

Technical Reference Suite Addressing Challenges of Providing Assurance for Fault Management Architectural Design

Presented to the 32nd Space Symposium

Date: 4/11/2016

Presenter: Rhonda Fitz (MPL Corporation)

Co-Author: Gerek Whitman (TASC, an Engility Company)

- Introduction to NASA IV&V
 - IV&V Methodology
 - IV&V Assurance Strategy
- Challenges with Fault Management
- SARP FM Architectures Encore Initiative
 - Adverse Conditions
 - Adverse Condition Database
- Technical Reference Suite
 - FM Architecture Matrix TR
 - FM Visibility Matrix TR
 - FM Assurance Strategy TR
- Conclusions

NPR 7150.2, NASA Software Engineering Requirements

The program manager shall ensure that software IV&V is performed on the following categories of projects:

- Category 1
- Category 2 that have Class A or Class B payload risk classification
- Projects specifically selected by NASA Chief of Safety and Mission Assurance

IV&V = Independent Verification and Validation [of Software]

Independence:

- Technical Independence
- Managerial Independence
- Financial Independence

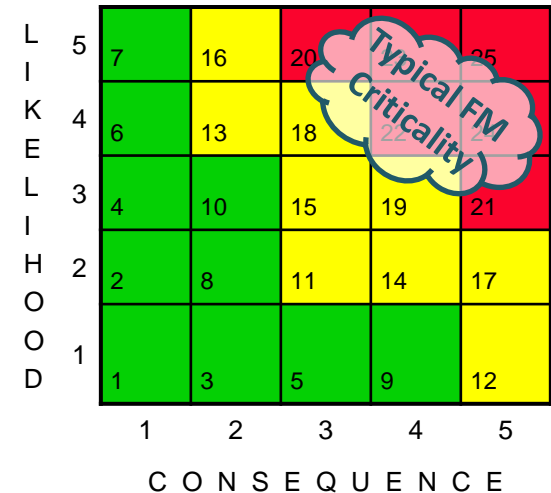
NPR 7120.5E defines Categories; NPR 8705.4 defines classification of payload risk

Criticality analysis assesses likelihood and impact of failed behaviors

- Plotted on a risk matrix
- Establish priorities and focus for analysis
- Generally, FM is high criticality

The goal of each IV&V project is to assure mission success by assuring that the critical software (mission-critical and/or safety-critical):

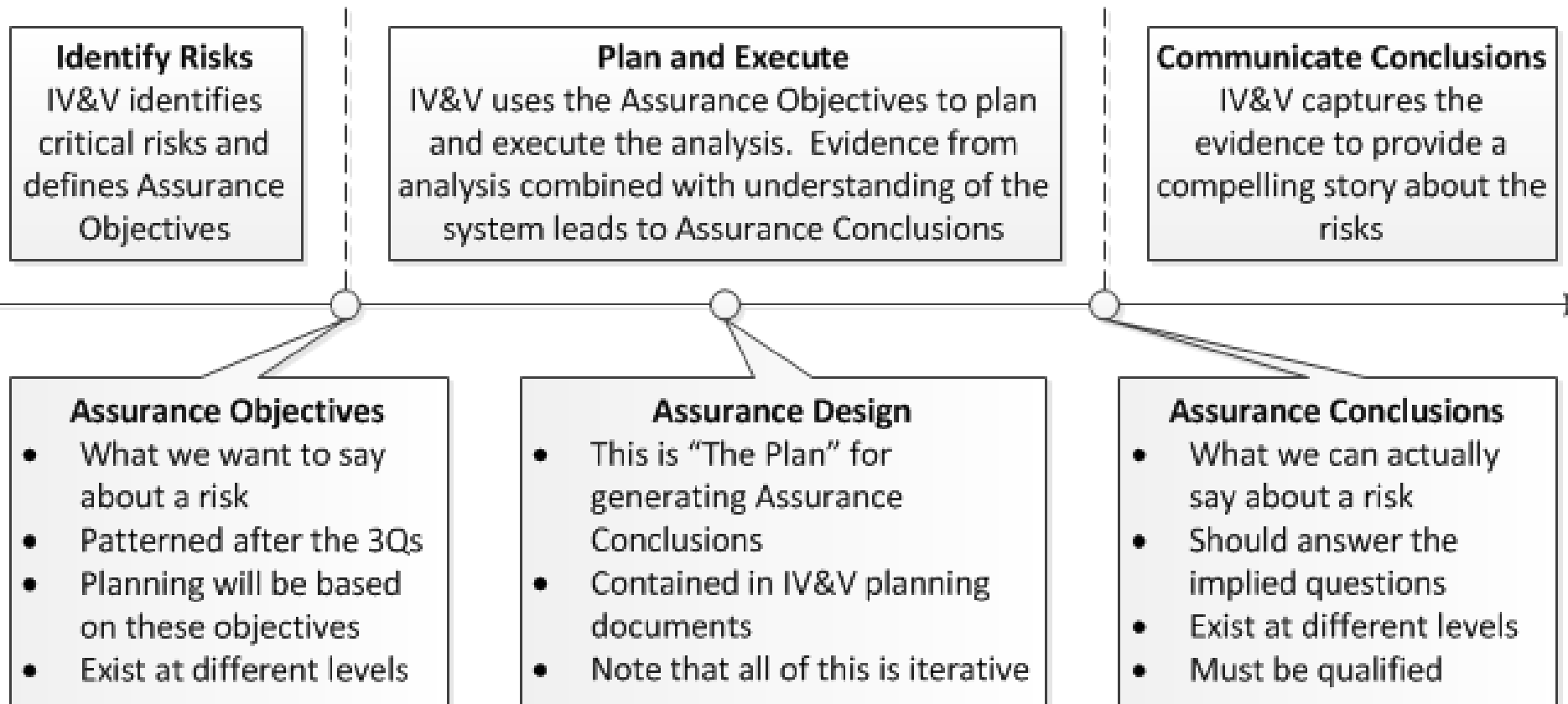
- Does what it is supposed to do
- Does not do what it is not supposed to do
- Performs appropriately under adverse conditions



a.k.a. "The 3 Qs"

IV&V assures mission success by validating and verifying critical software


IV&V Assurance Strategy



- Increasing FM complexity goes beyond traditional fault protection with the goal of not only averting catastrophe, but also maintaining capability
- FM systems, many times architected as reactive components embedded within the overall software system, must be validated against higher-level system capability requirements
- Off-nominal conditions are challenging to identify comprehensively, understand completely, and ascertain the optimal response to mitigate risk
- Existing software development and assurance practices applied to FM systems need improvement to provide a high level of assurance


Description/Goals

- Improve and expand upon the current analysis of NASA mission FM in a **Technical Reference suite** for more comprehensive coverage of architecture, visibility, and assurance strategies
- Develop and refine the prototype **Adverse Condition Database** for access to IV&V project fault, failure, and hazard data for more rigorous assurance and risk reduction with Q3 analysis
- Socialize products and findings with FM **Software Assurance Knowledge Exchange**



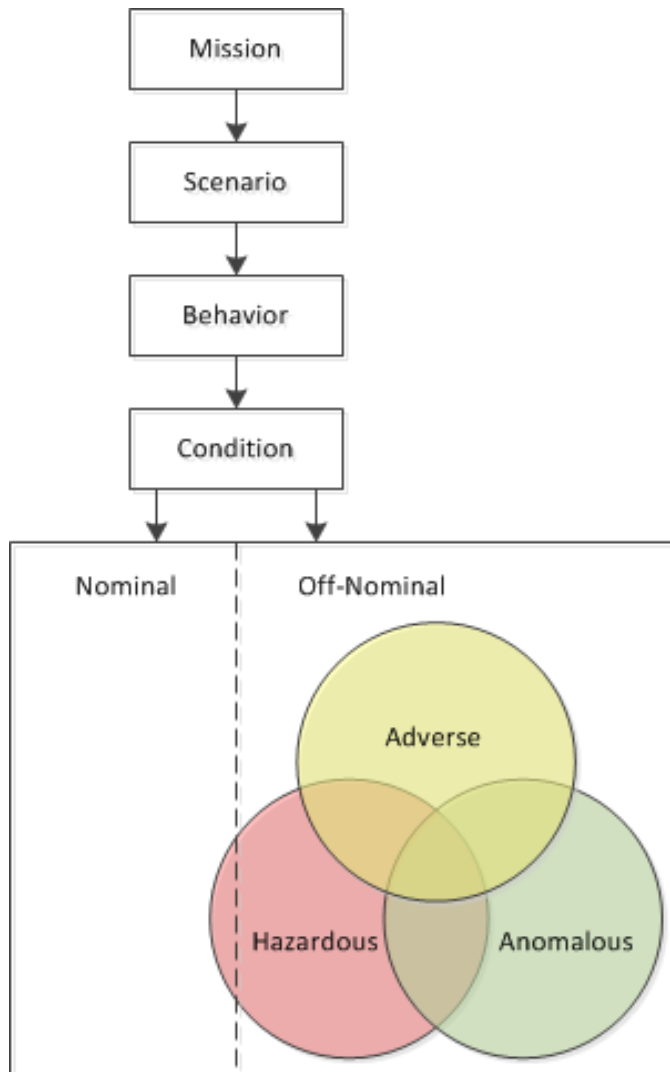
Products

- FM Architecture Matrix TR, FM Visibility Matrix TR, and dynamic FM Assurance Strategy TR with supporting IV&V methods employed across the development lifecycle
- Repository of NASA mission adverse conditions and associated project metadata
- Technical presentations, conference papers, and informal learning opportunities



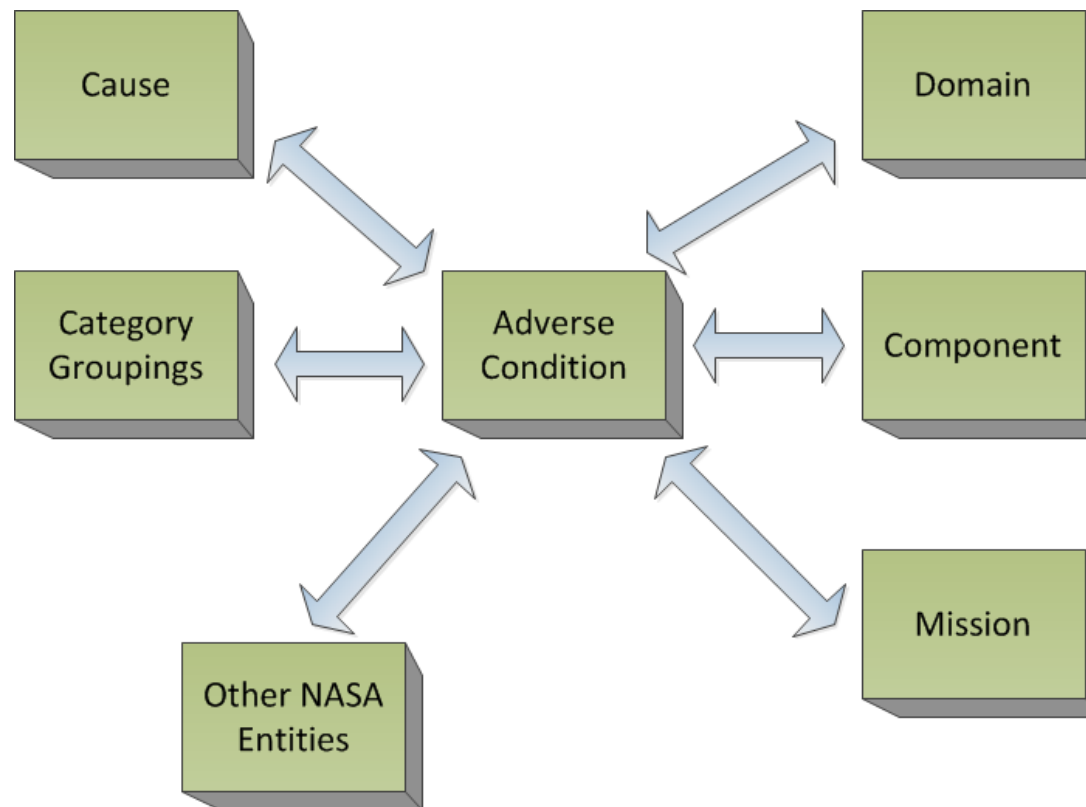
Value to NASA

- Promoting FM knowledge for IV&V Program, SARP, and NASA Engineering Network
- Improved assurance from the provision of more comprehensive data
- More rigorous Q3 analysis from identification of off-nominal scenarios
- Increased efficiency of analyst workflow and broader test coverage
- Greater focus on FM and project areas of vulnerability or high risk



- Examining Q2 and Q3 are major challenges of FM software
- Adverse Condition: A subset of an off-nominal state that prevents a return to nominal operations and compromises mission success unless an effective response to the causal fault is employed.
- How a system is architected to handle faults and adverse conditions is crucial for the satisfaction of functional and performance requirements for mission success

- Create a database that centralizes a compilation of adverse conditions and related data from NASA projects
- Architect the fields such that there may be sharing of data between projects and among the broader software assurance community for more rigorous analysis



IV&V Analyst Subject Matter Experts were surveyed from each of nine chosen projects with a variety of mission types, developers, and relative complexity

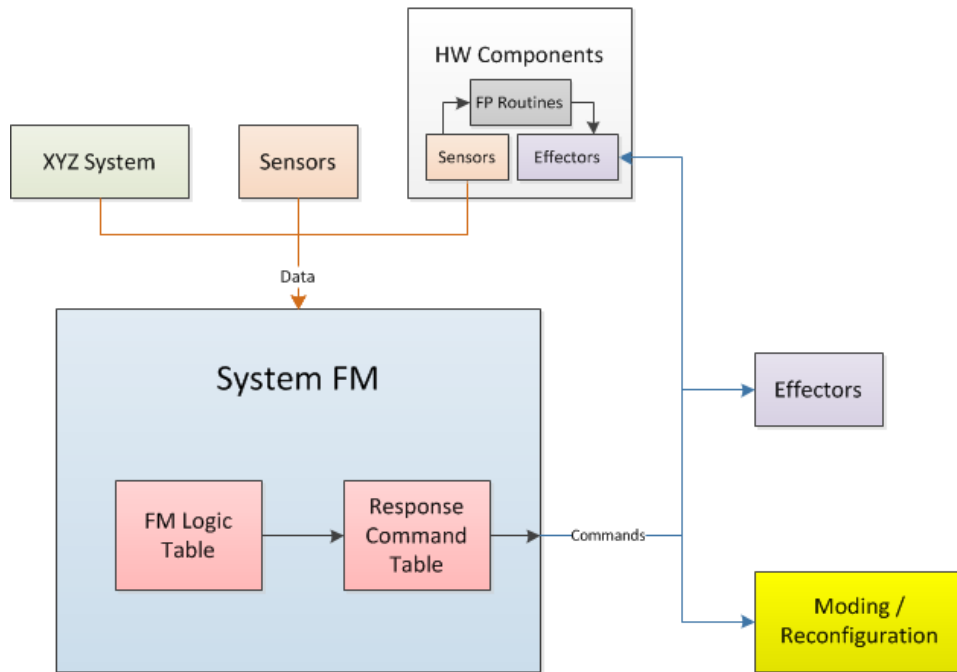
<i>Name</i>	<i>Mission Type</i>
Mars Science Laboratory (MSL)	Deep Space Robotic
International Space Station (ISS)	Manned Spaceflight
James Webb Space Telescope (JWST)	Deep Space Robotic
Multi-Purpose Crew Vehicle (MPCV)	Manned Spaceflight
Joint Polar Satellite System (JPSS)	Earth Orbiter
Magnetospheric Multiscale (MMS)	Earth Orbiter
Geostationary Operational Environmental Satellite R-Series (GOES-R)	Earth Orbiter
Solar Probe Plus (SPP)	Deep Space Robotic
Space Launch System (SLS)	Launch Vehicle

Architecture Matrix TR (excerpt)

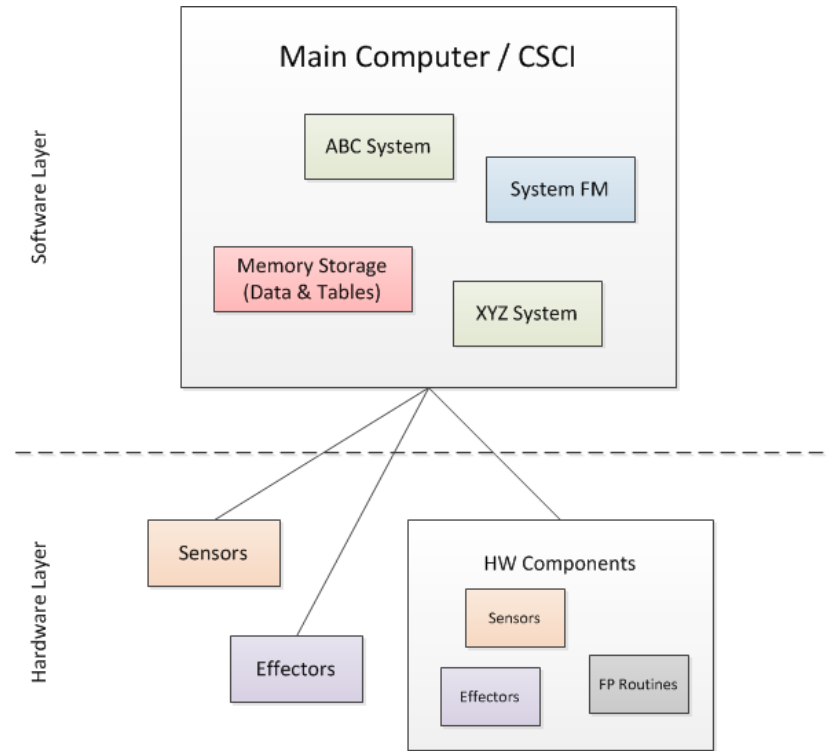
Survey Question	Cross-Mission Observations
Structure - How is it structured/organized?	
Is the FM architecture fully local? System? Hybrid? Some other organization?	A tradeoff exists between the simplicity of a centralized system level approach, and the robustness of a hybrid, tiered approach. The lower the level at which the fault can be handled, the less impact it has on the system. Earth Orbiters tend to be more centralized, Human-rated vehicles more distributed, and Deep Space falling anywhere along the scale depending on the mission parameters and developer.
How many tiers/layers are there in the FM architecture? Do these tiers/layers overlap?	Tiers are used to organize systems that are not centralized, but even the most centralized examples here still have hardware layer FM. Often there are two tiers: local and system. Sometimes FM is just primarily system level (with some additional hardware layer FP), and sometimes one or more intermediate tiers are used in between local and system, depending on the complexity of the spacecraft architecture. Usually these tiers have to overlap the same faults to allow them to be handed up from a lower tier to a higher one, but this is always done in a systematic, logical way.
Concept - What are the big design ideas?	
Is the system fully automated? Does it allow for human intervention? Is it designed with humans in the loop?	Timing often requires high autonomy, either because human reaction time is too slow, or because of communication delays. Most Earth-Orbiting and Deep Space missions are not designed around having human controllers constantly watching, and some don't even dictate regular contact, but ground ops is always given the capability to perform FM procedures. Degree of autonomy appears to correlate loosely with distance from operators (onboard or on the ground).
What was the process used to develop the FM architecture and system?	Developers tend to fall back on what they know and have experience in - heritage programs, prior life cycle processes, even ones that are of different mission domains. Human-rated missions require a slightly different approach, however, and may require a more unique process.
Implementation - How was it built, how does it work?	
At what stage of the mission life cycle was the FM system designed and built?	More and more, FM design is happening sooner, more in phase with the rest of the spacecraft systems, guided by heritage and previously-developed standardized architectures, but it still has the potential to lag behind, especially to adapt to changes in other subsystems.
How many fault monitors and unique responses does the system have?	The more requirements the FM system has for preserving functionality when something goes wrong, the more monitors and response logic it is going to need to do its job. Generally a system will have more monitors than responses, since different monitors or faults will trigger the same response.
Other Architecture-Related Questions	
Is this FM architecture inherited from another mission or based on a previously-developed standardized architecture?	All projects have some degree of inheritance, in the actual architecture and design or development process. Developers often draw from their accumulated knowledge of what does and does not work in FM architecture development.
How did the mission domain and parameters influence the design of the FM architecture?	Critical mission events and other significant mission parameters like autonomy, onboard crew, and failure tolerance are often the largest drivers for structural and functional FM architecture design.

Centralized FM Architectures

Functional Architecture

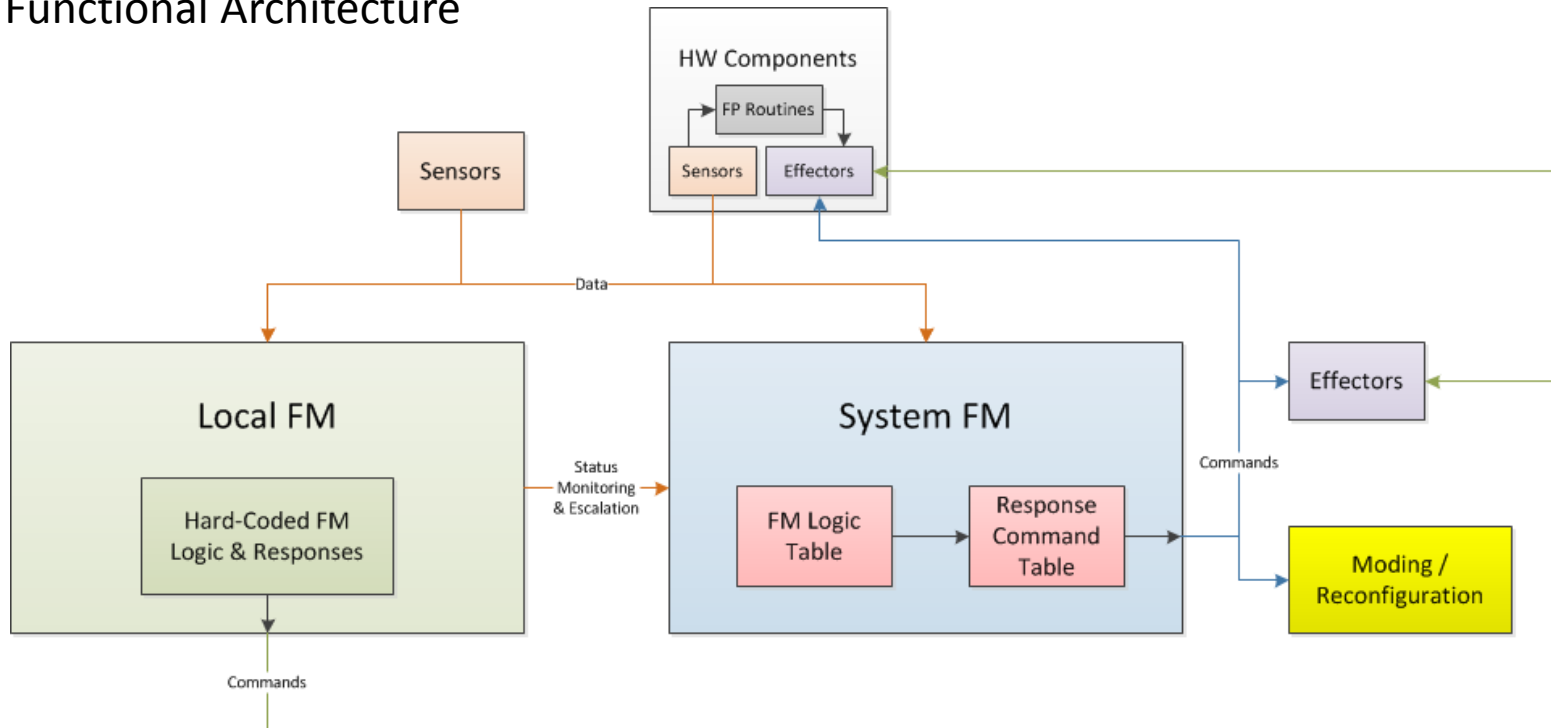


Structural Architecture



Centralized architectures are common in Earth Orbiters

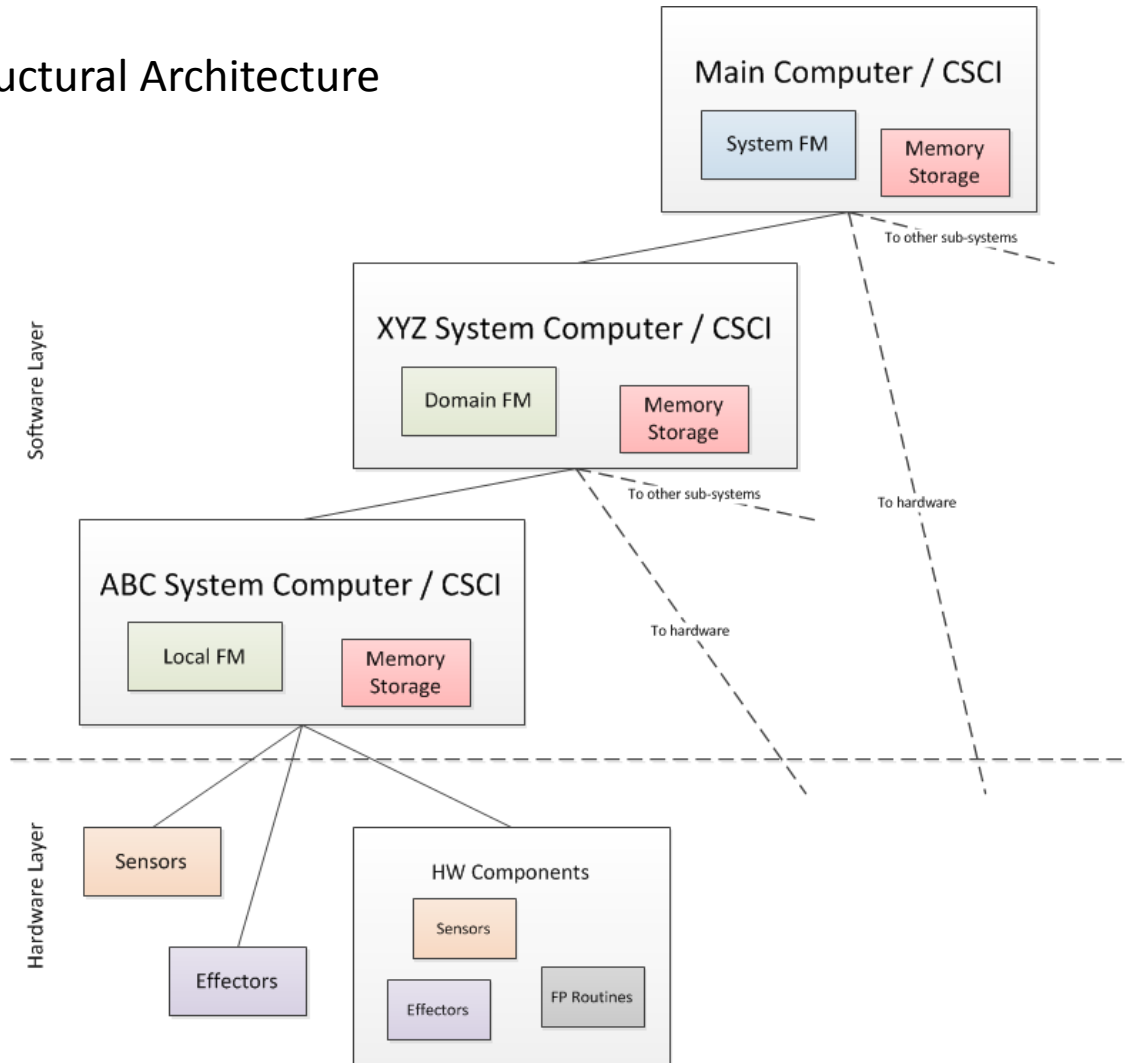
Functional Architecture



Human Spacecraft and Deep Space Robotic missions commonly use hybrid architectures

Hybrid FM Architectures (continued)

Structural Architecture



Visibility Matrix TR (excerpt)

Lifecycle Phase	Development Artifact	Architectural Visibility	Analyst Visibility
Concept	Fault Management Plan	When planned for and designed early in the lifecycle, FM architectures are generally more well-developed and documented, and therefore more visible, as opposed to architectures designed more as an afterthought.	Comprehensive knowledge of the development process may provide context for architectural decisions and thereby increase visibility. A top-down design approach may lead to higher visibility, simply due to the flow of designs and documentation.
Requirements	Functional Requirements Specifications (L5/L6), Interface Control Documents, Interface Requirements Specs	Requirements specify how software limits are employed to detect and guard against failure and recover from anomalous events and conditions. Missing and low-quality requirements or lack of traceability impede visibility.	Requirements for Fault Management Detection, Correction and Responsive behaviors are not always explicitly indicated. Requirements decomposition leads to multiple levels of abstractions. Establishing appropriate Fault Management details for each level is necessary. Requirement specifications serve as a further description of the architecture and hierarchy of the FM system, and how it is intended to operate.
Design	Physical and Functional FM Diagrams	Monitors are usually arranged and organized logically in the physical and functional system structures. A large number of dependencies can increase complexity and decrease visibility. If a system handles certain faults in different ways, it adds an additional layer of complexity that can challenge understanding.	The relationships and interfaces of FM systems are usually well-documented and understood by analysts. Instances where faults are handled differently are primarily hardware faults that trigger basic responses like redundant component swaps. These cases are usually documented in subsystem documents, but may be left out of system-level monitor/response lists because they happen on a low level, perhaps not even extending out of the hardware layer. An understanding of the physical components of the system is necessary in order to gauge whether appropriate monitors are defined.
Implementation	Source Code	More source code generally increases the complexity of the system. Factors affecting complexity include multitasking, inter-process communication, amount of auto-coding (and source of auto-coding), reuse, COTS.	Solid designs and well-written requirements enable code visibility. Language and code structures used can also impact the understandability of the software without clear supporting documentation. Complexity with items like multitasking or complex inter-module communications complicate code visibility, even when strong requirements and designs exist.
Test	Test Cases	N/A	Review of test plans may afford analysts additional insight into the types of testing that may be expected once test products are released. Single-tier testing vs. multi-tier testing, testing with simulations vs. testing with real hardware, etc.

Assurance Strategy TR (excerpt)

Typical Assurance Objectives or Conclusions	Source Mission Type	3Qs Mapping
Integration & Testing Phase		
"There are no inadvertent fatal Event Records in the code that could cause an unplanned processor reset."	Deep Space Robotic	Q2
"The Second Chance Entry, Descent, & Landing does not harm the core Entry, Descent, & Landing sequence."	Deep Space Robotic	Q2
"The analyzed fault management implementation has been proven correct and complete through verification testing."	Earth Orbiter	Q1
"All necessary fault paths were exercised in the identified validation testing."	Human Spaceflight	Q1
"The set of tests was comprehensive with regard to the Fault Management Design Document algorithms."	Human Spaceflight	Q1
"The FM data input parameters, persistence limits, CUI's, etc., were validated through appropriate testing."	Human Spaceflight	Q1
"The in-scope software will perform its intended functions for nominal [and addressed off-nominal] conditions at a higher risk level than a human-rated mission."	Human Spaceflight	Q1 Q3

- Developers' previous experience and mission heritage have a large effect on the FM architecture used, sometimes independent of the mission domain
- Analysts need to rely on their collective knowledge and experience to decide how best to build and execute an Assurance Strategy
- Planning is not always enough, however; analysts must also be prepared to adapt to visibility challenges as they appear
- The TR suite generated from this initiative builds a strong foundation to fill the existing gaps in the FM knowledge domain and is useful across the Agency and beyond
- Building a culture or community that values cross-project communication for continual improvement needs to be a priority for FM architectural design and analysis
- AC Database enhancements and investigation into how nontraditional processes (model-based FM within an Agile development) affect FM architectural design are aspects of SARP FMAE

References:

- [NASA IV&V Website](#)
- Fault Management Handbook (NASA-HDBK-1002) Draft 2
- Fault Management NASA Engineering Network
- [IV&V Technical Framework \(IVV 09-1\) Version P](#)

Contact Information:

Rhonda Fitz – rhonda.s.fitz@ivv.nasa.gov

Gerek Whitman – gerek.whitman@engilitycorp.com