifecycle.



System Safety Engineer
System Safety Analysis
Windchill ProjectLink and PPMLink, Primavera, MS Project, ...
Project Management
Project Manager
Electrical Engineer/Mechanical Engineer
CAD
MCAD (Creo, NX, CATIA, ...) & ECAD (Mentor Expedition, ...)
MBSE
SysML
Requirements
Integrity, Cradle, DOORS, RequisitePro,...
Systems Engineer
Software Engineer
Optimization
Mathcad, ModelCenter, Isight, OpenMDAO,...
Software Engineer
Simulation/CAE
Mathcad, Mechanica, MATLAB, Simulink, ABAQUS, ANSYS, Mathematica,...
Production Engineer
Manufacturing, Supply Chain
Creo View, Windchill MPMLink, Tecnomatix, SAP,...
Libraries / Databases
CAD models, cost models, analysis modules, parts and material databases, supplier database, ...
Systems Administrator
PLM and SCM Systems (Teamcenter, Windchill, GIT, SVN, ...)





DIGITAL ENGINEERING STRATEGY
1 Formalize Development, Integration and Use of Models
2 Provide an Authoritative Source of Truth
3 Incorporate Technological Innovation
4 Establish Infrastructure and Environments
5 Transform Culture / Workforce

MBSE is a critical incorporation within Digital Engineering

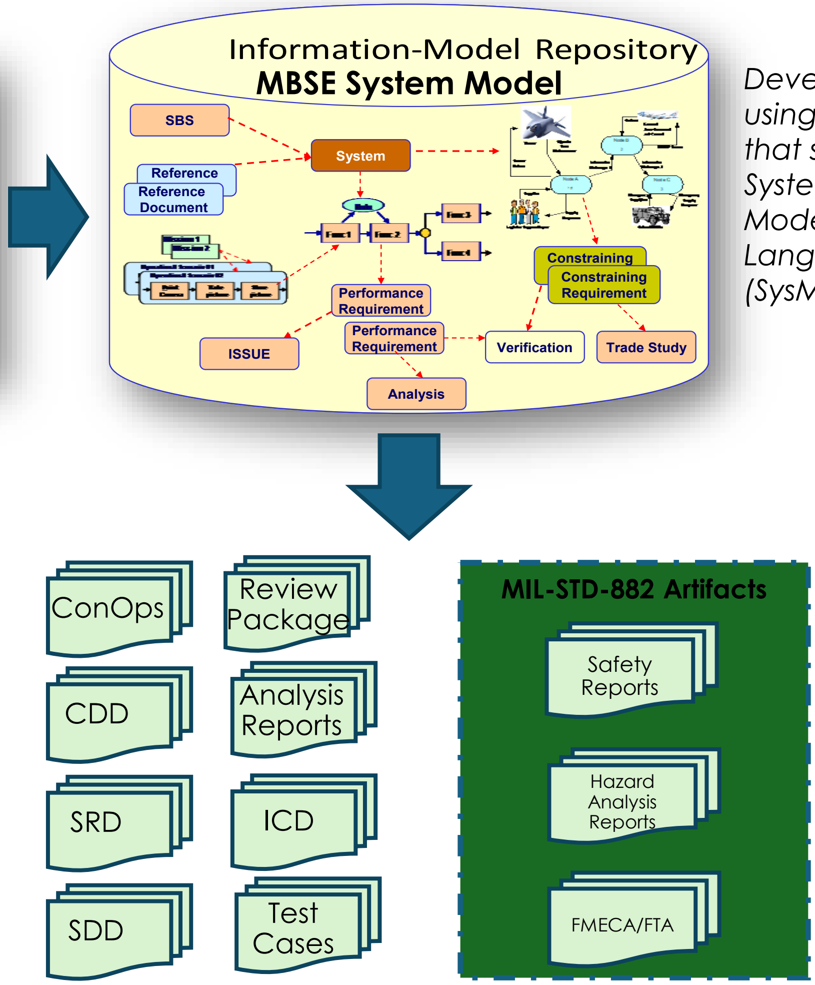
MBSE System Information Views

MBSE System Information Views

- Behavior
  - Mission and Stakeholder Requirements
  - System Functional Architecture
  - System States and Modes
  - Functional Allocations
- Structure
  - System Logical and Physical Architectures
  - External and Internal Interfaces and Definitions
- Requirement
  - System Requirements Definition
  - System Traceability Matrices
  - Verification Requirements
- Parametric
  - Trade studies
  - Performance and design analysis calculations

DIGITAL ENGINEERING STRATEGY
② Provide an Authoritative Source of Truth
TAL
ERING

The MBSE System Model is the Authoritative Source of Truth.



Developed using tools that support Systems Modeling Language (SysML)

Generating CDEs from Model Views

Failure Modes with associated diagrams

State Management (SM) – State Machine Diagram (STM)

Error Handling (EH) – Generic Table or Block Definition Diagram (BDD)

Timing (T) – Sequence (SE) or State Machine Diagrams (STM)

Sequence (SE) – Sequence (SD) or Activity Diagrams (ACT)

Data Definition (DD) – Generic Table or Block Definition Diagram (BDD)

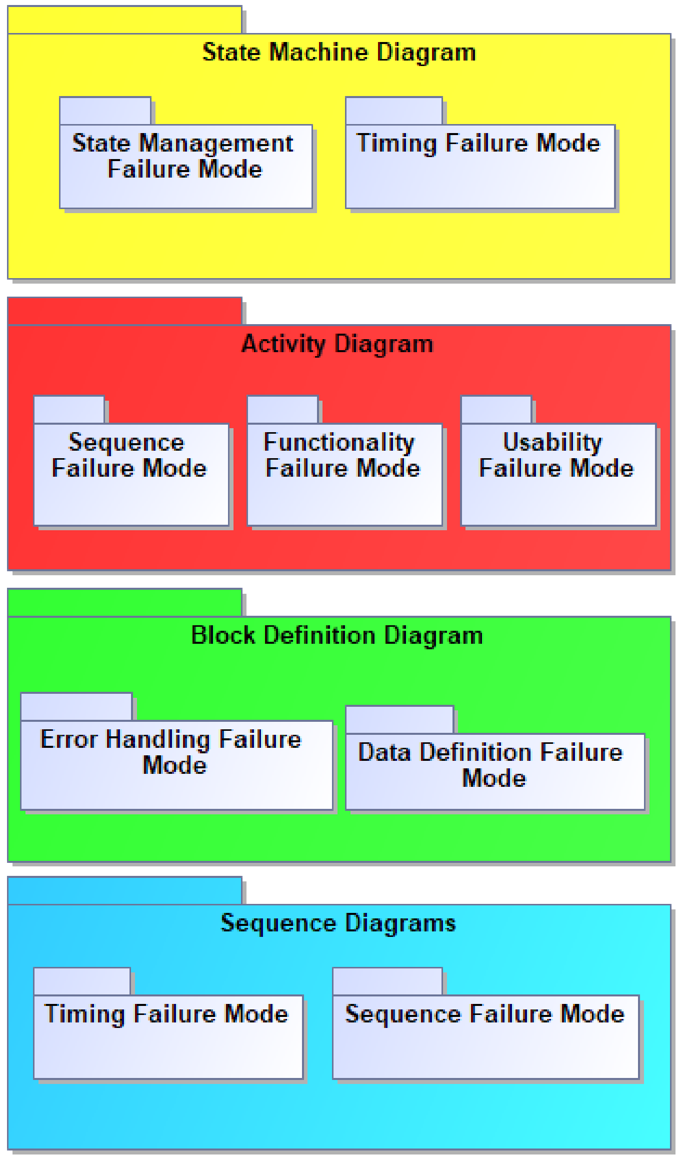
Functionality (F) – Generic Table or Activity Diagrams (ACT)

Usability (U) – Generic Table or Activity Diagrams (ACT)

Processing (PR) – Not relevant for MBSE

Algorithm (A) – Not relevant for MBSE

Machine Learning (ML) – Not relevant for MBSE



INTUITIVE®

MISSION READY SOFTWARE

Example: A traffic light

MISSION READY SOFTWARE

	Red	Amber	Green
Red	red_timer_expires = false	Prohibited	red_timer_expires = true
Amber	Amber_timer_expires = true	Amber_timer_expires = false	Prohibited
Green	Prohibited	Green_timer_expires = true	Green_timer_expires = false

These CDEs are auto-added to the Excel FMEA worksheet by the script:

Failure Mode Key:

- TL-SM-1 Prohibited transition is allowed
- TL-SM-2 Conditionally prohibited transition allowed
- TL-SM-3 Stuck state (fails to transition)
- TL-SM-4 Loss of power while in a state
- TL-SM-5 User abort while in this state

Prohibited transitions, stuck states, loss of power and user abort are rarely tested. The code can do any of these regardless of what the state design says. The effects of a prohibited transition can be different so each must be added to a different row on the SFMEA as shown next.

State Management Failure Mode Generation

INTUITIVE®

<u>Overall Methodology for the SM Failure Mode</u>

<u>Skip Step 1 and Step 2 if they are already in the model</u>

Step 1: Incorporate Requirements into an MBSE Model

Step 2: Develop a State Machine Diagram with an MBSE Model

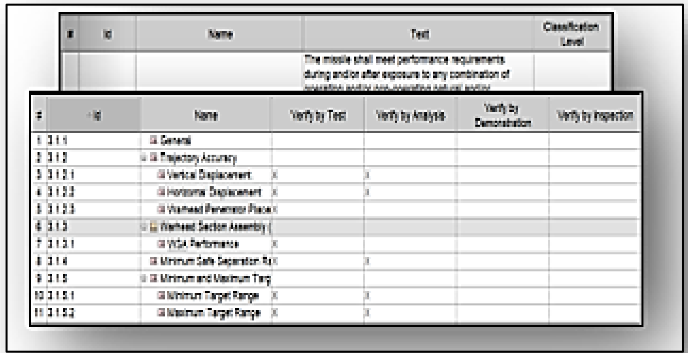
Step: 3 Generate a State Machine Table in an MBSE Model

- Output to Excel

Step 4: Autogenerate SFMEA from SFMEA Tool using script and an Excel output from MBSE Model

<u>Proof of Concept Example</u>

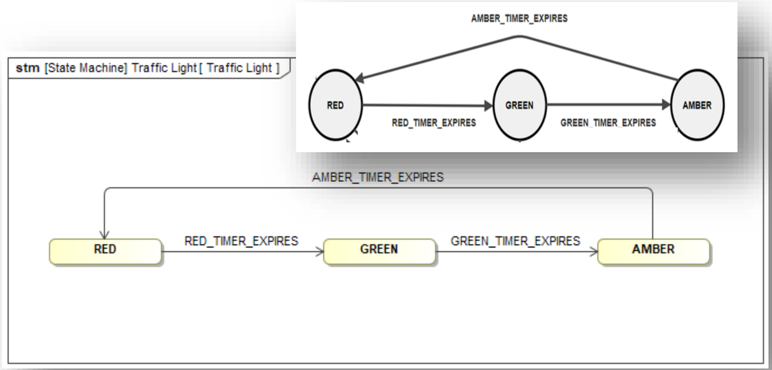
Step 1: Incorporate Requirements



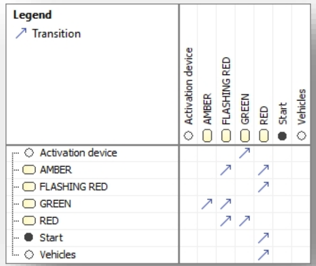
#	Id	Name	Text	Classification Level
			The missile shall meet performance requirements during and/or after exposure to any combination of [illegible]	

#	Id	Name	Verify by Test	Verify by Analysis	Verify by Demonstration	Verify by Inspection
1	3.1.1	General				
2	3.1.2	Trajectory Accuracy				
3	3.1.2.1	Vertical Displacement	X	X		
4	3.1.2.2	Horizontal Displacement	X	X		
5	3.1.2.3	Warhead Penetrator Place	X			
6	3.1.3	Warhead Section Assembly (				
7	3.1.3.1	WSA Performance	X			
8	3.1.4	Minimum Safe Separation Ra	X	X		
9	3.1.5	Minimum and Maximum Targ				
10	3.1.5.1	Minimum Target Range	X	X		
11	3.1.5.2	Maximum Target Range	X	X		

Step 2: Build a State Machine

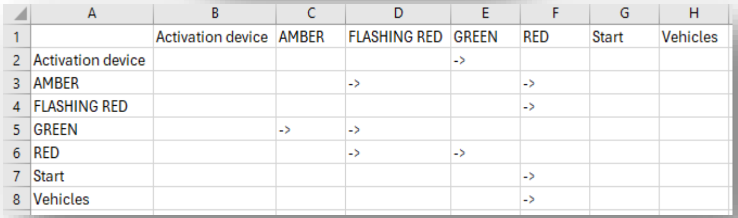


Step 3: Generate State Machine Table



Legend
↗ Transition

	Activation device	AMBER	FLASHING RED	GREEN	RED	Start	Vehicles
Activation device				↗			
AMBER			↗		↗		
FLASHING RED					↗		
GREEN		↗	↗				
RED			↗	↗			
Start					↗		
Vehicles					↗		



	A	B	C	D	E	F	G	H
1		Activation device	AMBER	FLASHING RED	GREEN	RED	Start	Vehicles
2	Activation device				->			
3	AMBER			->		->		
4	FLASHING RED					->		
5	GREEN		->	->				
6	RED			->	->			
7	Start					->		
8	Vehicles					->		

From the list of states and transitions, these rows are automatically added to the FMEA table from the script

MISSION READY SOFTWARE

TL-SM-1 – Prohibited transition allowed – Each of these is placed on its own row in the SFMEA table

- TL-SM-1 Red to amber transition allowed
- TL-SM-1 Amber to green transition allowed
- TL-SM-1 Green to red transition allowed

TL-SM-2 – Conditionally prohibited transitions are allowed – Each of these is placed on its own row in the SFMEA table

- TL-SM-2 Red to green transition allowed when the red_timer_expires = False
- TL-SM-2 Red to amber transition allowed when the amber_timer_expires = False
- TL-SM-2 Green to amber transition allowed when the green_timer_expires = False

TL-SM-3 – Stuck state – Transition not made when conditions are met

- TL-SM-3 Stuck in red when the red_timer_expires = true
- TL-SM-3 Stuck in amber when the amber_timer_expires = true
- TL-SM-3 Stuck in green when the green_timer_expires = true

TL-SM-4 – Software in an unknown state after loss of power while in a state

- TL-SM-4 Software is in an unknown state when there is a loss of power while in red state
- TL-SM-4 Software is in an unknown state when there is a loss of power while in amber state
- TL-SM-4 Software is in an unknown state when there is a loss of power while in green state

TL-SM-5 – Software in unknown state after user abort – *Note that there is no user abort, but the user will identify that when they do the analysis.*

- TL-SM-5 Software is in an unknown state when aborted while in red state
- TL-SM-5 Software is in an unknown state when aborted while in amber state
- TL-SM-5 Software is in an unknown state when aborted while in green state

From the list of states and transitions, these rows are automatically added to the FMEA table from the script

MISSION READY SOFTWARE

- The multifunctional team now has the first 3 columns of the FMEA filled out
- They work from left to right to analyze the effects, controls, likelihood, RPN, compensating provisions and recommended corrective actions
- Likelihood is assessed based on objective evidence
  - Existence – Does the specification specify the behavior if this failure mode and root cause happens? If yes, likelihood is low, otherwise high.
  - Manifestation – Is this a single or multiple point failure? Single point failure – high, multiple point failure - low
  - Control – How many independent controls that don't require user intervention? 2 - Low, 1 Medium, 0 - High
  - Detectability – Is there a written test procedure for this specific failure mode and root cause? Yes – Low, No – High
  - Average likelihood is average of above 4 scores converted to a numerical ranking
  - RPN = severity x average likelihood
  - Compensating provisions – Things that users might do to prevent the hazard
  - Recommended corrective actions – Adding specifications, changing specifications, changing the design, modifying the code for the new design or specification, writing test procedures for this failure mode and root

CDE	Failure Mode	Root Cause	Local effect	System effect	Recommended Severity	Preventive measures	Existence Likelihood	Manifestation Likelihood	Control likelihood	Detectability likelihood	Average Likelihood	RPN	Compensation provisions	Recommended corrective actions
TL-SM-1	Prohibited transition is allowed	TL-SM-1 Green to red transition allowed	What does the software do?	What does the system do?	See the PHA or FDSC for severity level	Unless there are design controls to prevent the a transition between two prohibited states, none.	Does the specification explicitly say that the transition is prohibited? If explicit then medium if implicit then high	Single point – High Double point – Medium Triple point - Low	No controls – high 1 control – medium 2 controls - low	There is a written test procedure for this specific CDE – Low Otherwise - High	Average of these 4 assessments converted to a number with same range as severity		Can users do anything to prevent hazard?	Changing or adding specifications, design, code and test procedures to address this root cause

Example: A traffic light components

Hardware components include the following, which are usually on a hardware schematic, system architecture diagram and/or parts list

- Light
  - Traffic light housing
  - Red light bulb
  - Green light bulb
  - Amber light bulb
- Equipment cabinet at intersection
  - Controller
  - Power supply
  - Phase selector controller
  - Detector amp
  - Battery
- Inductive loop vehicle sensor

Software components include the following which are usually identified on the system architecture diagram, software architecture diagram

- Controller software
- Vehicle sensor software
- ASCT – Adaptive Systems Control (machine learning software remotely located). Optimizes traffic flow.

Error Handling Failure Mode Generation

INTUITIVE®

Proof of Concept Example

Overall Methodology for the EH Failure Mode

Step 1: Incorporate Hardware parts into an MBSE Model

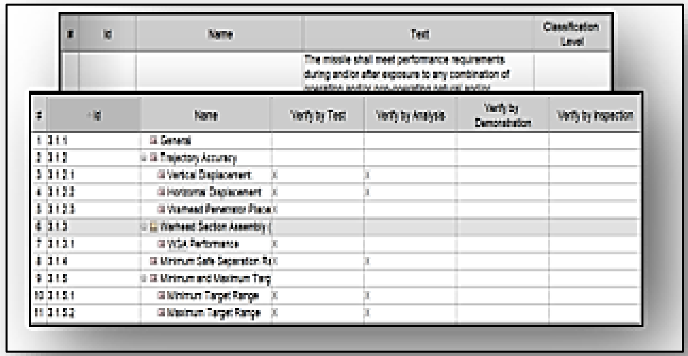
Step 2: Develop a Block Definition Diagram with an MBSE Model

Step: 3 Generate a Filled Generic Table in an MBSE Model

- Output to Excel

Step 4: Autogenerate SFMEA from SFMEA Tool using script and an Excel output from MBSE Model

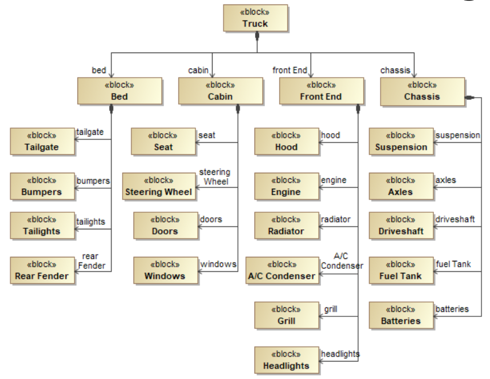
Step 1: Incorporate Hardware Parts



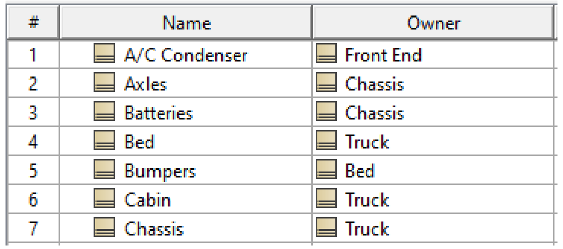
#	Id	Name	Text	Classification Level
			The missile shall meet performance requirements during and/or after exposure to any combination of operation and/or non-operation natural and/or	

#	Id	Name	Verify by Test	Verify by Analysis	Verify by Demonstration	Verify by Inspection
1	3.1.1	General				
2	3.1.2	Trajectory Accuracy				
3	3.1.2.1	Vertical Displacement	X	X		
4	3.1.2.2	Horizontal Displacement	X	X		
5	3.1.2.3	Warhead Penetrator Place	X			
6	3.1.3	Warhead Section Assembly (				
7	3.1.3.1	WSA Performance	X			
8	3.1.4	Minimum Safe Separation Ra	X	X		
9	3.1.5	Minimum and Maximum Targ				
10	3.1.5.1	Minimum Target Range	X	X		
11	3.1.5.2	Maximum Target Range	X	X		

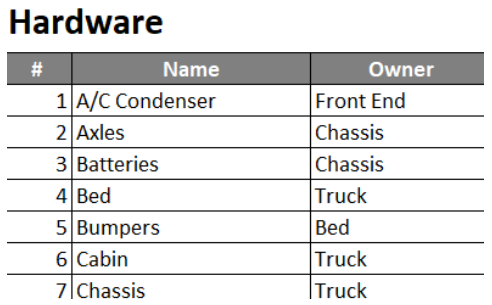
Step 2: Build a Block Definition Diagram



Step 3: Generate Filled Generic Table



#	Name	Owner
1	A/C Condenser	Front End
2	Axles	Chassis
3	Batteries	Chassis
4	Bed	Truck
5	Bumpers	Bed
6	Cabin	Truck
7	Chassis	Truck



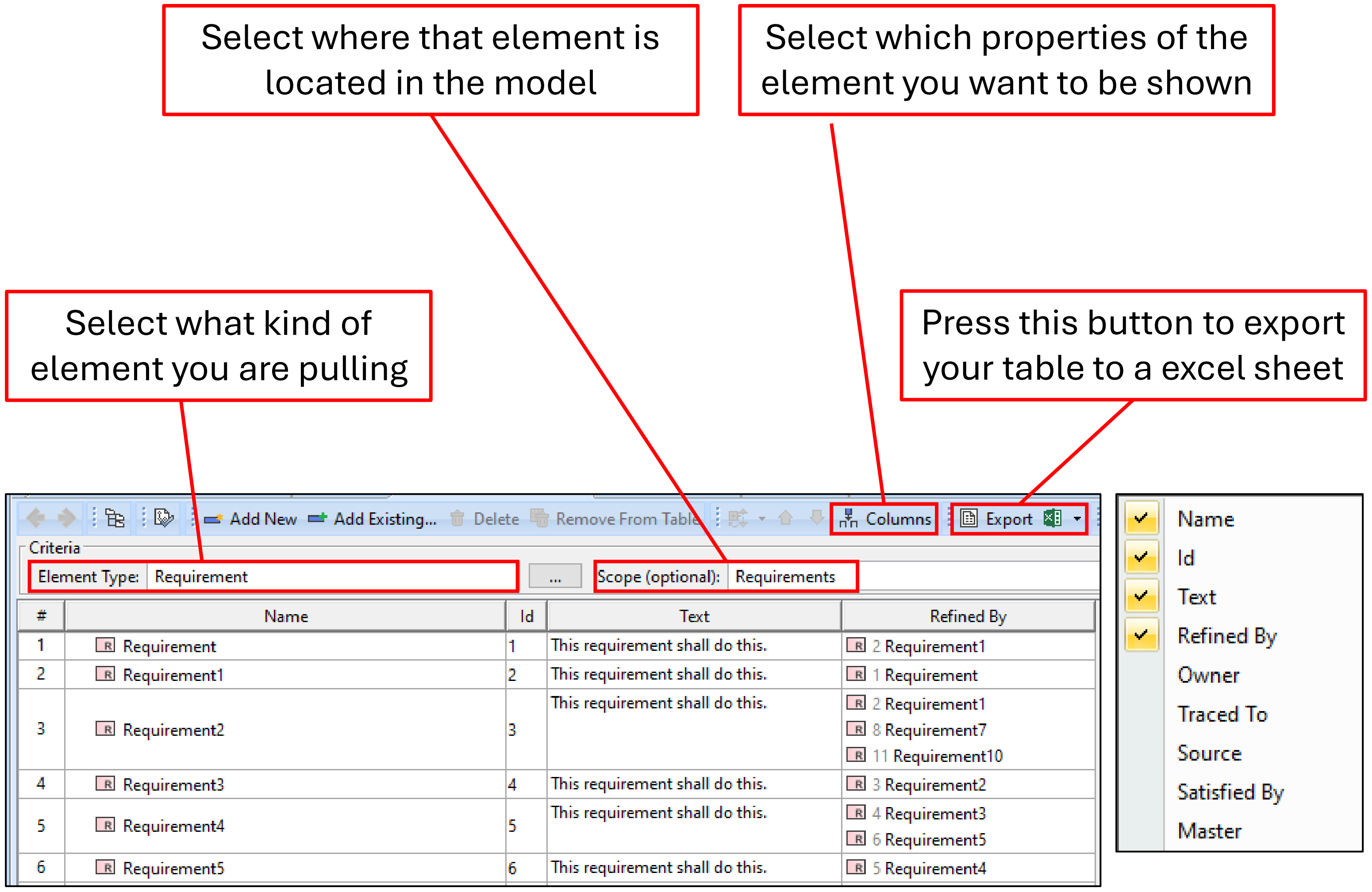
Hardware

#	Name	Owner
1	A/C Condenser	Front End
2	Axles	Chassis
3	Batteries	Chassis
4	Bed	Truck
5	Bumpers	Bed
6	Cabin	Truck
7	Chassis	Truck

System Model Accessibility

<u>What you need to know in the model</u>

- Most of the time you can skip step 1 and 2
- In step 3 you would pull the information out of the model but to do that you need a generic table
- A generic table would have 4 main components:
    - Element Type
    - Scope
    - Columns
    - Export
- Pulling Information from the model is very easy



Select where that element is located in the model

Select which properties of the element you want to be shown

Select what kind of element you are pulling

Press this button to export your table to a excel sheet

Element Type: Requirement

Scope (optional): Requirements

#	Name	Id	Text	Refined By
1	Requirement	1	This requirement shall do this.	2 Requirement1
2	Requirement1	2	This requirement shall do this.	1 Requirement
3	Requirement2	3	This requirement shall do this.	2 Requirement1 8 Requirement7 11 Requirement10
4	Requirement3	4	This requirement shall do this.	3 Requirement2
5	Requirement4	5	This requirement shall do this.	4 Requirement3 6 Requirement5
6	Requirement5	6	This requirement shall do this.	5 Requirement4

Auto generating software FMEA rows

With the list of hardware components, software components and communication sources these CDEs are auto generated by the scripts

For each of the below CDEs, one row is created for each hardware element

- TL-EH-1 Software fails to detect faulted hardware
- TL-EH-2 Software detects faulted hardware but executes the wrong recovery
- TL-EH-22 Software fails to detect faults that have been resolved
- TL-EH-23 Software is overly sensitive to faults– one row created for every hardware fault

For each of the below CDEs, one row is created for each software component

- TL-EH-29 The software fails to detect that another software component that is not or has stopped executing
- TL-EH-30 The software fails to properly handle and recover from another software component that is not or has stopped executing

Brainstorming the rest of the software FMEA template

MISSION READY SOFTWARE

- The basic framework of the FMEA is now laid out
- A group of subject matter experts now completes the effects of these failure modes and root causes
    - The effects can be mission, safety, both or none
- They assume that the failure mode and root cause exist, and they identify any controls for that failure mode
    - If it is not controlled it will happen
- They update the test procedures to test for any failure mode/root causes
    - This increases the detectability and lowers the risk
- The scripts are designed not to overwrite any analysis that is done in between revisions of the state model

Summary

- Using MBSE to export out state design, and hardware and software component names ensures that the software engineers, system engineers, reliability and safety engineers focus on the failure modes that are most likely to happen
- It also ensures that they avoid several of the 17 mistakes related to software FMEAs
- The first 3 columns of the software FMEA are the columns that virtually all analysts don't do correctly. The MBSE/scripts ensure that the FMEA gets off on the right footing
- A multi-functional team is necessary to assess the effects once the framework is laid out

CDE	Failure Mode	Root Cause	Local effect	System effect	Recommended Severity	Preventive measures	Existence Likelihood	Manifestation Likelihood	Control likelihood	Detectability likelihood	Average Likelihood	RPN	Compensation provisions	Corrective actions made
TL-SM-1	**Prohibited transition is allowed**	**TL-SM-1 Green to red transition allowed**	The wrong light is lit	Drivers in cross section are at risk of collision	10	None	Guaranteed - 10	Single point failure - 10	Uncontrolled - 10	No test procedure for this failure mode - 10	10	100	The drivers in the cross street can see them in the middle of the intersection	Rework design to check for previous light state before accepting a state transition to the next state

INTUITIVE®

Wrap Up and Discussion

- Questions?
- Comments?
- Contact: Matthew McCarrell

  Matthew.mccarrell@irtc-hq.com

  Ann Marie Neufelder

  ann.Neufelder@missionreadysoftware.com

MISSION READY SOFTWARE

About Mission Ready Software

- Collecting actual software reliability data since 1993
  - Vehicle
  - Defense
  - Medical
  - Energy
  - Space/Aerospace
  - Semiconductor
  - Major electronics
- Developer of the only machine learning model that predicts software failures and failure mode likelihood before the code is written
- World’s largest software reliability benchmarking study
- World’s largest database of software failures
  - Almost 1 million software failures analyzed for root cause