Results from the NASA Spacecraft Fault Management Workshop: Cost Drivers for Deep Space Missions

Marilyn E. Newhouse[1]
CSC
Marshall Space Flight Center, Alabama, 35812, USA

John McDougal[2], Bryan Barley[3], Karen Stephens[4]
National Aeronautics and Space Administration, George C. Marshall Space Flight Center,
Marshall Space Flight Center, Alabama, 35812, USA

Lorraine M. Fesq[5]
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA,91109, USA

Fault Management, the detection of and response to in-flight anomalies, is a critical aspect of deep-space missions. Fault management capabilities are commonly distributed across flight and ground subsystems, impacting hardware, software, and mission operations designs. The National Aeronautics and Space Administration (NASA) Discovery & New Frontiers (D&NF) Program Office at Marshall Space Flight Center (MSFC) recently studied cost overruns and schedule delays for five missions. The goal was to identify the underlying causes for the overruns and delays, and to develop practical mitigations to assist the D&NF projects in identifying potential risks and controlling the associated impacts to proposed mission costs and schedules. The study found that four out of the five missions studied had significant overruns due to underestimating the complexity and support requirements for fault management. As a result of this and other recent experiences, the NASA Science Mission Directorate (SMD) Planetary Science Division (PSD) commissioned a workshop to bring together invited participants across government, industry, and academia to assess the state of the art in fault management practice and research, identify current and potential issues, and make recommendations for addressing these issues. The workshop was held in New Orleans in April of 2008. The workshop concluded that fault management is not being limited by technology, but rather by a lack of emphasis and discipline in both the engineering and programmatic dimensions. Some of the areas cited in the findings include different, conflicting, and changing institutional goals and risk postures; unclear ownership of end-to-end fault management engineering; inadequate understanding of the impact of mission-level requirements on fault management complexity; and practices, processes, and tools that have not kept pace with the increasing complexity of mission requirements and spacecraft systems. This paper summarizes the findings and recommendations from that workshop, particularly as fault management development issues affect operations and the development of operations capabilities.

I. Introduction

The Discovery Program (DP) is a science program of frequent, mid-class spacecraft missions that perform high-quality focused scientific investigations. Initiated in 1992, the DP was defined to ensure frequent access to space for planetary system(s) science investigations, emphasizing missions that can be accomplished under the leadership of the scientific research community. Since its inception, DP has successfully completed missions to study the Moon,

[1] Principal Lead Systems Engineer; MSFC/VP23.
[2] Lunar Quest Deputy Program Manager, MSFC/VP23, AIAA Member.
[3] Lunar Quest Program Chief Engineer, MSFC/EE04
[4] Lunar Quest Program Integration Engineer, MSFC/VP23
[5] Principal Engineer, Engineering Development Office, Systems and Software Division, AIAA Senior Member

inner planets, asteroids, comets, and solar wind. Current missions in development or operations will continue exploration of the inner and outer planets, asteroids, comets, and the Moon. The New Frontiers Program (NFP) is a science program of medium-size spacecraft missions that perform high-quality focused scientific investigations. Initiated in 2003, the NFP was defined to pursue planetary missions that require resources beyond those available in the DP. The NFP currently includes two missions to study outer and dwarf planets. Both the DP and NFP comprise long-term series of space science missions that are independent and uncoupled, but share a common funding and management structure.

D&NF missions are ultimately defined in terms of the science return from the mission. Level I requirements include the baseline science mission: the full set of scientific requirements identified for the mission, and the threshold science mission: the minimum set of science requirements below which the mission is not considered justifiable for the proposed cost. Each mission is lead by a Principal Investigator (PI) who is held responsible for proper execution of all aspects of the mission, including implementation and execution within the confirmed mission cost and schedule.

The D&NF programs are managed by a single program office at Marshall Space Flight Center. As uncoupled, multi-mission programs emphasizing cost-capped PI-led missions, the ability of the D&NF programs to meet their launch frequency requirements is driven by the ability of each individual project to meet its proposed and confirmed LCC and schedule. A look at the history of the D&NF programs showed an increased frequency of cost overruns. Therefore, the D&NF Program Manager commissioned a study to investigate the cost escapes on recent D&NF missions, identify the primary growth drivers, and determine what *reasonable* things could be done as a program to either prevent the cost escapes or manage them better. Five missions were selected from the two programs based on a recent history of exceeding proposed or confirmed costs, and representing a spectrum of complexity, cost growth, and maturity. The study process and results are detailed in the final report, "Improving the Life Cycle Cost Management of Planetary Missions[1]."

The study found that four of the five missions experienced significant growth in fault protection and autonomy development and operations costs by launch. The study also found that the cost growth could be traced to problems that were embedded as a result of decisions that were made during formulation (phases A and B), although the impact was not realized until late in development (phases C/D) or in operations (phase E). In order to understand the drivers behind this cost growth, as well as similar growth seen on other missions in other programs, NASA's Planetary Science Division commissioned a Fault Management Workshop. The workshop was held in New Orleans in April of 2008. The results of the workshop are documented in "Spacecraft Fault Management Workshop Results for the Science Mission Directorate, Planetary Sciences Division[2]." The workshop was designed to bring together subject matter experts from across government, industry, and academia to assess the state of the practice in current mission development, understand the state of the art in academic research, identify current and potential issues, and make recommendations for addressing these issues.

One of the fundamental conclusions of the workshop was that fault management development is not limited by technology, that fault management cost growth is primarily driven by a lack of emphasis and discipline in both the engineering and programmatic dimensions. Some of the areas cited in the findings include different, conflicting, and changing institutional goals and risk postures; unclear ownership of end-to-end fault management engineering; inadequate understanding of the impact of mission-level requirements on fault management complexity; and practices, processes, and tools that have not kept pace with the increasing complexity of mission requirements and spacecraft systems.

This paper summarizes the findings and recommendations from that workshop, particularly as fault management development issues affect operations and the development of operations capabilities.

II. Fault Management for Deep Space Missions

The Fault Management Workshop focused on deep space, robotic missions that share many similar operational characteristics that drive fault management capabilities. In particular, most deep space missions must execute critical activities where there is often one and only one opportunity to execute the required activities to achieve mission success. For example, if the science goal is a fly-by of a solar system object, all science data must be collected within a window of hours to weeks, depending on the type of science measurement. There is no opportunity to repeat an observation on a subsequent orbit as for low earth orbiting missions. Even for missions designed to orbit another solar system object, constraints on mission duration combined with minimum observation requirements often result in minimal or no opportunities to revisit or repeat an observation. In addition, as the mission progresses the round-trip time delay for command and telemetry increases. Typically, by the start of science operations, the

round-trip time delay precludes real-time ground intervention in the execution of spacecraft activities (e.g., encounter or orbit insertion).

In these operational paradigms, fault management is a critical and enabling component of the system design. Continuation of spacecraft and science operations during critical activities (encounter, orbit insertion, landing, etc.) is paramount for successful execution of the science mission. The term "Fault Management" was chosen for the workshop over other names used throughout the aerospace industry, such as fault protection, to emphasize that for deep space missions, handling an in-flight fault requires more than just protecting the spacecraft and science instruments from damage. It also requires managing faults to enable continued execution of the science mission. Fault management provides the spacecraft and science instruments with the ability to identify a potential fault and respond in such a manner as to mitigate the fault and return to normal operations or to limit the effect of the fault on other spacecraft systems and avoid a "cascade" of events that would terminate critical science operations. Fault management also provides the tools to capture, store, and return the information (via telemetry) required for operations to characterize the original fault and any cascade effect, understand its effect on the science observation, and identify the root cause.

In addition, deep space missions typically have tight launch window constraints in order to reach the science target within launch mass and power margins. Missing a launch window often means a significant delay before the next available launch opportunity, or, as was the case for one D&NF mission, can result in a multi-year increase to the cruise phase duration. Thus, while launch delays are undesirable and costly for all space missions, launch delays for deep space missions are even more problematic. For some mission types, the response to late-breaking problems with fault management development and test progress can be to delay launch to accommodate any schedule slips. This is not always practical for deep space missions, where the more typical response is not to delay launch but to increase fault management resources. The staffing profile in **Figure 1,** from one of the mission case studies at the Fault Management Workshop (Ref. 2), illustrates this growth in fault management staffing late in the development cycle.

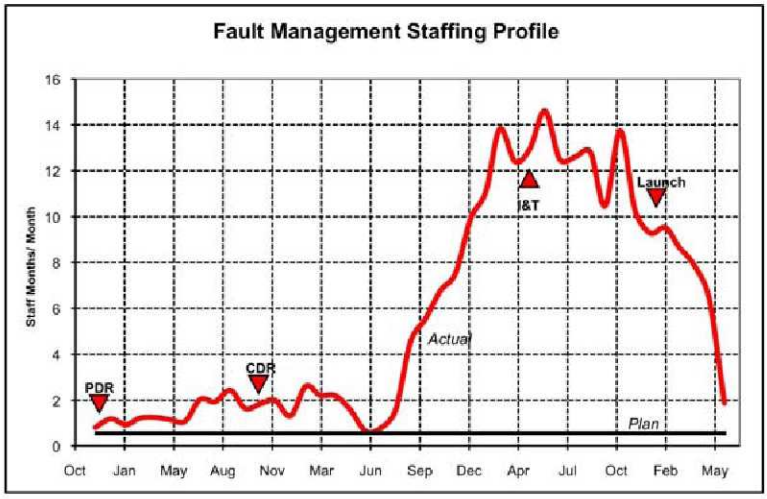


Figure 1. Planned vs. Actual Fault Management Staffing for A Workshop Case Study Mission

III. Fault Management Cost Drivers

A. Fault Management Concept, Requirements, and Architecture

Managing faults to enable continued execution of the science mission is a complex and potentially open-ended process. Setting the scope of and requirements for fault management becomes an exercise in risk management: balancing the cost of detecting, reporting, and mitigating potential onboard faults against the potential loss of science or the mission. This balance is particularly critical for D&NF's small to medium cost-capped missions.

Early allocation of resources by project management to fault management is a critical step in controlling fault management cost growth later in the project life cycle. Fault management is not always recognized as a separate and equal engineering discipline requiring support from project management and allocation of sufficient resources to ensure that fault management matures along with other spacecraft subsystems. As a result, missions often lack a clear concept for fault management during early formulation. Without a conceptual understanding of the operational scope and goals of fault management, project management cannot accurately estimate the support for development, test, and operation of fault management capabilities. All workshop participants agreed that fault management had been underestimated from the inception of their individual projects.

Defining and controlling fault management requirements was cited by all workshop attendees as an issue affecting development cost and cost growth. Projects lacked clear requirements that are carefully tied to the mission

requirements and risk posture, opening the door for scope growth and cost impacts as the design matures and operational planning begins. It is not unusual for a project to be proposed and awarded accepting a fairly high risk level in order to control total life cycle cost, and for the approach to become significantly more conservative as the launch and operations phases approach. These changes in risk posture may be imposed externally on the project by NASA, or can result from changes in project management and personnel, but just as often represent an evolution of project perspective. Regardless of the source, changes in risk posture drive fault management complexity, and therefore affect development and operations costs.

Even in the absence of a change in risk posture, case studies cited a lack of early emphasis on fault management engineering and specific fault management requirements, allowing mission requirements to be accepted or changed without a clear understanding of the impact on the complexity of the fault management implementation and the resources required to test and operate the resulting system. Given that fault management is not yet widely recognized as a separate engineering discipline, it is rarely identified separately from system engineering for trade studies, analysis, and impact assessment. This was exacerbated by unclear roles and lack of ownership of fault management as an identifiable spacecraft subsystem. Project system engineers may not have the time, or the detailed expertise in the fault management architecture and design, to accurately identify and assess all impacts to the development, testing and operation of the fault management system. The workshop concluded that project management needs to ensure that the roles and responsibilities for fault management development, testing, and operations are clearly defined and communicated to the project team, and that overall fault management complexity be included as a factor during approval of mission requirements and requirements changes.

Alternatively, heritage hardware or software may burden a project with unnecessary fault management complexity. Typically, projects propose reuse of heritage software and hardware as a means of controlling life cycle cost and reducing development risk. However, mission case studies at the Fault Management Workshop cited decisions to select a specific heritage fault management architecture or hardware configuration, without the necessary trade studies to assess applicability to the current mission, as drivers for unnecessary complexity, increased scope, and cost growth. In one case study, the heritage fault management design had limited redundancy and fault tolerance that was not consistent with the current mission implementation. Layering the required additional redundancy and fault tolerance into the heritage design greatly increased the complexity of understanding, testing, and operating the final fault management capabilities. In another case study, the process of integrating hardware and software components with different heritage into one cohesive fault management approach required several iterations of the architecture early in development, and design changes later on. In yet another case study, the heritage fault management system brought with it a level of complexity that was unnecessary for the current mission, increasing the overall support required to test and operate the fault management system. The workshop concluded that to a large extent fault management systems are not inheritable and any claims of heritage need to be carefully examined against the mission context and requirements.

Finally, the workshop concluded that there are no metrics for evaluating the relevance and merits of a fault management approach. Also, given that fault management systems are typically distributed across individual hardware components, software, databases, and ground procedures, there are no metrics for measuring the life cycle progress of fault management as a whole. This lack of metrics means that there are no easy mechanisms for estimating overall fault management complexity, estimating fault management development costs, or measuring the effect of mission requirements changes on fault management architecture, design, and operational support. Project managers need tools to understand the complexity of, understand the interdependencies of, and control development of the fault management system as a whole.

Controlling fault management growth requires clear, concise, and justifiable fault management requirements tied directly to the mission goals and requirements. It also requires an architecture selected for, and validated against, the specific mission and hardware configuration. To achieve this, project management must emphasize fault management from the earliest stages of a project. Recognizing fault management as a separate and equal engineering discipline and ensuring that the necessary resources are available to support defining the fault management concept, specifying management requirements, and performing appropriate trade studies, would be a major step toward improving early estimates of fault management development and operations costs, and controlling cost growth as the project matures.

B. Tools and Processes

During the workshop, it became clear that aside from project management issues, fault management as an engineering discipline has not matured consistent with other hardware and software engineering disciplines. Some tools do exist to support fault management engineering, for example Fault Tree Analysis (FTA) or Probabilistic Risk Assessment (PRA). The general consensus of workshop attendees was that the available fault management

architectures support the necessary fault management capabilities and designs. However, tools, clearly tied to the fault management design, that document, and help visualize the fault management system behavior are limited at this time. One case study cited the difficulty even for an experienced fault management engineer to understand the interaction of all monitors and responses for a complex fault management system. Others cited the difficulty in presenting the fault management capabilities in a clear, coherent manner to engineers from other discipline areas in order to obtain a meaningful review. One case study cited the difficulties tracing an in-flight anomaly back through the layers of a complex fault management system, in order to understand the root cause(s) and determine appropriate recovery response(s). Tools that allow management, subsystem leads, and operations personnel to understand the proposed fault management capabilities and design are critical to ensuring the final product will provide the required capabilities to support operations. Ultimately, this understanding, early in the project formulation, is critical to correctly estimating the required resources for development, test, and in-flight operations.

Another result of the under-emphasis on fault management engineering as a separate discipline is the lack of standardized terminology for fault management systems and a corresponding lack of formality in the documentation of fault management designs. One very simple example of the differences in terminology is the definition of "single fault tolerant." In one case studied at the fault management Workshop, a contractor assumed in discussions that single fault tolerance referred to a single error (e.g., spacecraft fault or operator error), while the project team assumed that to be single fault tolerant a system must be robust enough to handle a spacecraft fault and an operator error. When this discrepancy was understood and resolved, it contributed to the increase in fault management scope and the associated effort.

In another case study, misleading terminology regarding the "keep out" zone for a piece of spacecraft hardware resulted in an in-flight anomaly. From a heritage perspective, the keep out zone had been defined traditionally to preclude any penetration of the zone; in the mission context, strict enforcement of the keep out zone was not practical. However, the re-interpretation of the keep out zone was not fully understood by all project personnel, resulting in missing requirements. While the actual in-flight anomaly was recoverable, the post-anomaly analysis showed that in a different scenario, the problem could have had a significant impact on mission success.

C. Integration, Verification, and Validation

Fault management issues are exacerbated during the integration, validation, and verification (IV&V) phase. As shown in **Figure 1** above, requirements and design issues begin to manifest themselves towards the end of implementation as the project approaches IV&V. Fault management cost growth peaks during IV&V as projects add fault management resources in order to resolve emerging issues. Given that the impact of mission-level requirements on fault management complexity is not well understood, several case studies cited decisions made in the early design and implementation phases without understanding the full impact to IV&V and operations. IV&V testing is where the increase in complexity resulting from early requirements and design decisions is finally understood and becomes an issue for testing, launch preparations, and operations.

In addition, many projects include the definition of fault management as part of a systems engineer's role, and then allocate implementation to the various hardware and software subsystems, leaving it up to the integration, verification, and validation (IV&V) and operations teams to "pull it all together" at the end. That process of pulling the hardware and software together into operational threads drives out problems that reflect back into both the design and implementation of the fault management system. In addition, IV&V is typically the first point in the development life cycle when the forensic and diagnostic capabilities of fault management are available and exercised. "Quick fixes" to problems, whether "bugs" or missing capabilities, in order to support launch increase the overall complexity of the implementation and testing, as well as the likelihood of unanticipated interaction between fault management tests and responses that are driven out in subsequent tests. This iterative process of test, fix, and retest under the pressure of impending launch, becomes what one case study referred to as the fault management "death spiral" and drives the fault management cost growth both prior to and after launch.

Compounding the IV&V problems, initial underestimation of the fault management complexity is reflected in inadequate hardware resources to support fault management testing. As the complexity of the fault management implementation grows, the number of tests required to verify or re-verify the system increases even faster. This comes at a time when competition by operations and subsystem engineers for hardware resources is particularly intense. As one case study noted, this competition can increase to the point where "test as you fly" becomes in reality "fly as you test" and operations is restricted to configurations and scenarios that have been fully tested prior to launch.

Finally, while not a direct cost growth driver, two case studies cited the late fault management development and test pressure as contributors to in-flight anomalies. In one case, the root cause for an in-flight anomaly could have been identified during pre-launch testing, if test schedules had allowed for long duration tests. However, schedules,

driven by the approaching launch, were required to concentrate on validating fault management paths (monitors and responses) and did not allow time for more operationally realistic long duration scenarios. In another case study, the root cause for an in-flight anomaly was indicated in the test data, but test schedules were predicated on successful completion of the primary test goals, and did not permit the detailed analysis of the data using the full diagnostic capabilities of the fault management system to look for secondary indicators of potential problems.

All of these issues point back to the early project phases and lack of emphasis on fault management as a separate engineering discipline. They underline the need to clearly define requirements and capabilities, from which the overall fault management system complexity can be understood. The complexity can then be used to more accurately estimate support requirements, both personnel and hardware.

D. Mission Science and Fault Management

The ultimate purpose for all the missions studied was the collection and analysis of science data. Thus, the ultimate goal of the fault management system is to ensure the successful collection of science data and transfer of that data to the ground. Yet, more than one project found that the emphasis on mission science contributed to the increase in fault management scope and the potential for or actual loss of mission science.

D&NF missions are defined in terms of their science return at two levels: baseline science and threshold science. The difference between baseline and threshold science is intended to provide the trade space for descoping mission science in order to contain total LCC, but the ultimate responsibility for the mission lies with the PI. It is very difficult for PIs to make that trade and sacrifice science in order to compensate, for example, for increasing fault management costs.

After launch, it is even harder for teams to ignore pressure to maximize science time on target. At one level, this is understandable given that the science provides the purpose for the mission. However, in one case study, as a result of pressure to return to normal operations to protect critical science, the team bypassed the normal recovery procedures in favor of a "quick recovery" process. Instead of saving science time, the procedure resulted in an incomplete recovery to an intermediate safe mode level, and ultimately the loss of more science time due to the analysis of and recovery from the intermediate safe mode.

In another case, hardware control parameter values had been set based on analysis of the worst case scenario, but these values were decided to be too restrictive on science. Shortly before launch, the parameter values were modified based on a more complex analysis of routine operational scenarios in order to increase the available science time. Well into the operational mission, the change in parameter values contributed to an in-flight anomaly that resulted in loss of science time, and, under other circumstances, could have resulted in loss of mission.

While neither of these last two examples contributed directly to fault management cost overruns, they provide clear examples of the secondary implications of a lack of fault management engineering discipline, unclear ownership of fault management capabilities, and missing or poorly scoped fault management requirements on the mission and mission science. Project management needs to have the appropriate tools: clear requirements defining both science and fault management, in order to make the necessary trades between the two both before and after launch.

IV. Conclusion

Fault management is a critical and enabling component of deep space missions. It is a primary means of controlling mission risk and ensuring mission success. Yet the workshop concluded that fault management does not yet have the recognition and support necessary for controlled implementation. From project inception, project management needs to establish fault management as a separate engineering discipline within the project and allocate personnel and hardware resources accordingly. In addition, project management needs to ensure that mechanisms are in place to measure fault management progress and growth as a whole, taking into account the interdependencies across all hardware, software, and operations components. Project management, systems engineering, and fault management engineers need to work together to ensure that the proper processes and controls are in place to ensure that fault management requirements are fully developed and directly mapped to mission requirements early in the development life cycle, that the appropriate trades are performed to match the fault management architecture and design solution to the mission, and to ensure that trade decisions made in other engineering disciplines are fully analyzed for any impact on the overall complexity of the fault management system. In addition, fault management needs to be addressed as an independent engineering discipline. The associated terminology needs to be standardized and supporting tools developed to ensure effective communication between all project members regarding the fault management approach, design, and capabilities.

None of these are enabling or limiting technologies. Clearly the overall mission success rate indicates that NASA is effectively building fault management systems that protect and enable science missions. Yet, this success is not without cost to NASA, and as with the case for the D&NF programs, not without cost to the ability to fund future missions to further scientific exploration of the solar system.

Appendix A. Glossary

Autonomy: The ability of the spacecraft to operate without intervention from ground operations. Together with fault protection, it defines the capability of a spacecraft to execute a critical science activity in deep space (e.g., encounter) without real-time intervention.

Baseline Science Requirements: That mission which, if fully implemented, accomplishes the entire set of scientific objectives identified at the initiation of the mission.

Fault Management: The detection of and response to in-flight anomalies. The response may be "layered," some occurring autonomously onboard, others requiring intervention from the ground. Combines aspects of Fault Protection and Autonomy.

Fault Protection: The use of cooperative design of flight and ground elements (including hardware, software, procedures, etc.) to detect and respond to perceived spacecraft faults. Its purpose is to eliminate single point failures or their effects and to ensure spacecraft system integrity under anomalous conditions. [3]

Heritage Systems: Hardware, software, and procedures with previous flight history that are reused for a new mission in order to enable a mission capability or reduce overall mission cost, schedule, or risk.

Inheritance: The process of evaluating the compatibility and benefits of heritage systems to the requirements of a new project, and validating the level of reuse or rework (design, fabrication or coding, process or procedure development, documentation) required to use the heritage system in the new mission environment.

Threshold Science: The minimum scientific requirements below which the mission is not considered justifiable for the proposed cost. Also referred to as **Minimum Science** and **Science Floor**.

Acknowledgments

The Discovery & New Frontiers Program Office Life Cycle Cost Study was performed under the direction of Paul Gilbert (MSFC), led by Bryan Barley (MSFC), and supported by Kenny Mitchell (MSFC-retired) and Marilyn Newhouse (CSC).

The Fault Management Workshop was organized by Lorraine Fesq (Caltech/JPL) and hosted by the Discovery and New Frontiers Program Office at Marshall Space Flight Center for the Planetary Science Division in the Science Mission Directorate at NASA Headquarter. Guidance for the workshop was provided by the Steering Committee members: John McDougal (NASA/MSFC), Chris Jones (Caltech/JPL), George Cancro (JHU-APL), Steven Scott (NASA/GSFC), and Raymond Whitley (NASA/GSFC).

Part of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1]Barley, B., Gilbert, P., and Newhouse, M., "Improving the Life Cycle Cost Management of Planetary Missions," February 2010.

[2]Fesq, L., Cancro, G. Jones, C., Ingham, M., Leitner, J., McDougal, J., Newhouse, M., Rice, E., Watson, D., Wertz, J., "Spacecraft Fault Management Workshop Results for the Science Mission Directorate, Planetary Sciences Division," March 2009

[3]NASA Preferred Reliability Practices, "Fault Protection," PD-EC-1243, October 1995.

National Aeronautics and Space Administration

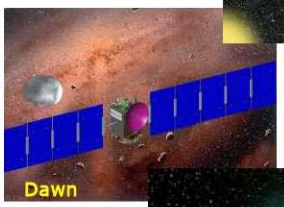
Results from the NASA Spacecraft Fault Management Workshop: Cost Drivers for Deep Space Missions

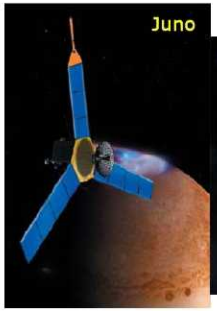
SpaceOps 2010 Conference

John McDougal, MSFC
Marilyn Newhouse, CSC
Bryan Barley, MSFC
Karen Stephens, MSFC
Lorraine Fesq, JPL

D&NF Program Science Missions

- The Moon (Lunar Prospector, M^3, GRAIL)
- Mars (Mars Pathfinder, ASPERA-3)
- Inner Planets (MESSENGER, Strofio)
- **Outer Planets (New Horizons, Juno)**
- Comets (CONTOUR, Stardust, Deep Impact, EPOXI, NExT)
- Asteroids (NEAR, Dawn)
- Interplanetary Space (Genesis)
- Extra-Solar System (Kepler)





Definition of Fault Management (FM)

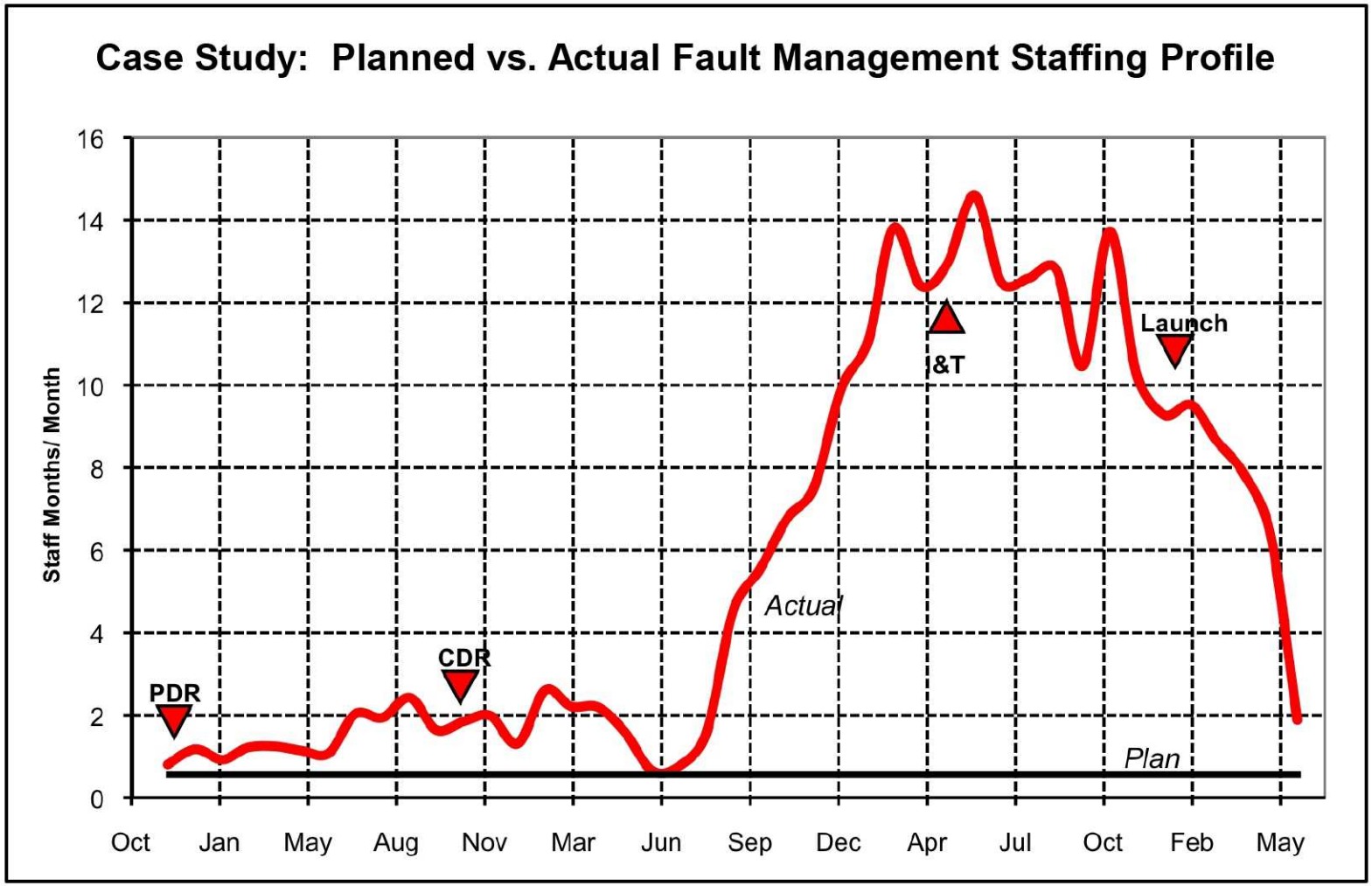
Fault Management (FM): The detection of and response to in-flight anomalies. The response may be "layered," some occurring autonomously onboard, others requiring intervention from the ground. Combines aspects of Fault Protection and Autonomy.

Fault Protection: The use of cooperative design of flight and ground elements (including hardware, software, procedures, etc.) to detect and respond to perceived spacecraft faults. Its purpose is to eliminate single point failures or their effects and to ensure spacecraft system integrity under anomalous conditions.

Autonomy: The ability of the spacecraft to operate without intervention from ground operations. Together with fault protection, it defines the capability of a spacecraft to execute a critical science activity in deep space (e.g., encounter) without real-time intervention.

"Fault Management" emphasizes the requirement to enable continued execution of the science mission, not just protect the spacecraft and science instruments from damage

Fault Management Growth



The Discovery and New Frontiers Life Cycle Cost Study found that all missions studied experience significant growth in the resources required to implement and test FM capabilities

Fault Management Cost Growth Drivers

In April 2008, NASA held a Fault Management Workshop

- **Assess the state of the practice**
- **Understand the state of the art**
- **Make recommendations for addressing the issues identified**

The findings from the workshop identified several areas as cost drivers for FM implementation

- **Different, conflicting, and changing institutional goals and risk postures**
- **Unclear ownership of end-to-end fault management engineering**
- **Inadequate understanding of the impact of mission-level requirements on fault management complexity**
- **Practices, processes, and tools that have not kept pace with the increasing complexity of mission requirements and spacecraft systems**

FM Concept, Requirements, and Architecture

All workshop participants agreed that fault management had been underestimated from the inception of their individual projects

- Insufficient resources allocated early in project formulation by FM by project management
- Lack of recognition of FM as a separate and equal engineering discipline
- Unclear roles and lack of ownership of FM requirements
- Unclear concept for FM during early formulation and changing risk posture over implementation
- Lack of clearly defined FM requirements tied to the agreed upon risk posture
- Allowing mission requirements to be accepted or changed without a clear understanding of the impact on FM complexity
- Accepting heritage hardware or software components without careful analysis of the impact on FM complexity
- Lack of FM metrics for
  - evaluating the relevance and merits of a FM approach
  - measuring the life cycle progress of FM as a whole.
  - estimating overall FM complexity, development costs, or
  - measuring the effect of mission requirements changes on FM architecture, design, and operational support

FM Tools and Processes

Available fault management architectures support the necessary fault management capabilities and designs, but tools to document and visualize the fault management system behavior are limited

- **Difficult to understand the interaction of all monitors and responses for a complex fault management system**
- **Difficult to presenting the FM capabilities in a clear, coherent manner to obtain a meaningful review**
- **Difficult to trace an in-flight anomaly back to understand the root cause(s) and determine appropriate recovery response(s**

FM lacks a standardized terminology for fault management systems and a corresponding formality in the documentation of fault management designs

- **Confusion over the concept of "single fault tolerant"**
- **Misleading use of the term "keep out"**

FM Integration, Verification, and Validation

FM issues begin to manifest themselves as the project approaches IV&V; FM cost growth peaks during IV&V

Cost growth factors

- **Increase in complexity resulting from early requirements and design decisions**
- **Process of pulling the FM components together into operational threads drives out problems that reflect back into the design and implementation of FM**
- **Initial execution of the forensic and diagnostic capabilities of fault management**
- **Increasing overall FM complexity due to quick fixes to problems, whether "bugs" or missing capabilities**
- **Inadequate hardware resources to support FM and operations testing**

Contribution of late fault management development and test pressure to in-flight anomalies

- **Insufficient time for long duration tests resulted in overlooking the root cause of a subsequent in-flight anomaly**
- **Insufficient time for detailed analysis of test data resulted in overlooking a secondary indicator of a subsequent in-flght problem**

Mission Science and FM

The ultimate goal of the FM system is to ensure the successful collection and transmission of science data and transfer of that data to the ground

Overemphasis on mission science contributes to the increase in FM scope and the potential for or actual loss of mission science

- **Principal Investigators and project management trade between FM complexity and science return in order to compensate for increasing FM cost**
- **Teams must balance the pressure to maximize science time on target against the risk of accelerating recovery after an anomaly**
- **Teams need to carefully analyze in-flight deviations from "fly as you test" designed to increase the available science time**

Conclusion

FM is a critical and enabling component of deep space missions and a primary means of controlling mission risk and ensuring mission success

Yet, the workshop concluded that FM does not yet have the recognition and support necessary for controlled implementation

- **Industry needs to address FM as an independent engineering discipline**
  - **FM terminology needs to be standardized**
  - **FM support tools need to be developed to ensure effective communication between all project members**
- **Project management needs to establish FMas a separate engineering discipline within the project and allocate personnel and hardware resources accordingly**
- **Project management needs to ensure that mechanisms are in place to measure FM progress taking into account the interdependencies across all hardware, software, and operations components.**
- **Project management, systems engineering, and fault management engineers need to work together to ensure**
  - **FM requirements are fully developed and directly mapped to mission**
  - **appropriate trades are performed to match the fault management architecture and design solution to the mission**
  - **trade decisions made in other engineering disciplines are fully analyzed for impact on the complexity of FM**

Acknowledgments

The Discovery and New Frontiers Program Office Life Cycle Cost Study was performed under the direction of Paul Gilbert (MSFC), led by Bryan Barley (MSFC), and supported by Kenny Mitchell (MSFC-retired) and Marilyn Newhouse (CSC)

The Fault Management Workshop was organized by Lorraine Fesq (Caltech/JPL) and hosted by the Discovery and New Frontiers Program Office at Marshall Space Flight Center for the Planetary Science Division in the Science Mission Directorate at NASA Headquarter. Guidance for the workshop was provided by the Steering Committee members: John McDougal (NASA/MSFC), Chris Jones (Caltech/JPL), George Cancro (JHU-APL), Steven Scott (NASA/GSFC), and Raymond Whitley (NASA/GSFC).

Part of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Supplemental Data and Backup

Acronyms

AO	Announcement of Opportunity
AR	Acceptance Review
ARR	ATLO Readiness Review
ATLO	Assembly, Test, and Launch Operations
CDR	Critical design Review
CR	Confirmation Review
CRR	Confirmation Readiness Review
CSR	Concept Study Report
D&NF PO	Discovery and New Frontiers Program Office
DOE	Department of Energy
DPI	Deputy Principle Investigator
EM	Engineering Model
EVM	Earned Value Management
FPP	Flight Practices and Procedures
FRR	Flight Readiness Review
FTE	Full-Time Equivalent
FY	Fiscal Year
IAT	Independent Assessment Team
ICE	Independent Cost Estimate
IRT	Independent Review Team
LCC	Life-Cycle Costs
LRR	Launch Readiness Review
LV	Launch Vehicle
MM	Mission Manager
MOA	Memorandum of Agreement
MRR	Mission Readiness Review
PBR	Project Budget Report
PCA	Program Commitment Agreement
PDR	Preliminary Design Review
PDS	Planetary Data System
PI	Principle Investigator
PLRA	Program Level Requirements Appendix (to the Program Plan)
PM	Project Manager
PO	Program Office
PSD	Planetary Science Division
RTG	Radioisotope Thermoelectric Generator
RM	Resource Manager
RY	Real Year
SCU	???
SMD	Science Mission Directorate
SRB	Standing Review Board
SRR	Systems Requirements Review
TMCO	Technical, Management, Cost, and Other
WBS	Work Breakdown Structure

Glossary

Autonomy: The ability of the spacecraft to operate without intervention from ground operations. Together with fault protection, it defines the capability of a spacecraft to execute a critical science activity in deep space (e.g., encounter) without real-time intervention.

Baseline Science Requirements: That mission which, if fully implemented, accomplishes the entire set of scientific objectives identified at the initiation of the mission.

Fault Management: The detection of and response to in-flight anomalies. The response may be "layered," some occurring autonomously onboard, others requiring intervention from the ground. Combines aspects of Fault Protection and Autonomy.

Fault Protection: The use of cooperative design of flight and ground elements (including hardware, software, procedures, etc.) to detect and respond to perceived spacecraft faults. Its purpose is to eliminate single point failures or their effects and to ensure spacecraft system integrity under anomalous conditions.

Heritage Systems: Hardware, software, and procedures with previous flight history that are reused for a new mission in order to enable a mission capability or reduce overall mission cost, schedule, or risk.

Inheritance: The process of evaluating the compatibility and benefits of heritage systems to the requirements of a new project, and validating the level of reuse or rework (design, fabrication or coding, process or procedure development, documentation) required to use the heritage system in the new mission environment.

Threshold Science: The minimum scientific requirements below which the mission is not considered justifiable for the proposed cost. Also referred to as **Minimum Science** and **Science Floor**.