Improving Operations: Metrics to Results

Grant B. Faris[1] and Larry W. Bryant[2]
Jet Propulsion Laboratory/California Institute of Technology
Pasadena, CA 91109

As a result of the mission failure of the Mars Climate Orbiter (MCO) spacecraft in 1999, the Jet Propulsion Laboratory (JPL) initiated the development of a Mission Operations Assurance (MOA) program to be implemented across all flight projects managed by JPL. One of the initiatives undertaken in 2001 was the collection of data on command file errors occurring in the operational phase of the mission. This paper defines command file errors and how and where they occur in the operations process. It also describes the problem reporting system (PRS) in use for mission operations at JPL. We examine the recent modifications to the PRS that enable the collection of metrics, specifically on command file errors. This paper discusses what the data show us since metrics have been collected for the operational missions conducted by JPL. We examine the evolution of an operational working group initiative to evaluate proximate, contributing, and root causes for the errors. As part of this discussion we see what the metrics have indicated over a decade. At the macro level, we can say that the aggregate command file error rate has been cut to roughly one third of the initial 2001 level by the end of 2011. Additionally, we explore efficient and innovative means to continually integrate the findings and recommendations from the working group back into the flight operations environment.

I. Introduction

In direct response to the mission failure of the Mars Climate Orbiter (MCO) spacecraft in 1999, the Jet Propulsion Laboratory (JPL) mandated a Mission Operations Assurance (MOA) program for implementation across all flight projects. Mission Assurance (MA) programs were well established for flight project development, and MOA had been a developing discipline since the Galileo launch timeframe in 1989. The MCO failure provided a wakeup call about the need to have a robust MA/MOA program for the post-launch timeframe. An early initiative undertaken within the MOA program was the identification of and the collection of data on command file errors occurring in the operational phase of the missions. The consensus was that command file errors could represent a significant threat to mission success, but a threat that could very likely be mitigated more readily than some of the other threats. Below we define command file errors and describe the evolution of the metrics data collection process. To improve the collection and analysis process, we introduced modifications to the Problem Reporting System (PRS) to support capture of metrics and characterize command file errors during mission operations. Over the years, a number of error mitigations have been implemented. The data show a generally decreasing trend in command file errors since metrics have been collected. An institutional operations working group has evolved and is looking at proximate, contributing, and root causes for the errors. We now have initial results of efforts taken to integrate findings and recommendations back into the operational environment, including specifics of the Gravity Recovery And Interior Laboratory (GRAIL) and Juno missions, which launched in the fall of 2011. The collection of data and analysis of command file errors began with our working group under the auspices of the Mission Management Office.

[1]Chief Mission Operations Assurance Manager, 5150-Mission Assurance Management Office, 4800 Oak Grove Drive, MS 264-235, 818-393-1406
[2]Mission Operations Assurance Manager for Juno, MRO, MER, and WISE, 5150-Mission Assurance Management Office, 4800 Oak Grove Drive, MS 264-767, 818-393-7875, Senior Member

II. Early Operations Working Group

As part of the mission operations assurance program, an initial multimission operations working group composed of JPL and contractor personnel was formed in 2001 to look at ongoing command file errors committed in flight operations. At this point the focus was on four specific projects supported by the JPL Mission Management Office. These projects were Mars Global Surveyor (MGS), Stardust (SDU) Mars Odyssey (ODY), and Genesis (GEN). The outputs of the working group were proposed improvements to the command uplink process emphasizing command reliability. The working group was composed of individuals from mission management, systems engineering, operations assurance, real-time flight control, mission planning and sequencing, and ground data systems. The group originally sorted the errors into four areas: real-time, process, unexpected result, and noninteractive. Real-time errors were those that occurred during the uplink process after the commands had been generated, tested, and provided to the operations controller for transmission to the spacecraft. Examples of these errors are sending a command at the wrong time or sending it in violation of constraints placed on the transmission process. Process errors identified mistakes occurring somewhere between the initiation of the commands and their transfer to the operations controller. An inaccuracy in the time requested for execution or overlooking undesirable end states while reviewing test results are errors in this category. An unexpected result is a category of errors that normally results from an unknown feature of hardware or software behavior. While these are not common, an example would be a device rotating in a counter-clockwise direction to reach a specific position because the device thought it was near a hard stop in the clockwise direction. The fourth category of noninteractive errors was a "catch all" for mistakes by instrument teams in requesting commands that did not interact with any other elements of the spacecraft, but which contained an error that resulted in an unexpected/undesired behavior of the instrument. Throughout this effort, there were 16 specific process improvements identified for implementation. These were incorporated in the corrective actions documented as part of the PRS closure process for reports documenting command file errors. Additionally, 40 action items were identified. The seven overarching recommendations included:

1) Take more time to plan and review.
2) Minimize the use of the real-time command process.
3) Aggressively bring in past personnel expertise during critical events, new activities, recovering from anomalies, and transitioning from one mission phase to another.
4) Use the standardized command uplink process.
5) Use the test-bed capability more. When in doubt, err on the side of too much testing.
6) Conduct flight team training on a regular basis for both new and ongoing missions that includes command file error statistics.
7) Conduct command file error prevention training in a case study format for flight team members.

From 2002 to 2009, command file error statistics continued to be collected and presented at flight team training sessions for new and ongoing flight projects. In 2009, a relatively small operations working group was formed with JPL and contractor personnel to discuss command file errors with emphasis on root cause and corrective actions. A command file error reporting template was developed with the format shown in Table 1.

Table 1. Command file error template.

Project: G	ISA #	Incident Date:	Command File Error Category:	Command File Error Criticality:
Command File Error Description (Undesired outcome) including a generic and more detailed description.				
Proximate Cause (The event(s) that occurred, including any condition(s) that triggered the undesired outcome.)				
Contributing Cause (The event(s) or condition(s) that may have contributed to the occurrence of an undesired outcome but, if eliminated or modified, would not by itself have prevented the occurrence.)				
Root Cause (The event(s) or condition(s) that led to the proximate cause and subsequent undesired outcome and, if eliminated, or modified would have prevented the undesired outcome.				
Corrective and Preventive Actions (include immediate and long-term), and Recommendations across projects.				

III. Institutional Operations Working Group

As this effort continued, it was determined a more formalized operational working group should be instituted to analyze the errors and explore ways to integrate the recommendations across all flight projects in operations. In November 2010, a JPL institutional operations working group was formed to provide a forum for the identification, discussion, and implementation of corrective action over a broad range of operational issues across JPL institutional boundaries. The initial task was to review the command file errors in 2009, identify in the process where they occurred, and draw lessons to be incorporated back into the projects and institution. The group met over the next three months and documented their recommendations, which were then disseminated to development projects approaching launch and early cruise operations (Juno, GRAIL, MSL, NuSTAR) as well as ongoing flight missions. The proximate/root causes were grouped into four major categories of:

1) Loss of rigor (inattention to detail, complacency, inadequate review, procedures not followed, miscommunication, distraction, multitasking)
2) Situational awareness (inadequate knowledge of the spacecraft state as a function of time)
3) Flight team work overload (stress, fatigue, rush in getting task done)
4) Nonstandard activities (doing activities in different ways, first time events on the spacecraft)

The themes/findings fell into nine general categories that subdivided into development and operations related areas, listed in Table 2.

Table 2. Themes/Findings.

Categories	Operations Recommendations	Development Recommendations
1. Test-bed/ simulation/ modeling	1. Maintain a rigorous review process of the test-bed results; functionally verify whenever possible	1. Implement a fast simulator (faster than real-time) for operations 2. Implement a test-bed time jump capability 3. Work on spacecraft models and simulations while still in ATLO and continually update throughout operations. Historically, flight rules come in late in the development process as subsystem experts are involved in testing of their flight hardware.
2. Spacecraft state/parameter tracking		1. Implement an automated flight system state/parameter tracking system that ties the uplink command system to the downlink telemetry (What was commanded – What was expected – What was seen). 2. Do not allow invisible onboard application-to-application commanding resulting in no visibility of parameter states. 3. Ensure every command has a default value. 4. Ensure all commandable states can be seen in downlink telemetry.
3. Operations sustainability	1. Perform and document a thorough tabletop planning session at the beginning of each uplink activity to include as a minimum beginning/ending spacecraft states, detailed verification/validation plans, transmission constraints, and contingency commands. 2. Look for ways to bring back technical experts for subsystem parameter reviews prior to critical events. (Loss of corporate knowledge is a significant risk item as the operations continues.) 3. Perform a detailed technical review of the impact to other subsystems in the presence of a spacecraft hardware anomaly (as part of the daily operational activity of the flight team). 4. Plan and fund for software tool modifications and periodic hardware upgrades to keep up with the flight system changes as flight operations goes along (including extended missions).	

Categories	Operations Recommendations	Development Recommendations
	5. Perform a detailed technical spacecraft subsystem review when transitioning from the primary to the extended mission, including key documentation. 6. Set up a schedule to update operational procedures on a regular basis (quarterly, semi-annually, annually), and engage flight team members in the activity. 7. Maintain the spacecraft idiosyncrasy document throughout the testing and operations phase of a project. 8. Maintain an operational risk management process throughout operations. 9. Include as part of a project's closeout activities a thorough review and possible changes to the spacecraft states with a view toward a follow-on mission. (The spacecraft should be placed in a known and documented configuration in anticipation of follow-on activity.)	
4. Operations development		1. Incorporate extended mission requirements early in the design of flight and ground systems. 2. Develop requirement for an operations environment that is easily maintained and documented. This may include the implementation of a model based engineering effort. 3. Identify flight operational tools early in the development process to ensure availability for flight operations. 4. Consider performing a wholesale SEQGEN revitalization effort.
5. Operations training	1. Loss of experienced personnel needs to be supplemented with a well-defined training program for new flight team personnel coming into operations. 2. Implement a flight team training program throughout the mission operations phase including: anomaly response exercises, flight team schools, operational error case studies, review of operational procedures, etc. 3. Implement a flight team certification process tied, in part, to the training program. Certification should assess the readiness of each individual to perform his/her assigned duties. Team members should be recertified on a regular basis (yearly or at most every other year).	
6. Human factors	1. Look for ways to keep the flight team engaged in the day-to-day operational activities. Complacency can easily creep in as the mission proceeds and all is going well. 2. While overtime and team fatigue may be inevitable at times (ATLO comes to mind), managers should monitor team fatigue levels and relieve overloads whenever possible. Watch out for individuals oversubscribing themselves.	
7. Configuration management	1. Maintain a rigorous configuration management process with emphasis on command products, testbeds, and operational procedures (nominal and contingency).	

Categories	Operations Recommendations	Development Recommendations
8. Operability infusion		1. Include operations personnel on the project systems engineering team early and throughout the project development phase, to facilitate a flight system design that is less error-prone to operate. 2. Develop and impose appropriate operability requirements on both the flight and ground systems. 3. Establish an operability working group during development to track and address operability issues.
9. Expert systems		1. Look for ways to automate flight software reasonableness checks with a goal of checking the intent of a command or sequence load.

IV. Improving Metric Collection

As we continued to collect metrics, it became imperative that the process be formalized and integrated into the Incident, Surprise, Anomaly (ISA) reports that are used in operations and part of JPL's web-based PRS. Using JPL's Anomaly Resolution Standard and the template shown in Table 1, a new module for the PRS was developed and deployed in the spring of 2011. As part of the ISA report closure process, the Mission Operations Assurance Manager (MOAM) is now required to identify each ISA documenting problems that result from a command file error, which is defined as one of the following, regardless of the effect on the spacecraft:

1) an error in a command file that was sent to the spacecraft;
2) an error in the approval, processing, or uplinking of a command file that was sent to the spacecraft; or
3) the omission of a needed command file that was not uplinked to the spacecraft.

Having identified an ISA for a problem resulting from a command file error, the MOAM is required to fill out the fields in the Command File Error section of the ISA shown in Fig. 1.

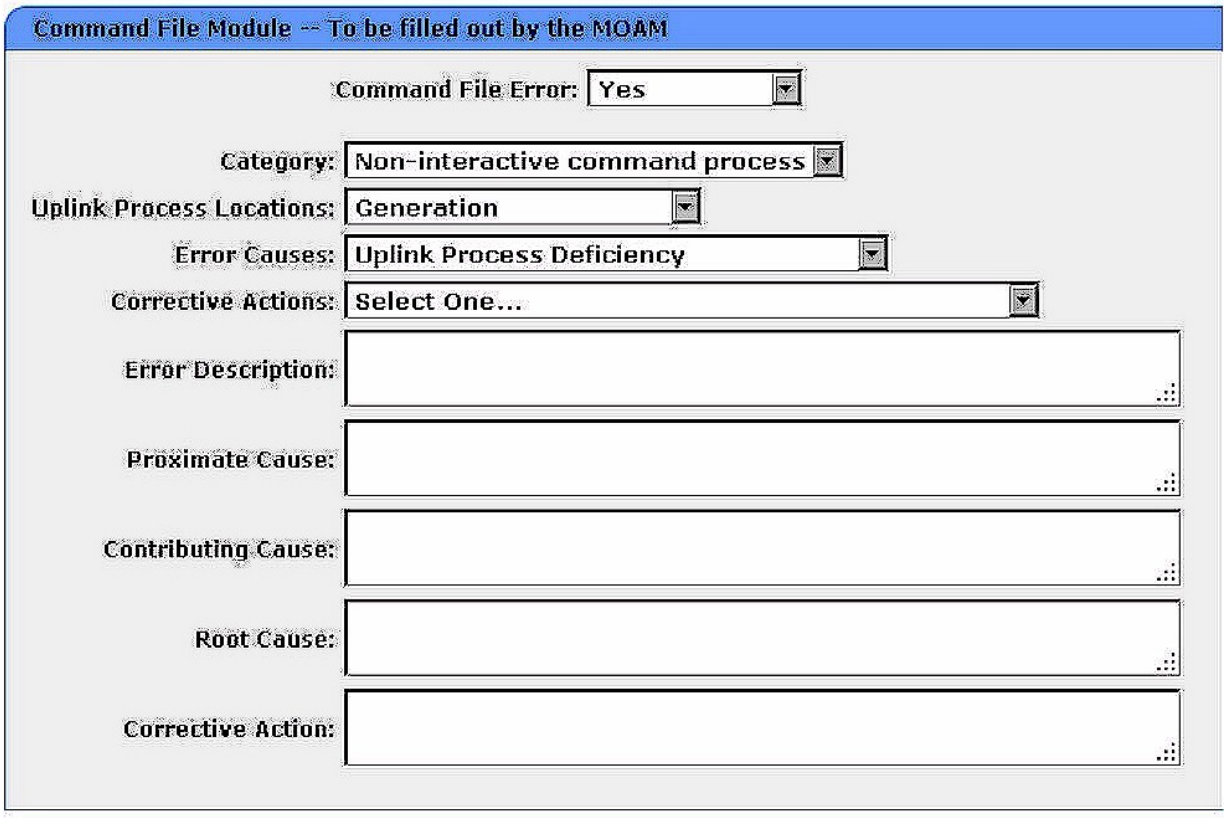

Figure 1. Command file error module in PRS.

The command file error is categorized as occurring in the interactive or non-interactive command uplink process. The place in the process where the error was first introduced is identified. The options include initiation, generation, testing, review, approval, processing for radiation, and uplink. At a high level, there are drop-down menus for error causes and corrective actions, based on JPL's Anomaly Resolution Standard. Selectable causes include human/procedural errors, uplink process deficiency, testbed/simulation/modeling setup errors, post-launch flight software error, ground software error, configuration management deficiency, and tracking configuration. The corrective action menu includes flight team training, procedural command uplink process modification, automated command uplink process modification, flight software modification, ground software modification, and configuration management modification. An example of the automated template is shown in Table 3.

Table 3. Command file error report from PRS.

Command File Error Report

4/19/2012 at 11:53:06 AM

Project: Project Y	**ISA#:** XXXXX **Incident Date:** 1/6/2011 **Command File Error Location:** Review **Criticality:** 2
Command File Error Description (Undesired outcome) including a generic and more detailed description.	Investigation into the cause showed that a part of the sequence caused the rejection: The argument must be EQ 0 or EQ -1
Proximate Cause (The event(s) that occurred, including any condition(s) that triggered the undesired outcome.)	Parameter value used outside of range of acceptable values.
Contributing Cause (The event(s) or condition(s) that may have contributed to the occurrence of an undesired outcome but, if eliminated or modified, would not by itself have prevented the occurence.)	Flight rule check had an error in the values it checked.
Root Cause (The event(s) or condition(s) that led to the proximate cause and subsequent undesired outcome and, if eliminated, or modified would have prevented the undesired outcome.)	Not understanding the cascading/interactive effect of a sequence which wasn't necessarily needed failing and causing the entire sequence to fail.
Corrective and Preventive Actions (include immediate and long-term), and Recommendations across projects.	Updated flight rule check.

Project: Project Z	**ISA#:** XXXXX **Incident Date:** 3/30/2011 **Command File Error Location:** Uplink **Criticality:** 3
Command File Error Description (Undesired outcome) including a generic and more detailed description.	The commands were not sent with the specified timing after last bit of previous command' but erroneously was commanded with no delay between commands.
Proximate Cause (The event(s) that occurred, including any condition(s) that triggered the undesired outcome.)	None
Contributing Cause (The event(s) or condition(s) that may have contributed to the occurrence of an undesired outcome but, if eliminated or modified, would not by itself have prevented the occurence.)	None
Root Cause (The event(s) or condition(s) that led to the proximate cause and subsequent undesired outcome and, if eliminated, or modified would have prevented the undesired outcome.)	Inattention to standard practice for command transmission.
Corrective and Preventive Actions (include immediate and long-term), and Recommendations across projects.	operator counseled on reviewing command forms and following proper practice.

V. Command File Error Statistics

Finally, the command file error statistics are tabulated by dividing the number of command file errors per year by the total number of commands transmitted to the spacecraft during the year. Figure 2, contains statistics for command file errors from 2001 through 2011. As you can see, the overall command file error rate has been significantly reduced since we began collecting metrics back in 2001 and has been holding fairly steady since 2006. Projects included in the metrics are MGS, Stardust, Mars Odyssey, Genesis, Spitzer, Mars Reconnaissance Orbiter, Phoenix, Galex, Dawn, MER A, MER B, Cassini, GRAIL A, GRAIL B, Juno, and MSL.

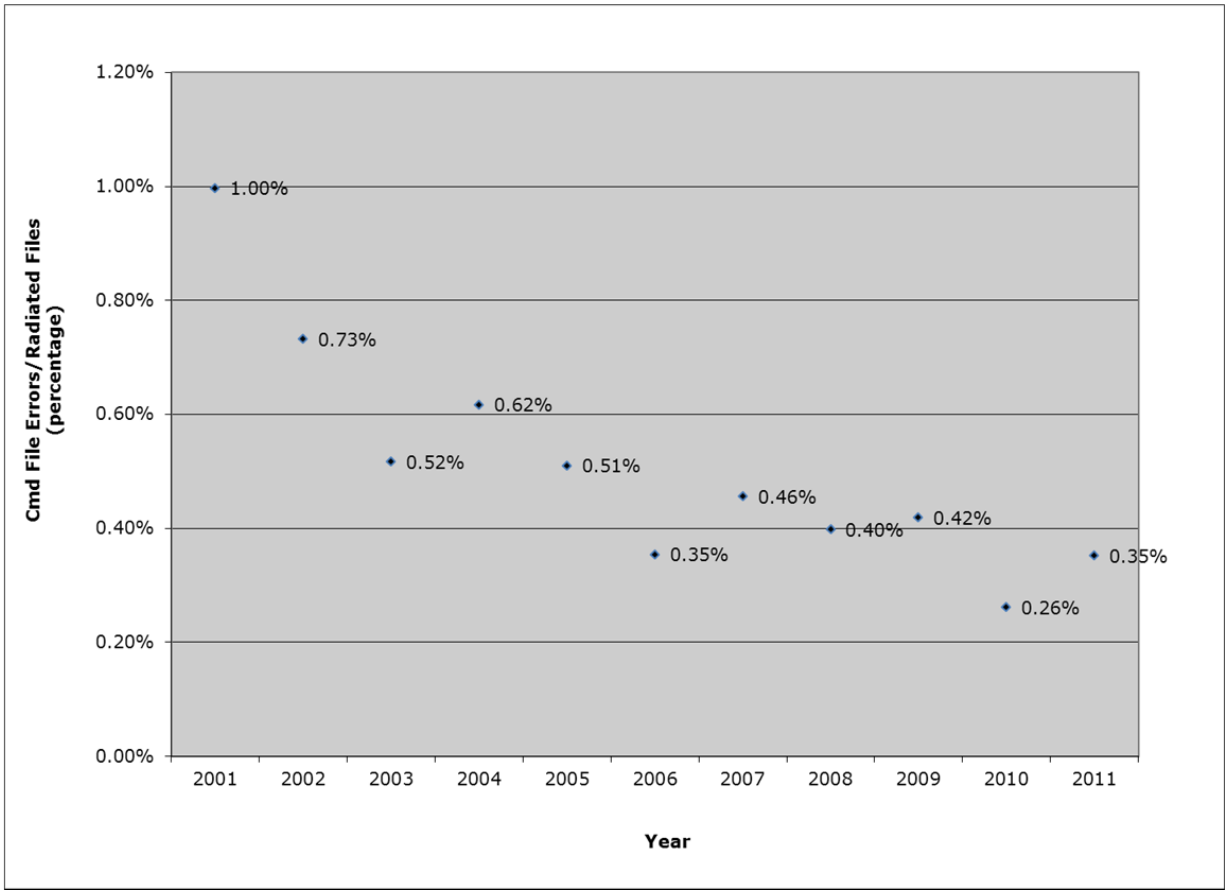

Figure 2. Aggregate command file error rate for each mission year of operations.

Another way to look at the data is in tabular form, shown in Table 4. The command file errors are broken out by criticality ratings, with one being of most concern and four being of least concern. The table also provides the number of command files transmitted throughout the years along with the average number of command file errors committed by the projects per year. In 2001, each project, on average, was committing one command file error per month which steadily declined over the years to one command file error being committed per project every three to four months in 2011.

Table 4. Aggregate command file errors by criticality for each mission year of operations.

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011
Criticality 1	11	0	7	11	7	3	5	2	3	0	0
Criticality 2	26	16	12	23	19	3	14	13	16	1	13
Criticality 3	22	21	26	46	38	43	43	54	30	27	30
Criticality 4	1	2	5	4	2	1	1	0	0	0	2
Total Command File Errors	60	39	50	84	66	50	63	69	49	28	45
Total Command Files Transmitted	6019	5331	9682	13618	12944	14131	13834	17339	11700	10709	12798
# Projects Included	5	5	7	9	9	9	10	10	9	9	13
Aggregate Command File Error Rate	1.00%	0.73%	0.52%	0.62%	0.51%	0.35%	0.46%	0.40%	0.42%	0.26%	0.35%
Command File Errors/# of projects	12.0	7.8	7.1	9.3	7.3	5.6	6.3	6.9	5.4	3.1	3.5

VI. Applying The Lessons of the Working Group

In preparation for her launch, Juno implemented the standard reviews, training, and uplink processes that had been evolving since 2001. Following the launch of Juno in August 2011, after an unanticipated flurry of command file errors, the project initiated an additional proactive effort involving the spacecraft, science, navigation, and sequence teams to analyze and continually improve their operational processes. In February 2012, the project convened a review of the implemented and proposed improvements to a panel of independent experts to comment and provide suggestions. The project categorized the command file errors into five areas including: configuration file management, command stem errors, command file content errors, flight rule violations, and on-console/transmission errors. By far the largest numbers were errors in the content of command files.

One example content file error occurred when a block in a background sequence expanded to the extent that a subsystem power-off unexpectedly took place after a power-on for the same subsystem. The intent was for the power-on to remain in effect beyond the block execution. Another power-on was sent after the block completed and before the subsystem was needed for the other planned activities, so no harm was done by this error. Another example was an instrument team having a limited command set available for a test and selecting to transmit a file that performed the desired function but also commanded a mode change, which, based on the instrument's state, was an illegal mode change. Clearly, in these two examples, things were not being thoroughly planned in advance. In response, the project increased the rigor in their sequence kickoff process and change request documentation to ensure better communication amongst the teams. The project also addressed the concern that unplanned instrument activities in early cruise were placing additional workload on the flight team. Both of these issues related directly to finding 3.1 in Table 2, which states, "Perform and document a thorough tabletop planning session at the beginning of each uplink activity to include as a minimum beginning/ending spacecraft states, detailed verification/validation plans, transmission constraints, and contingency commands" and finding 6.2, which states, "While overtime and team fatigue may be inevitable at times (ATLO comes to mind), managers should monitor team fatigue levels and relieve overloads whenever possible. Watch out for individuals oversubscribing themselves."

The independent review concluded that Juno was indeed taking the proper approach to minimize future command file errors. It can also be noted that the GRAIL project had, through the design of their mission and mission operations system, effectively incorporated these two findings in their process. Because of the short cruise timeframe, GRAIL had planned and tested all of their mission activities prior to launch, which meant that the interaction of all activities was well understood before being implemented. Additionally, the project had separate Systems Leads for each of the GRAIL A and GRAIL B spacecraft and sufficient subsystem staffing to support simultaneous tracks of each spacecraft when needed. With the similarity of spacecraft and operational activities, this allowed the team to effectively spread the workload and maintain a reasonable level of strain on all members of the flight team. The result of GRAIL's efforts has been a significantly lower rate of command file errors than seen on past projects in the first few months after launch. In short, these two projects provided validation of the efforts of the institutional operations working group.

VII. Conclusion

Beginning well before the issues with Mars Climate Orbiter, a mission operations assurance discipline was evolving at JPL. With the MCO incident, the increased emphasis on MOA resulted in additional risk reduction through a concerted effort to address the threat of command file errors. The operations working group initiated and developed an operations process improvement effort, which first looked at the collection of command file error metrics and then the evaluation of the information identifying findings/themes that could be folded back into project and institutional operations. With a goal of continuing to reduce operational errors, the working group provided recommendations to be implemented during development and operations. In particular, evaluating the command file error statistics specific to each project led to several obvious conclusions. Experience strongly suggests that the command file error rates increase with launch and early cruise operations. This is particularly true when operations are going so smoothly, that it seems a simple effort to begin adding additional activities to the mission plan. Increased operational tempo around activities such as comet encounters and orbital operations often shows a spike in the error rate as a result of increased (real or perceived) pressure to succeed and tight timelines to prepare for very complicated activities. We see similar effects with the execution of first-time spacecraft activities. As projects approach periods of high operational activities, the chances of committing command file errors increase, and the projects should take steps to reduce this likelihood within the budget and schedule constraints of the operations. The analysis of command file error data collected led to changes in software, procedures, and processes for the overall operational uplink process used by JPL projects. The command file error rates over a decade indicate that the changes have been effective in reducing the errors and, consequently, the risk to flight projects. The modifications to the PRS to enhance data collection have permitted more effective feedback to the projects on command file errors and their causes to all projects. The Juno and GRAIL experiences validate the results of the working groups' efforts and show the importance of making the effort to fully implement their recommendations.

Appendix

Acronymn List

ATLO	Assembly, Test, Launch Operations
GEN	Genesis
GRAIL	Gravity Recovery And Interior Laboratory
ISA	Incident, Surprise, Anomaly report
JPL	Jet Propulsion Laboratory
MA	Mission Assurance
MCO	Mars Climate Orbiter
MER	Mars Exploration Rover
MGS	Mars Global Surveyor
MOA	Mission Operations Assurance
MOAM	Mission Operations Assurance Manager
MSL	Mars Science Laboratory
NuSTAR	Nuclear Spectroscopic Telescope Array
ODY	Mars 2001 Odyssey
PRS	Problem Reporting System
SDU	Stardust

Acknowledgments

The authors thank Dr. Philip Varghese, for his forsight and encouragement to the participants in this effort from its inception and continuing to the present both in his capacity as the manager of the Mission Management Office and later the Mars Odyssey and Mars Reconnaisance Orbiter project manager.

The authors thank Kyle W. Martin and the rest of the team at Lockheed Martin Space Systems Company for their proactive effort to identify, characterize, and mitigate command file errors since the program's inception in 2001.

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