Developing and Implementing a Process to Verify and Validate the NASA InSight Mission's Instrument Command Products

Nimisha Mittal
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
Nimisha.Mittal@jpl.nasa.gov

Kyle Cloutier
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
Kyle.D.Cloutier@jpl.nasa.gov

Abstract—**The NASA InSight Mars lander is a robotic spacecraft which is investigating the interior structure of Mars through a series of seismological measurements and experiments. Much of the commanding of activities performed by the instruments was done using 'blocks' – reusable functions written in Virtual Machine Language stored onboard the spacecraft– and sequences (series of commands) that were created by teams to operate their instruments. It was vital that these blocks and sequences (collectively called command products) be thoroughly examined prior to onboard execution using a comprehensive test program to reduce risk to the spacecraft and instruments and ensure smooth operations on the surface of Mars. The Verification and Validation (V&V) process that was designed and implemented during the months leading up to InSight's launch and landing successfully uncovered a number of issues that could have caused operational delays or impacted hardware. The process also helped adapt ground tools to better model onboard activities, create operational principles to avoid unintended consequences during flight, and generate a reference database for command product use during operations. This allowed InSight to meet its surface operations goals in the desired timeframe. This paper describes the development, implementation and results of InSight's command product V&V process, its contribution to mission success, and challenges faced.**

TABLE OF CONTENTS

1. INTRODUCTION

The InSight Mars lander is a robotic spacecraft which landed on Mars in November 2018, and is studying the interior structure of Mars through a series of seismological measurements and experiments. Its scientific payload includes a seismometer, a heat probe, and a suite of weather sensors. The primary mission goal is to understand the evolution of rocky planets by making measurements of Mars' tectonic levels, meteorite impacts and heat flow through the interior of the planet.

Spacecraft and Instrument overview: The Insight spacecraft is based on NASA's 2007 Phoenix mission, which landed near Mars' North Pole in 2008. The main spacecraft bus which includes all the engineering subsystems such as C&DH, telecommunications, thermal, power was built by Lockheed Martin Space Systems (LMSS), located in Denver, Colorado. The seismometer instrument (SEIS) was contributed by Centre National d'Études Spatiales (CNES) and its partners. The Heat Flow and Physical Properties Probe (HP3) was built and operated by Deutsches Zentrum für Luft- und Raumfahrt (DLR). The Auxiliary Payload Sensor Suite (APSS), a collection of pressure, wind, temperature, and magnetometer sensors was provided by various institutions – including the University of California Los Angeles, and Centro de Astrobiologia (CAB).

Mission Overview: Insight landed on the Martian surface, close to the equator, then used a robotic arm to deploy the seismometer and heat probe from the lander deck to strategic locations near the lander and perform checkouts within the first 90 days of the mission. Its weather sensors take daily measurements to help correlate the seismometer measurements. The telecom antenna is used to perform experiments as well. The mission will collect measurements on the planet for at least two earth years.

Operations and commanding: All spacecraft activities are primarily accomplished via commanding sent from ground operators to the various instruments and sub systems. During the first 90 days, commands were built and sent to the spacecraft a few times a week, after which there was a gradual transition to once a week planning. Commands for most activities were built and tested prior to the launch and landing of the spacecraft in order to help accomplish the mission's deployment goals in the allocated timeframe.

Purpose/need for V&V of products

The need for InSight's command product V&V process stemmed from one of its Level 3 mission operations system (MOS) requirements: The MOS shall be able to verify and validate any science/instrument sequence or real-time command during development and flight operations. All command products were independently created and tested by their respective institutions prior to their use in the testbed and on the spacecraft. However, given the varying fidelities of test environments available to each institution, and the distributed nature of the development of various parts of the software, it was necessary to develop and implement a standalone verification and validation (V&V) process such that all products were subjected to consistent checks in testbed venues of appropriate fidelities. Early InSight operations needed to be as streamlined as possible due to its strict deployment schedule. By preparing and testing all known command products in advance, only minimal changes were needed during actual operations.

A comprehensive test program ensured that the products posed minimal risk to spacecraft and instrument health and safety by 1) confirming that they were performing activities as intended (for all known use cases), and 2) verifying compatibility with flight software and ground tools.

V&V ensured the products were created in the correct formats, flight rules were met, and information related to the product was well documented (description, resources, constraints) which would later help the operations team during activity implementation.

2. COMMANDING TERMINOLOGY

VML language

Virtual Machine Language (VML) is a human readable text based language used to create InSight's command products. These are translated to binary files by a VML compiler, which can be interpreted by the onboard flight software.[1]

VML is a versatile language and allows for flexibility in commanding. VML allows the user to take advantage of global and local variables, timeouts and wait statements, and call new functions to run sequentially or in parallel. It allows the addition of logic with various outcomes depending on the conditions satisfied. While VML excels in providing ease in commanding and operability, it increases difficulty in tactically validating and understanding sequence function and path, thereby adding complexity to the V&V effort. The V&V process therefore included steps to model all the block and sequence branches, the correctness of global variables used, the impacts of timeouts, and any consequences of parallelism introduced in commanding.

Command products – blocks and sequences

Command products are a series of instructions to a flight or instrument software component to perform a particular function. On InSight, these products are defined as two constructs: Blocks and Sequences. At the highest level, blocks are reusable functions stored onboard the lander that can accept parameters in order to perform functions based on the coded logic, whereas sequences can be new or reusable, and can pass arguments to blocks used within them. Both blocks and sequences contain VML statements and spacecraft commands. Blocks, sequences, and sequence templates (defined below) were tested via the V&V process.

Blocks are intended to accomplish repeated functions in an instrument or subsystem. Blocks are part of block libraries and are stored onboard the spacecraft, thereby making those products quickly accessible and eliminating the need for the operations team to create and send the products to the lander every day (which adds time and effort on the ground to thoroughly review the products and also takes up radiation time in the uplink window). Due to their reusable nature, blocks are relatively timed. Examples include: a block to power the electronics board within an instrument on or off.

Blocks are used by all major systems on the spacecraft. Engineering system and instrument deployment system blocks were created and validated by LMSS and the JPL Instrument Deployment teams respectively. On the instrument side, roughly 100 blocks created by the SEIS, HP3 and APSS teams are distributed across three payload libraries which reside in the lander file system. Any change to a single block requires an entire block library to be updated on the lander. A small number of infrastructure blocks, used for sequence management, were also created and tested by the JPL team.

Sequences can be absolute or relative timed. If they are relative timed, they can be reusable. They can also be created depending on the needs of a particular day. For example, a sequence could include a call to the block which powers on SEIS, followed by a command to change its data collection rate to a particular value, followed by a call to the SEIS tilt measurement block, followed by a SEIS power off block. Many of these functions (power on, tilt measurement, power off) are separate blocks that are stored in the onboard block libraries, but the sequence can be tactically created by the operations teams on a particular sol. (Note: Such sequences can also be stored onboard if repeated frequently. However, sequence storage space onboard is limited, so careful sequence management is sometimes needed).

Sequence templates are sequences which allow certain fields and arguments to be updated by ground operators based on the needs of a particular day. For example, a sequence template for SEIS power on could allow a user to update the side of the instrument to power on (SIDE 1 or SIDE 2), the primary or redundant switch, select the sensors etc. All anticipated sequence templates (with iterations of inputs) were created and validated in advance, which allowed expedited sequence creation during operations.

Sequence engines

Engines are virtual machines onboard the spacecraft which store and execute sequences and blocks. Engines are a way for a spacecraft to allow and constrain the number of parallel processes being commanded. InSight contains 24 engines, which include three to store the payload block libraries, and one for each of science instruments to execute their respective activities. Each engine is also configured to have its own special fault response, depending on the severity of impact on the overall system.

The V&V process needed to ensure all command products were designed to run only in their designated engines, there were no engine collisions (i.e. initiating execution of commands on an engine that was already performing another task), and products were sized correctly to load and execute on engines given any constraints.

FSW interactions

The main flight software (FSW) on the lander has separate modules that interact with each of the instruments. Commands from the ground are received and processed by these instrument modules, which either perform an action based on the command, or pass them directly to the instrument internal software for further processing. The V&V process performed end-to-end checking of these interactions between flight-like commanding, FSW's response to those commands, and proper receipt of the commands by the instrument by testing command products on testbeds which modeled these interactions.

Sequence architecture

On InSight, commanding for each spacecraft wakeup period is done via master sequences. Master sequences define the time available to command instrument, arm, or other engineering activities. The master activates a submaster, which contains calls to various instrument sequences and blocks, which may run in series or in parallel. Absolutely timed cleanup sequences unloaded certain engines that still had activities executing, sometimes leading to an instrument safing response (an onboard fault response which marks an instrument unavailable for use until ground operators can investigate the issue). The sequence architecture influenced the design of test cases for command product V&V. The test cases accounted for corner cases, sequence cut off times, as well as activity interactions.

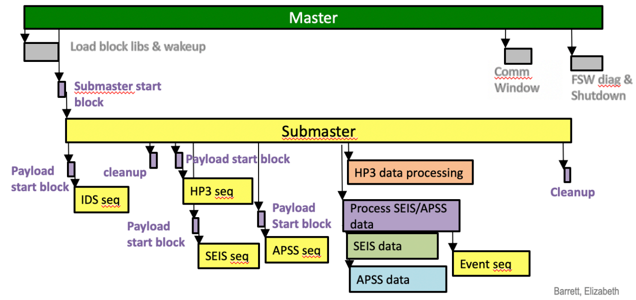

Figure 1: InSight Sequence Architecture [1]

3. DEVELOPING THE V&V PROCESS

Defining Goals

The driving requirement for this task was to ensure all command products are subjected to a V&V process, therefore a set of testing goals were baselined in order to define the scope of the work.

The main goal of the V&V process was to ensure that all blocks and sequences were tested in as close to a flight like environment as possible, in order to prove that the products were performing their intended functions. This required testing and characterizing all failure paths, not typically seen in other project testing such as ATLO (Assembly Test and Launch Operations) or ORTs (Operation Readiness Tests) during which only the nominal paths in sequences/blocks were run. Some secondary goals that were directly related to developing and implementing this process included 1) ensuring ground tools were modeling the products correctly and 2) creating a database which could track testing progress and serve as a repository for detailed documentation on the use of the products.

Defining complexity and priorities

The process had to be developed keeping in mind the number and complexity of products that needed to be tested, and the mission timeline.

Complexity analysis: at the start of the V&V effort, an initial analysis was conducted to characterize the complexity of all the products developed till that point (i.e. blocks). Points were given to each product based on the number of criteria each of them met – for example, the number of logical paths they contained, arguments, the use of special VML statements such as timeouts or onboard construction of commands, lines of code, parent/child blocks, and other features that distinguished them from other products. These points were then used to categorize the products as low, medium and high complexity using scores. The analysis showed that roughly 1/2 of the products had low complexity, less than 1/3 had high complexity, while the remaining were in the middle. This helped define a tentative schedule for testing. Table 1 shows information on the blocks that were available at the time of the analysis. Others were added/removed later.

Table 1: Command Product Complexity Analysis

Complexity	Lines of code	No. of inputs	Conditionals	Special cmds	No. of blocks:
5 - Highest	>250	High	High	High	4
4 - High	>50	High	High	High	27
3 - Medium	50-100	Some	Some	Some	25
2 - Low	<50	Low	Low	Low	26
1-Lowest	<50	None	None	None	33

This analysis was later used by the FSW team to help distribute the blocks into payload libraries, when it became evident that highly complex blocks also look longer to load into FSW, and it was important to maintain equivalent load times across the libraries.

Prioritization: Due to the limited time available, and the overlap in product development and testing, careful prioritization of products and testing was required. All cruise command products were delivered and tested prior to launch. Products developed for use on the Martian surface were given priorities based on their earliest use and criticality in achieving mission objectives.

Products were tested in order of highest priorities:

- Highest priority: Products used during the cruise phase, during surface deployment (first 90 days), high probability contingency products
- Medium priority: Science monitoring products (used in later routine phases of the mission)
- Low priority: Infrequently used products for non-critical activities, low probability contingency events

Using this method, it was determined that roughly ½ of the products (~60) planned to be developed would be used for the first time during the cruise phase of the mission, and the remaining would first be used on the surface. Later, the number of surface products developed doubled as the teams became more familiar with the daily operational constraints and requirements.

Designing the process

Development of the V&V process was iterative and required an understanding of the resources available, and the kinds of issues detected by the different tools available for use.

The Test All Method: Two tools were already available to the V&V team to use with minimal training and setup.

i) SEQGEN: An in-house JPL heritage tool that is adapted for different missions, and enables expansion, modeling and constraint checking of spacecraft commands [2]

ii) OLVM: "OffLine Virtual Machine" – this is an LMSS tool which represents the sequence expansion portion of the VM flight software that is executing onboard the spacecraft, wrapped with a user interface for use on the ground.[3]

An initial method was developed to quickly test all blocks using Seqgen and OLVM with a set of representative parameters that matched the argument types. This gave the team a quick look at the most common issues, and understand the benefits and limitations of using these tools in the V&V process.

This quick and simple check caught syntax errors, type coercions, typos, and out-of-range inputs, and allowed for quick turnaround times for updates. While higher fidelity testbeds mimic the actions of the spacecraft and flight software more closely, the simplicity of Seqgen and OLVM allowed the V&V team the ability to manipulate the path the sequence proceeded down. Unlike with higher fidelity testbed venues, the team could explore all possible paths and test edge case scenarios. More importantly, using Seqgen early in the V&V process allowed the blocks to be tested in the environment in which they would be run in the tactical setting during real time operations. This was advantageous for Seqgen and tactical tool/process validation as well — bugs in the tactical tools were found and fixed to better suit the needs of the blocks and sequences.

Automating VML rules and guidelines: A set of rules and guidelines for writing command products in VML had been compiled by the InSight team over the course of the development and testing of products created for other project activities. For instance, the rules included proper labeling of command products, being consistent with variable types, proper syntax for the use of global variables among many others. These rules were originally checked manually, which was time consuming and prone to human errors. As such, an exercise was conducted to categorize the rules into required and optional (by characterizing the consequences of violating them). The rules that were redundantly checked by Seqgen during the process were removed from the list. A python script was created to check all the rules that could be automated. There were a small set of rules that could not be automated and thus were checked manually - for example, checking if the off-nominal path of a block should continue execution, which was an evaluation only an instrument expert could make.

Test Venues

Ideally, all products are best tested in the highest fidelity test environment available. This is typically the flight vehicle itself; however, access to the vehicle is limited during its development due to other higher priority activities it is needed for. More importantly, extra care would be needed in designing tests to be sure no damage is caused to the flight vehicle. As such, testing is conducted in other environments, such as engineering models of the spacecraft and instruments, which mimic most of the functions of the full spacecraft with some exceptions.

Higher fidelity test venues are shared with other teams on the project – such as the deployment team, fault protection etc. Therefore, it was important to understand what would be gained (or lost) by executing the command products through the higher fidelity test environments in order to limit the tests to only the most essential ones. To do this, the team created some mock tests to execute delivered products through the test environments. The results helped the team define issues that were uniquely flagged by the higher fidelity venues, and those that were also discovered by other tools such as Seqgen and OLVM. Existing test reports were also reviewed – which alerted the team to the fact that results could vary on the testbed compared to the flight vehicle due to different configurations and hardware/simulator differences.

Information was compiled through existing documents and trials of the tools to better understand all their advantages and constraints, which helped define how they should be incorporated into the V&V process.

Table 2: Testbed Venues

Tool/Testbed Venue	Pros	Cons	Command products
VML check script	Checks VML rules	Some checks manual	All blocks/templates (all paths)
SEQGEN	Type coercion/syntax, flight-like validation, verbose logic expansion, expose gaps in seqgen adaptation	Adaptation ongoing	All blocks/templates (all paths)
OLVM	Command validity, Engine utilization/conflicts	No subsystem simulation No FP/telemetry modeling	All blocks/templates (all paths)
SoftSIM (Software simulator)	Interaction with lander and instrument FSW/telemetry Realistic duration modeling	Uses instrument SIMs, not EM	All blocks/templates (subset of paths)
STL + EM (S/C test lab + instrument engineering model)	Interaction with lander and instrument FSW/ using instrument EM (most flight like interface)	Not always available	Subset of blocks

Expert review: The above methods helped with the 'verification' part of the testing requirement – i.e. making sure products were executing without errors. The 'validation' of products – i.e. ensuring the correct products were built for their intended functions - required involvement of various instrument and FSW experts who were most familiar with the instrument and subtle command and FSW interactions. As such, a review of the command products early in the V&V process was instituted during which the product was inspected, its use cases discussed, and any special test cases documented. Reviews were also held separately with the fault protection team to ensure that any major off-nominal scenarios triggered by instrument products could be handled by the spacecraft.

The V&V process

With all these considerations, the V&V process for command products was categorized into 3 main parts:

a. Kickoff – product walkthrough with experts, collection of information for lower and higher fidelity V&V

b. Initial/lower fidelity V&V: the test-all OLVM/Seqgen method, VML checker script

c. Higher fidelity V&V: Test scenario development and implementation

The process was expanded into the following steps:

- Initial V&V – an internal review of the command product was performed within the JPL team. The V&V team checked VML rules and guidelines (manual and with helper script), designed test cases to ensure maximum path coverage in Seqgen and OLVM. Reports were posted to JIRA. The instrument team also performed checks using their in-house simulators and posted reports. This helped weed out several problems early in the process, which enabled better success of the product during higher fidelity testing.

- Product kickoff – the command product was discussed in detail during a teleconference with the instrument and FSW experts. Existing test reports were reviewed, and cases for higher fidelity testing were proposed and discussed, expected test-as-you-fly exceptions noted, and the scope of the work identified.

- Higher fidelity V&V: SoftSIM and STL testing – Test cases were designed to execute in higher fidelity venues such as SoftSIM and instrument engineering models (EMs). Since instrument simulation capability on SoftSIM was limited, some commanding was only properly modeled on the EMs. Some test cases required testbeds to be specially configured. At a minimum, all nominal paths as well as off-nominal paths that contained instrument specific commands were executed on the instrument EMs. All other paths were tested in SoftSIM, Seqgen and OLVM, which had sufficient fidelity to reliably identify issues.

- Review of test reports – results of the test cases were reviewed internally. Any questions or detailed reports from the highest fidelity venues were sent to instrument experts for comment.

- Repeat testing if needed – More often than not, testing had to be repeated when a product was updated due to an issue that was discovered, or if the test did not run correctly, or if a new version of the Seqgen adaptation code was released.

- Complete documentation – All test information was documented in a database created specially to track V&V.

- Final approval – When all necessary steps were completed, the product was approved and delivered to a write-protected official flight product repository.

Logistics

Finally, infrastructure was needed to execute the V&V process for multiple versions of over 200 command products. A secure location was created on a server for teams to deliver their products using a defined naming convention, which helped with configuration management (making sure the products being delivered and tested had proper version control). All test reports were labelled with information on the product version, tool versions, and FSW versions running on the testbeds.

A payload block library update infrastructure was created for the V&V process in order to ingest new or updated blocks for testing. This was critical because it simplified the team's ability to test blocks in higher fidelity simulators in a flight-like way, and keep track of the latest block versions. Without this infrastructure, incorporating updated blocks in simulator tests would have been time-consuming and error-prone.

The need for a database to store all command product use and test related information was realized. This database needed to be accessible to JPL as well as foreign national teams, and be configurable to meet the needs of the project and the test program. After some research, the team settled on JIRA, an Atlassian Tool typically used for task tracking which had flexible features allowing its use as a database. It allowed searching for various products, making parent/child links, labelling products using easily searchable keywords, tracking the status of testing, and making quick reports on the overall status of products.

In addition to infrastructure setup, the team also started training in the use of the testbed simulators available, and setting up required account access.

4. IMPLEMENTING THE V&V PROCESS

Performing V&V

Given InSight's relatively short command product V&V development and implementation timeframe, V&V was started while the process was still being developed. As products were updated, they were circulated through parts of the process repeatedly, which helped increase the overall confidence level in the products, and made the V&V process more streamlined. Due to the magnitude of the task, it was helpful for different team members to focus on separate instruments, since each instrument team had a distinct style of writing command products and operating its instrument.

Generally, V&V was started on 3-5 command products per week. These products were then moved through various parts of the process, and took up to 4 weeks to complete. Therefore, at any given time, up to 20 products could be in the V&V pipeline. The timeline for highly complex products was much longer. The number of products being tested per week also varied when products were updated and had to partially repeat testing.

Test Case Design

Selecting and designing test cases was one of the most crucial and time-consuming part of the V&V process. A system was created to ensure all required test cases could be covered. Test sequences were first created to execute most block and sequence paths, with the correct initial conditions. These conditions were then changed to some likely off-nominal scenarios (eg. component not being fully powered, or the instrument being in an unexpected state) to test if the product (or corresponding instrument FSW) was robust to the errors. Other test cases requests were sometimes made by instrument operations experts.

Difference in testing for blocks vs sequences

Slightly different methodologies had to be adopted for testing the different command products.

Blocks perform the bulk of the logic processes and decision making. Therefore, during block V&V, the focus was primarily on testing various logical paths. Blocks were also largely tested standalone – i.e. block interactions could not be tested until the sequences using them had been created, unless the blocks had a parent-child relationship (blocks within blocks).

Sequences are typically a wrapper around calls to multiple blocks. This meant that sequences quite often had no logic implemented in them and thus testing was straight forward and single path. When sequences relied upon the return value of a block, or a value of a global variable set by the spacecraft, test sequences could be manipulated to adequately test all possible scenarios. However, there were instances when blocks used within sequences had to be updated due to an unforeseen interaction that was discovered during sequence V&V.

Testing sequence templates required significantly more setup. Several versions of a sequence (with various inputs, representative parameters, and conditions) had to be created and compiled separately, which increased the number of test runs and reports needed.

Schedule

Creation of command products for InSight was an iterative process; as products were run through the V&V process, issues were discovered, which required the products to be updated. Therefore, a significant amount of re-testing of products was typical. Repetition of various steps of the V&V process depended on the change – if changes were made to actual instrument commands and parameters for instance, the product would be exercised through higher fidelity venues; however, if the change was minor, and only made to a non-essential message or VML statement, lower fidelity tools were deemed sufficient.

Blocks were defined early, while sequences were created later in the process when the daily deployment timeline was finalized. The chart below shows the actual product creation and testing schedule. V&V was completed on all cruise products prior to InSight's launch, and surface products prior to landing on Mars.

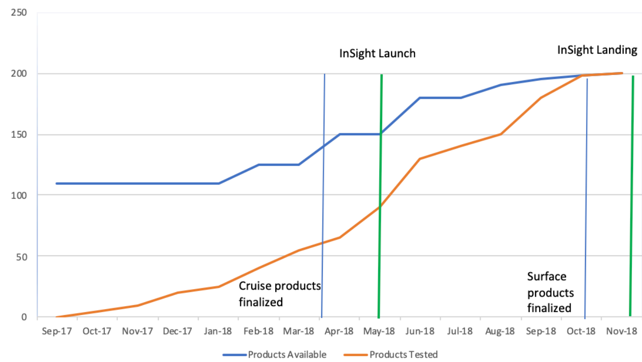

Figure 2: Product Development/Testing Schedule

Challenges

Implementing the V&V process came with a set of challenges. Most command products were unique and required significant effort to test at the unit and system level. There were also budget and personnel limitations due to the need to balance the priority of this task with other higher priority mission activities that were occurring in parallel before launch and landing events. As a result, the team remained small; however, more support was added once products were ready to be tested and personnel completed other tasks.

At the time V&V started, many of the tools and rules were still under development, which added the number of times the products had to be checked as newer versions became available. Due to the distributed nature of the mission, a complete testbed (which included both engineering and instrument hardware and software) was not available in a single location, therefore the testing had to be conducted in pieces in order to cover all interfaces. Having a distributed team also meant that instrument expertise was dispersed across several time zones, which made communication challenging. Although weekly teleconferences with instrument experts were held to discuss details related to the execution and progress of the V&V process, a limited number of Technical Interchange Meetings (TIMs) which were organized to discuss product development and testing and meetings with team members during other scheduled project activities were very beneficial in resolving issues and keeping the schedule.

5. RESULTS

Implementation of the process was a success. The process as a whole ensured the primary and secondary goals of V&V were being met. Testing confirmed that all the logic within the products was verified, command products were compiled correctly into final spacecraft files, compliant with flight rules, all nominal and off-nominal paths in the command products were exercised in flight like test environments, and all required documentation of the products was complete for future reference.

Issues Found

The V&V effort proved to be useful in identifying several issues which ranged from minor ones to higher impact issues which had the potential of causing onboard hardware damage.

The most common issues found included syntax errors, use of inconsistent variable types across products, flight rule violations, incomplete logic (i.e. return paths from child blocks not being checked at the parent block level). Most of these issues, if not fixed, could have resulted in the products not passing the daily checks performed by the operations team prior to building final spacecraft products, or performing the intended function onboard, resulting in operational delays.

In some cases, command products were found to have incorrect parameters which could have resulted in instrument safing. Other issues such as incorrect spacing of commands or incorrect ending states in contingency products had the potential to cause hardware damage but were resolved.

The process inherently checked expected instrument/spacecraft state before initialization, which helped discover instances when certain global variables were not configured properly.

Seqgen modeling

Seqgen served as the tactical V&V tool, therefore, it was important that its modeling of products be consistent with onboard behavior. The V&V effort helped fix certain modeling issues found in the Seqgen tool related to flight rule compliance checks. Actual data from higher fidelity test runs helped provide information on how some of the more complex instrument activities should be modeled in terms of logical paths and duration.

JIRA

Test progress and command product details were tracked in the JIRA database setup for this task. The V&V effort required collection of information such as command product descriptions, use cases, activity pre-requisites, applicable phases of missions, durations, constraints, flight rules. Since most of this information was relevant to the surface operations team, it became the official reference database for the team during planning operations. JIRA was referenced whenever a new product was used in flight, or interactions between products needed to be verified, or a response to a particular test case reviewed.

Out of scope checks

It is important to identify what aspects were not covered explicitly by the V&V process. The command product V&V process was not responsible for verifying data quality, correctness of telemetry being updated nominally, and characterization of power or data volume resources. Other tests, such as thread tests, operational readiness tests, and ATLO verified those aspects.

6. CONCLUSIONS

Lessons Learned

The V&V process designed for InSight command products was a success – nearly all tactical ground tool issues were discovered and resolved prior to product use in real operations. Onboard block and sequence execution were also successful during the deployment and science monitoring phases of the mission. However, some lessons would benefit future efforts to perform V&V on similar discovery class missions.

i. Any rules and guidelines for product development should be characterized early and shared with teams building the products. Implementing basic rules after products have been created leads to delays in updates and tests.

ii. Experience in instrument commanding on flight hardware and testbeds would help with the V&V effort – if personnel on the V&V team are involved in instrument and testbed testing early on, the steep learning curve to perform V&V would be greatly simplified.

iii. Out of scope testing should be defined early in the process. Projects should decide how much risk is acceptable given any staffing and schedule constraints

iv. As products were tested and issues discovered, updates were made, and the products re-tested. However, not all issues were discovered at once; it was common for teams to change commanding strategies, or discover missing elements in products after the completion of most testing. While updates were usually justified, they significantly increased the V&V effort. As a result, products were often not finalized until the last possible product submission deadlines. To avoid this, sufficient margin should be added to product submission deadlines, and only the most crucial updates entertained after those deadlines pass.

Evolution of the V&V process post landing

The rigorous V&V effort prior to landing aided in the team's ability to modify or create new blocks and sequences to suit tactical needs once on the surface. The V&V team and instrument teams' knowledge and experience gained prior to landing was a positive attribute to skillfully and proficiently operating on the surface. Off-nominal scenarios and new surface activities were able to be quickly tested and accommodated into tactical planning.

Future improvements

The V&V process could benefit from more automation - starting from delivery of a new product version to basic checks through lower fidelity tools. Better regression tests, if setup early in the V&V process, could greatly save time and effort as products are updated.

Although JIRA served its purpose as an easily accessible database for tracking V&V details, it was cumbersome to update hundreds of individual products with the frequent changes. A smarter database that can detect new reports and test runs and automatically be updated would be a significant improvement for products of this scale.

ACKNOWLEDGEMENTS

This work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration. Sincere thanks to Jared Call for setting up the test-all method, Shaheer Khan for developing the VML checker script, Elizabeth Barrett for JIRA updates, reporting, and HP3 V&V, the SI-OPS, SEIS, HP3 and APSS operations team for extensive help in implementing the V&V process, , the JPL testbed team for setting up and running numerous tests, the LMSS systems and FSW teams for advice and help in completing V&V on schedule, especially Brian Rak for modeling the most obtuse SEIS recentering block paths, and the JPL CAE team for JIRA customization. Thanks also to Julia Bell and Jaime Singer for sharing their block V&V experiences.

REFERENCES

[1] Virtual Machine Language (VML) v2.0 Users Guide, Issue Date: April. 29, 2005 JPL D-28342A

[2] Bliss, D., Morales, L., Reduce Costs with Multimission Sequencing and a Multimission Operations System, https://descanso.jpl.nasa.gov/RCSGSO/Paper/A0023Paper.pdf

[3] Verma, V., Jonsson, A., Simmons, R., Estlin, T.Survey of Command Execution Systems for NASA Spacecraft and Robots, https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20050240167.pdf

BIOGRAPHY

Nimisha Mittal *is the Instrument command product V&V lead on InSight. She has previously worked on the operations teams for the Mars Exploration rovers and Cassini missions, and payload systems on the Mars Science Laboratory mission. She holds a BS in Aerospace Engineering from the University of Texas at Austin and an MS from Stanford University. Currently she is working as a flight operations development and V&V engineer on the NISAR mission, a collaboration between NASA and ISRO.*

Kyle Cloutier *is a Sequencing Integration Engineer for the InSight Mission. She has previously worked on the Mars Exploration Rovers (MER) Mission and the Cassini Mission, and is currently a Payload Systems Engineer with the Surface Water and Ocean Topography (SWOT) and Sentinel-6 Missions. She has a BS in Aerospace Engineering from the University of Maryland.*