

Independent Validation of Software Safety Requirements for System of Systems

by

S. Driskell, J. Murphy, J.B. Michael, M. Shing

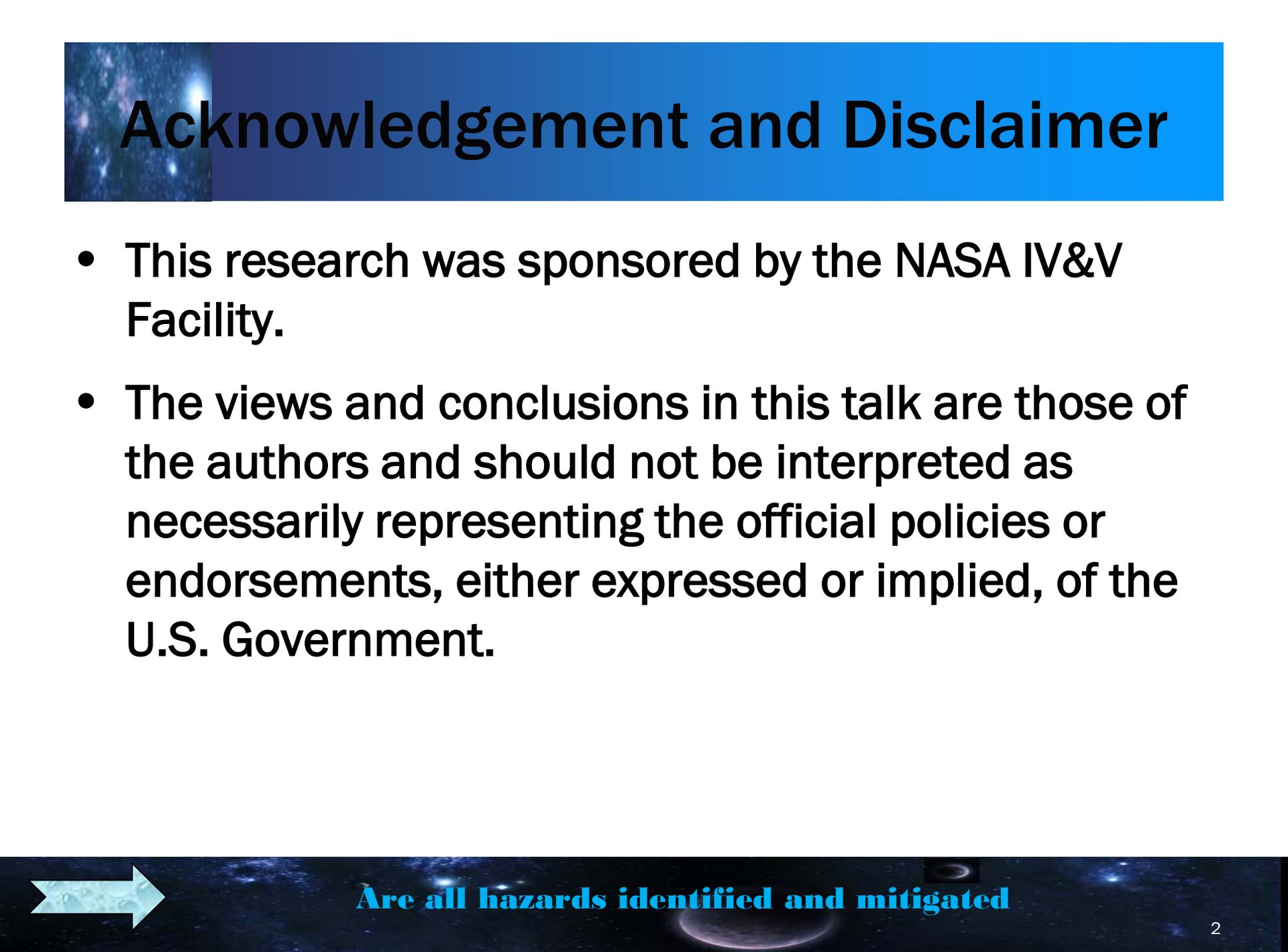
Presented by

Stephen Driskell

Stephen.Driskell@TASC.com

Judy Murphy

jmurphy@mpl.com



Acknowledgement and Disclaimer

- This research was sponsored by the NASA IV&V Facility.
- The views and conclusions in this talk are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.



Are all hazards identified and mitigated



Overview

- Address the need for IV&V to assess the quality of the software safety engineering early in the development of a System of Systems (SoS)
- Provides a proactive approach to the independent validation of safety requirements for systems of systems
- Approach
 - Develop SRMs to capture their own understanding of the safety requirements
 - Use the SRMs to evaluate the project's hazard identification and hazard analysis effort for sufficiency and completeness of the safety requirements
 - Examine requirements trace between identified safety critical failures, fault management requirements, and system, subsystem and components design



Are all hazards identified and mitigated



Who, What, Why

- The mission of NASA's IV&V program, under the auspices of the NASA Office of Safety and Mission Assurance (OSMA), is to provide the highest achievable levels of assurance for mission- and safety-critical software. The NASA IV&V Program provides assurance to our stakeholders and customers that NASA's mission-critical software will operate dependably and safely
- The NASA IV&V Program conducted a safety case study for spacecraft safe hold. Safe hold is the autonomous software for managing spacecraft hazards, without ground intervention
- Mission success and spacecraft safety are both improved through contingency hazard management and the resulting failure risk reduction



Dependability & Safety Qualities

- **IV&V dependability analysis tasks include the following assessments**
 - **Q1: Will do what it is supposed to do**
 - Availability, reliability, security
 - **Q2: Not do what it is not supposed to do**
 - Safety, security (test validation, verification)
 - **Q3: Will perform as expected under adverse conditions**
 - Availability, performance, safety, maintainability, security



Dependability quality factors are mapped to the 3Qs



Study Objectives

- **Build a dependability and safety case for safehold**
 - Does the autonomous action comprehensively manage the loss of spacecraft or mission hazards ensuring safety?
 - Are all subsystem faults requiring safe hold included in safe hold monitor?
 - Ensure hazards are managed and failure risk is reduced
- **Assess spacecraft faults and fault management action to ensure spacecraft safety**
 - Address the IV&V “3 Questions”
 - Sufficiently and adequately mitigate the potential hazards posed by a SoS
 - Identify missing safe hold requirements
- **Deliver a reusable standardized spacecraft software safety case for Independent Verification & Validation (IV&V)**



Build a dependability & safety case for SoS testing

Safety Engineering Process

- Starts with the system safety engineering activities to identify potential hazards and safety-critical functions, which are then traced through design into safety-critical hardware and software functions.
- Ends with validation and verification (V&V) of derived software safety requirements for controlling the hazard causal factors
- Team of software engineers, who are not the members of the development team, are tasked to validate and verify the SoS's software and requirements

 **Build a SoS safety case for critical functionality managing hazards**

Industry Software Safety Validation

- It is becoming a standard practice in system safety to require the developer to provide the certifier or regulator with a safety case
 - Contains well-documented evidence to provide “a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context”
- Cruickshank *et al.* presented a framework for gauging the sufficiency of the software safety requirements
 - Thoroughness of hazard identification
 - Thoroughness of hazard analysis leading to software safety requirements
 - Completeness of traceability from hazards to requirements
- Instead of relying on final testing to reveal any validity issues with the software safety requirements, application of the framework helps to identify potential problems early on in the development lifecycle

 Use Metrics to Gauge Sufficiency of SW Safety Requirements

IV&V System Reference Model (SRM)

- **Includes sets of Modeling Artifacts**
 - Use cases
 - Activity Diagrams
 - Sequence Diagrams
 - Statecharts
 - Domain Models (Class Diagrams, Communication Diagrams)
- **Test cases developed by IV&V analysts**
 - Independent Test Capability (ITC) provides infrastructure to run the tests
 - IV&V analysts execute tests
- **SRM is a *concise* description of the IV&V team's understanding of the problem**
 - Analysis tool
 - Communication tool
- **Captures expected system behaviors**
 - 3 Questions



Capture IV&V team's understanding of system behaviors

IV&V System Reference Model (SRM) (cont'd)

- **What is the IV&V fault conditions independent list and how is it used?**
 - Fault conditions list developed by the IV&V
 - Based on past mission experience
 - Living artifact
- **Two approaches to use the fault conditions list**
 - Take the fault conditions list and compare it to what you already know about your Fault Management (FM) and Failure Modes and Effects Analysis (FMEA)
 - If you do not know where to start, look at the fault conditions list and apply it to your mission to check for conditions and functions such as over/under temperature, over/under voltage, command issues, telemetry monitors, etc.
- **Identify gaps in your mission's FM and FMEA**
- **Identify gaps in the IV&V fault conditions independent list**



Capture IV&V team's understanding of system behaviors

Modeling Safety-Critical Behaviors

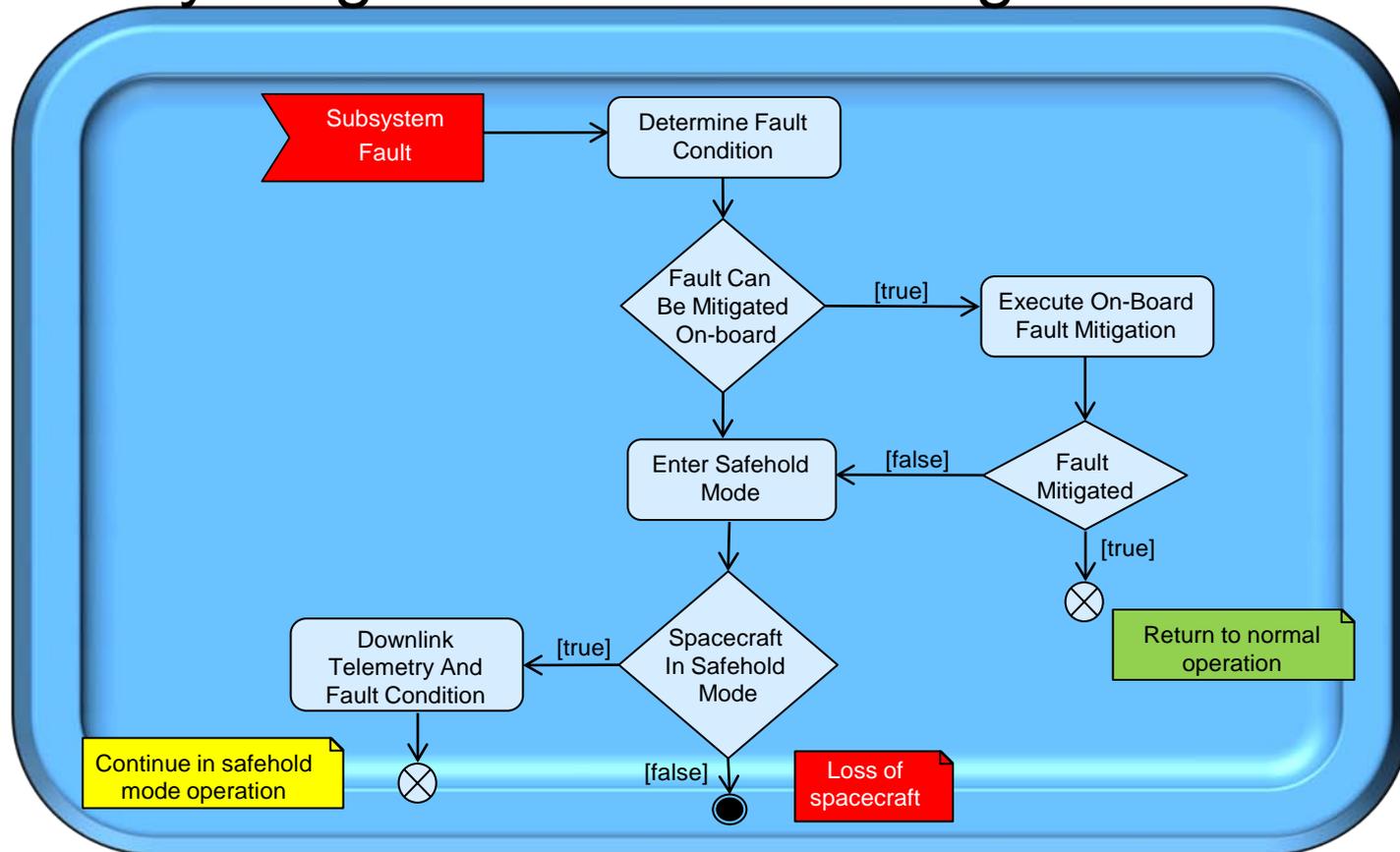
- **Develop high-level use cases and use case diagrams**
 - The starting point of both understanding and documenting system behaviors
 - Useful for identifying the functionality of the system
 - Records behavioral requirements for the software
 - Use case narrative depicts step-by-step flow of the expected behavior
 - Activity diagrams provide graphical representation of a complex thought which should reflect the use case specification
- **Mapping the scenarios of the use cases to activity diagrams, sequence diagrams and statecharts helps highlight the assignment of responsibilities among the component systems of a SoS**



Build and map SoS requirements to an independent SRM

Sample SRM Artifact

- Activity diagram for fault management



Reusable high-level fault management example



Safety Case Study

NASA Mission Assessment

- **Analyze high priority behaviors**
 - Maintaining the health and safety of the spacecraft which involves execution of and response to faults
 - Checkout of the spacecraft which includes safe hold mode and autonomous operations
- **Address the following questions**
 - Does the autonomous action comprehensively manage both the loss of spacecraft and the mission hazards ensuring safety?
 - Are all subsystem faults requiring safe hold included in the safe hold monitor (a safety executive)?
 - How does the mission under study compare to the IV&V fault conditions, independent list?
 - Does the IV&V fault conditions independent list require updating?



Build a safety case for on-orbit (operational) safe hold



Safety Case Study

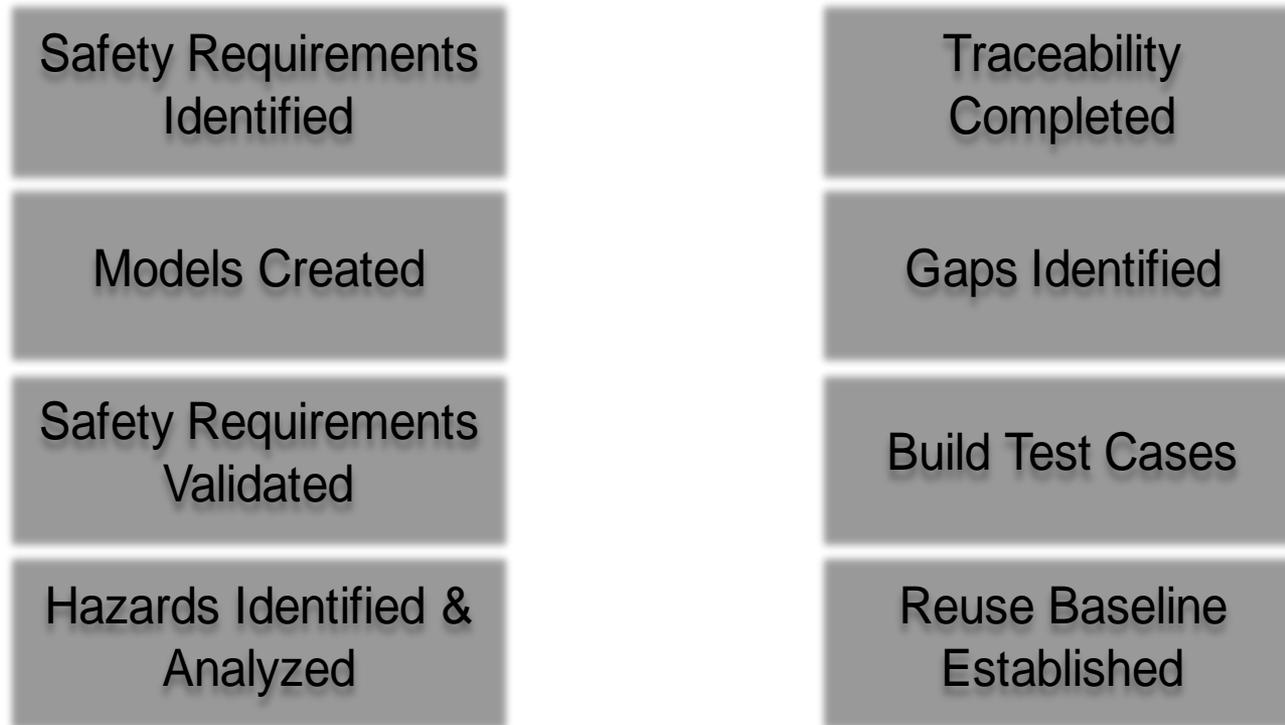
NASA Mission Assessment (cont'd)

- Ensure these hazards are managed and failure risk is reduced
- Deliver a reusable standardized spacecraft software safety case for IV&V
- Identify missing safe hold requirements
- Provide software test scenarios



Sufficiently and adequately mitigate the potential hazards posed to a SoS

V&V Software Safety Analysis Process



 **The right process identifies missing requirements**



Evaluation of the Developer's Software Safety Products

- Review the following artifacts from developer:
 - Core Performance Requirements (Level 3)
 - Core Spacecraft FM
 - Core Spacecraft FMEA
 - Core Spacecraft Flight Software (Level 4)
 - Guidance Navigation and Control System Requirements Specification (Level 4)
 - Core Observatory Command and Data Handling Subsystem Requirements (Level 4)
 - Software Safety Program Plan (SSPP)
 - Preliminary Hazard Analysis (PHA)

Level 3 = subsystem-level requirements

Level 4 = internal, all-software, requirements.



Perform an independent sufficiency & adequacy assessment



Evaluation of the Developer's Software Safety Products (cont'd)

- The IV&V team evaluated the developer-provided artifacts against the OSMA safety criteria which included the FMEA and FM provided artifacts
- **Sufficiency and Adequacy**
 - The degree to which discrepancies between the IV&V fault conditions and the project's FMEA and FM artifacts and the necessary software safety requirements to manage the safety-critical faults exist
 - Gaps are assessed
 - Sufficiency and adequacy are communicated to the developer



Perform an independent sufficiency & adequacy assessment



Evaluation of the Developer's Software Safety Products (cont'd)

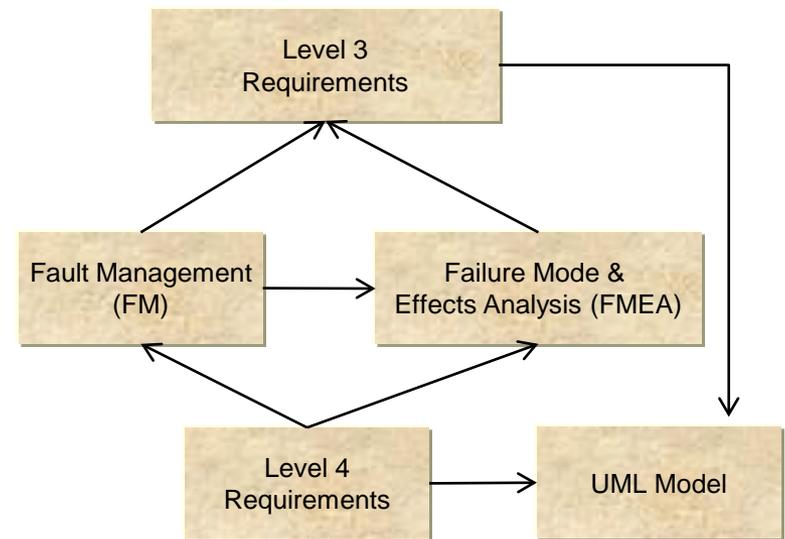
- Through executing this process it is possible to discover if the safety requirements are potentially incomplete and if there is room for improvement in FMEA and FM to eliminate gaps in the failure events and fault management



Perform an independent sufficiency & adequacy assessment

Evaluation of the Developer's Software Safety Products (cont'd)

- Dependability & safety case traceability
 - Hazard created by failure - do the failure responses satisfy all safety requirements by managing loss of Spacecraft or Mission?
 - Hazard Effect – loss of spacecraft or loss of science mission
 - Hazard Mitigation – autonomous safe hold
 - Trace top level requirement to hazard mitigation, where autonomous safe hold, when comprehensive, ensures hazard management



High-level safety case

Traceability helps determine if safety requirements are met



Where We Are Today

- **Mapped IV&V list of fault conditions to MSL Fault and Failure Analysis (FFA) data**
 - Partially successful
 - MSL FFA data is at a different level than the IV&V list of fault conditions
- **IV&V efforts on MAVEN and other science missions have decided to build safety cases using this process**
- **This approach will be applied to other behaviors besides safehold**



Execution of a reusable process and product

Where We Are Today (cont'd)

- Categorize current and future missions by mission manager and developer to establish fault condition similarities
 - Managed by GSFC
 - Glory, GPM, ICESAT-2
 - SDO → LRO → GPM
 - Managed BY JPL
 - MSL, SMAP, MAVEN,
 - JUNO, JWST, GRAIL

 **Utilize manager and developer commonalities and legacy**



Conclusion

- **Dependability & Safety Case based assessment is reusable with simple changes for architecture, subsystems, science experiments and behaviors**
 - These results can be applied to the next spacecraft SoS family
 - Contributes goodness to any System of Systems by improving Mission Safety & Dependability
- **Sufficiency and adequacy enhanced by creation of a standard template SoS safety case**
 - Safety case portfolio builds a super set of requirements which can be applied to any similar SoS as a starting point for safety
 - Safety case builds examples for specific SoS implementation



Reusable safety process identifies requirements & safety gaps



Conclusion (cont'd)

- Update the IV&V list of fault conditions
 - Apply lessons learned from MSL FFA mapping
 - Faults should be based on functionality/behavior as opposed to a specific device/card/element/acronym. For example:
 - Specific telemetry/command issues
 - » Telemetry parity error
 - Temperature and voltage issues independent of a specific device
 - Allow the developer to “*assign*” that issue(s) to a specific device



Update IV&V fault conditions independent list