Six Years and 184 Tickets: The Vast Scope of the Mars Science Laboratory's Ultimate Flight Software Release

Alexandra Holloway, Jonathan Denison, Neel Patel, Mark Maimone, Arturo Rankin
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109
alexandra.holloway@jpl.nasa.gov

Abstract—**The Mars Science Laboratory (MSL) Curiosity rover is about to receive its sixth and likely final complete flight software update after having operated on Mars for more than a decade. Software transitions on MSL provide an opportunity to add or replace functionality, fix bugs, and prepare for future capabilities. The penultimate full software release, R12, was installed on Curiosity in 2015, three years after its August 2012 landing, and was followed over the subsequent seven years by many patches as engineers worked to address new mission constraints quickly. Because each additional patch increases the complexity of maintaining and operating the rover, a new flight software update called R13 was proposed, which aimed to make operations more straightforward by incorporating existing patches, improved software capabilities, and new software capabilities into a single monolithic rover flight software image. The R13 development effort kicked off in early 2017. Over the next six years, the scope of R13 expanded to include many desired capabilities and bug fixes – some of which were proposed even earlier than 2015 but were unable to be implemented in R12. Overall, the MSL Change Control Board approved 56 bug fixes and 53 new features for R13 development. Twenty-seven developers implemented these changes over a 3.5-year period. Following a 2.25-year testing campaign, R13 was approved for use in flight onboard Curiosity. In this paper, we detail the path of the R13 flight software release from its proposal in April 2016 to its approval for use in flight in September 2022.**

TABLE OF CONTENTS

1. INTRODUCTION

The Mars Science Laboratory (MSL) Curiosity rover has been operating on the surface of Mars since August 2012. Between landing and the installation of flight software (FSW) version release number 12 (R12) in January 2015, Curiosity's flight software had been updated with five full releases, two cold patches, and over 52,000 hot patch installations [1]. One lesson learned from the Mars Exploration Rover (MER) mission was to expect flight software updates throughout the lifetime of a rover mission. With Curiosity's potential for years of future operations, a significant number of patches were anticipated in Curiosity's future. In fact, within 15 months of the transition to R12, six additional patches were under consideration related to ChemCam sun safety, terrain adaptive speed control to reduce wheel wear [2], Non-volatile Parameter Memory (NPM) wear leveling, visual odometry thinking while driving [3], visual odometry drive step truncation, and robotic arm inverse kinematics [4].

978-1-6654-9032-0/23/$31.00

Software upgrades for the MSL rover take one of three forms: hot patch, cold patch, and full update [5]. Table 1 summarizes these options. Although releasing a full image for installation on the vehicle would roll many pending improvements into a single, cohesive binary, the development, test, and installation process for full flight software release is often prohibitively expensive. Instead, a patch release targets specific issues and can be deployed quickly, until a more comprehensive release can be achieved.

Patching is the act of modifying compiled flight software to add or replace functionality, fix bugs, and prepare the system for future use cases. A hot patch is optionally installed on every computer boot-up; a cold patch is installed once and permanently modifies the software image so every boot following contains the change. Each patch must be stored as a separate file in the rover's file structure, thus introducing a new operations challenge: it becomes more and more cumbersome to keep the ground version of flight software in sync with the onboard version. Patches are necessarily CPU-specific, hence patches designed for the flight CPU (a RAD750) cannot be applied to versions of flight software running on ground workstations (PCs). As a result, each added patch increases the complexity of maintaining and operating the vehicle.

A hot patch contains one or more individual "pokes," or memory modifications of the currently-loaded FSW in RAM, sometimes with additional compiled code, and/or a VxWorks shell script to install the modifications. The hot patch is the most common method of patching on MSL as it is quick to author, review, test, and fly. However, a hot patch must be installed onto the rover computer every boot, since it works by modifying volatile memory. That said, if a hot patch is not desired for a particular boot, the installation script can be omitted and the hot patch will not be applied. Hot patches also have a particular benefit in that in the event of a vehicle Safing event, the FSW will avoid applying them when booting into safe mode. Therefore, any problems that might have occurred because of a hot patch would not be present in the rebooted system. The Viking I mission ended when a software patch inadvertently (and permanently) overwrote flight software critical for communicating with Earth [6], so the benefit that hot patches aren't applied when booting into

Table 1. Main software upgrade methods used on MSL

Type	Description
Hot patch	One or more individual memory modifications of the currently-loaded FSW in RAM, sometimes with additional compiled code, and/or a VxWorks shell script to install the modifications. Since only RAM is modified, this change only persists for the current boot cycle.
Cold patch	Applied like a hot patch, but ultimately also burns the resulting modified image back into non-volatile memory. Unlike a Hot patch, cold patches are permanent changes that will automatically persist into all future reboots.
Full update	Entire new image uplinked and burned into non-volatile memory.

safe mode is an important consideration when evaluating the mission impact of a new hot patch. MSL rover flight software has been the recipient of 13 hot patches, installed over 57,000 times as of October 2022 (sol 3610).

The cold patch is similar to a hot patch, in that its pokes modify volatile RAM memory; but unlike a hot patch, the resulting binary code is burned back into non-volatile memory. This allows the patch to remain installed across boots. A cold patch can be undone with another cold patch. Three cold patches have been installed on MSL as of October 2022 (sol 3610).

Finally, a full flight software update collects improvements across all FSW modules. The entire codebase is recompiled and a new FSW image is uplinked and burned into non-volatile memory. On MSL, full updates have been expensive processes requiring significant time and personnel to support the work; not only proposing and developing each individual change, but also testing them in the most appropriate venue during development, regression testing, and overall Verification and Validation (V&V) efforts. Many development and testing experts who completed earlier updates in less than a year had already left the project, making it more challenging for the current team to update procedures to match the new capabilities. For example, the most recent full flight software upgrade, made only on the backup CPU in December 2020 (sol 2963), completed in a blazingly fast 19 months [7] [8]. MSL has seen six full rover flight software upgrades in total as of October 2022 (sol 3610).

Improved Patching Capability

Not long after the release of R12, additional patches became necessary to address growing operational needs and changing hardware and software limitations. And the flight software team identified several issues that made it difficult to manage patches.

- Some hot patches are brittle by nature. For example, some patches only work in the release for which they are developed, and not in sandboxes where development occurs, complicating future patch development.
- Most hot and cold patches cannot be unit tested easily because they are, by design, not integrated into the full flight software monolith; therefore, the patches will always remain separate from the flight software source code used to build unit tests.
- Multiple patches working simultaneously create exponential parallelism challenges for testing, because patches must be tested in various ops-realistic combinations.
- Small patches are easily accommodated, but larger ones reduce our limited available VxWorks heap space significantly.
- The Mobility component is built into the flight software image and cannot be removed, and any new full component update would be too large to fit in the remaining memory.
- There was no fixed size enforcement of flight software module binaries, eliminating the possibility of creating small compiled patch deltas.
- There was no uniform process for patch creation and management, resulting in multiple places where mission-ending mistakes were possible.

2. CURIOSITY FLIGHT SOFTWARE HISTORY

Figure 1 illustrates Curiosity's traverse, starting with its landing in August 2012 on the left and following the rover's path to the right [9]. Flight software releases along its path are marked in orange for full release and blue for a cold patch. Hot patches are not shown. The software upgrades are marked with the Martian day (sol) on which the software was applied.

MSL landed with flight software version R9.4.7, and quickly switched out cruise and entry-descent-landing capabilities for surface software with R10 on the fifth sol of operation (R10.5.7, August 2012). The next update occurred as a cold patch on sol 217 (R10.5.8, March 2013) following a major in-flight anomaly [10], and a full flight software release was installed on sol 264 (R10.6.4, May 2013).

The next major flight software upgrade (R11) was installed on sol 446 (R11.0.4, November 2013). R11 included many maintenance and capability updates, including a permanent FSW change to incorporate temperature-dependent engineering camera models to mitigate problems experienced with using the backup Navigation Cameras (NavCams) [11]. Another in-flight anomaly was addressed with a cold patch on sol 772 (R11.0.5, October 2014) and was followed by a full update with flight software release (R12) on sol 875 (R12.0.3, January 2015).

In 2016 (sol 1389), the rover experienced a file-system related anomaly in-flight. In response, the team patched the flight software on sol 2808 (R12.0.4, June 2020) to correct this issue and address an issue seen in another spacecraft which shares part of the code base [12]. During this time, MSL's backup computer experienced an unrelated in-flight hardware anomaly with failed non-volatile memory. As a result, the R12.0.4 cold-patch was applied only to the prime computer, as the backup computer's memory was understood to be untrustworthy and unusable. The project mounted a separate effort to restore the backup computer to "lifeboat" status, so it could be used in the event of a swap. The effort was called R-Hope, combining the nomenclature "R" for release and "Hope" for the team's desire this should work. R-Hope was installed on sol 2960 (December 2020) only for the backup computer. At this point, the primary computer contained cold-patched R12 and the backup computer contained R-Hope.

When R12 was first deployed on sol 875, the rover had driven 9.8 km and had climbed just 62 meters above its landing site elevation. From sol 875 through sol 3610 (October 2022),

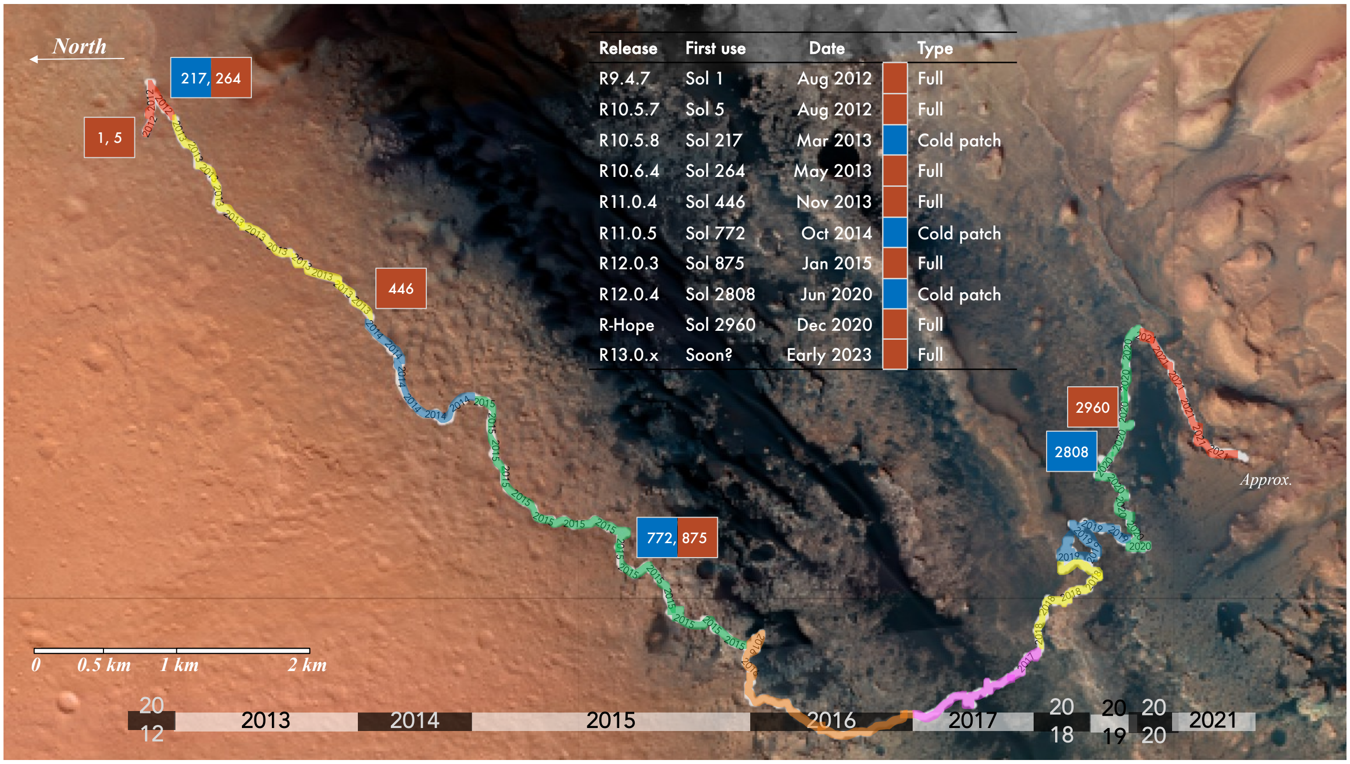

Figure 1. Flight software releases during the rover's traverse, 2012 to 2021. More than six years have elapsed between R12, the last full flight software release for both prime and backup computers on sol 875, and the expected R13 release date. Full releases (orange); cold patches (blue); hot patches not shown. The colors on the dots indicating rover location are not meaningful, they just serve to indicate distinct Earth calendar years, which are labeled in the legend at the bottom of the figure.

MSL drove an additional 19.2 km. As of sol 3610, the rover has driven a total of 28.998 km and climbed 692 meters above the landing site, exceeding the pre-R12 elevation by 630 meters as the rover makes its way up the side of Mount Sharp.

3. Evolution of R13

In April 2016, the flight software team proposed to the project the development of a full release (called R13) with the primary motivation to make improvements to the flight software such that future patches would be less onerous to implement and safer to install. To accomplish that, functions and commands would be written to aid future development of patches [1]. Another benefit to an R13 release would be eliminating the existing patches by merging them directly into the R13 flight software build. These improvements would reduce risk by enabling the project to respond to future flight software problems more quickly and effectively.

The flight software team's proposal emphasized that our ability to perform flight software updates was diminishing due to the expectation that many experienced flight software developers and V&V experienced personnel would leave MSL for the Mars 2020 Perseverance mission (M2020) by the end of 2016. By starting an R13 prior to that, it could provide valuable training for the Engineering Operations (EO) team, which supports strategic analysis and development as well as tactical mission downlink operations. Another benefit of an R13 was that it would provide an opportunity to consider changes that would result in more science return in less planning time. Each subsystem could propose flight software changes, and each approved change that was implemented in R13 could feed forward into M2020 flight software.

The flight software team was tasked with generating an R13 cost estimate for the project. In January 2017, they provided cost estimates for two options, A and B. Option A entailed minimal changes to R12, designed to only incorporate existing patches and improve the usability of future patches. Option B included the Option-A scope plus a feature-rich representative set of additional high-value changes, noting that the project could be extremely selective which high-value changes to pursue. Option A included seven change requests and had a total cost of $462K. Option B included 18 change requests and had a total cost of $659K. The cost per change request was lower for Option B because multiple changes were in the same software module and the cost of regression-testing a module was the same, regardless of the number of changes to that module.

The project decided to pursue Option B and kicked off an R13 effort in early 2017, more than two years since the last MSL flight software update. Over the subsequent six years, the scope of R13 developed to include many desired capabilities and bug fixes since the last upgrade – some of which were proposed even earlier than 2015 but were unable to be flown in R12. The project issued a call for ideas, and flight software team members worked with system and subsystem engineers to finesse the proposals and bring them to the MSL Change Control Board (CCB) for consideration. This approach resulted in more than 113 proposals brought forward for inclusion in R13 – an overwhelming number.

To address the high number of proposals and prioritize their approval and development, the flight software team triaged them and assigned each to one of four categories, aligned with project objectives, as follows:

(A) Does it enable streamlined patching in the future?
(B) Is it an existing or necessary patch to address a mission or vehicle vulnerability?
(C) Is it something that we believe will be very low effort and small delta risk?
(D) Is there demonstrably high return on investment that would justify the added effort or risk?

Proposals were categorized to speed up approval and development. The highest-priority items were related to health and safety of the rover, and vehicle operability during the greatly extended mission phases. After discussing the repair of known vulnerabilities, the analysis turned to operability. Project leadership believed R13 would be the last full flight software release to fly on MSL; thus, patching would be critical to maintaining operations going forward. Moreover, project personnel were believed to be moving off-project; thus, patching needed to be made easier for less-experienced engineers to accomplish safely.

Next, if a software fix was low-risk and easy – such as a one-line change in a module already affected by another issue – it could be conditionally included, provided the module was already being worked on for R13. Because the module would already be going through the development processes of unit testing and code review, and V&V and regression testing, sliding in an easy change presented little risk to the mission.

Finally, the project established a category for large items – new features, for example – that would significantly improve operations in the extended mission. Large items were hard to implement and incurred significant risk for the mission because of their scope and the amount of resources required to complete them. However, they promised a high return on investment. These proposals would allow the mission to operate more efficiently by addressing limitations and adding features which operations had been requesting for years.

Figure 2 shows the approximate relative distribution of the proposals. Almost half of all proposals were large, high-effort, but high-return changes. About a third of proposals aimed to make future patching easier and safer by rolling in existing patches and introducing infrastructure to reduce operator error, and address known vehicle vulnerabilities. And about one quarter were easy one-line fixes to existing software modules.

The A/B/C/D categorization was effective in explaining to the CCB approvers how big or important any one item might be. It allowed the team to request and allocate resources. The team hoped the categories would be effective at triage, where some proposals would be rejected by the board, but ultimately all proposals except one were approved, and development work began. At this point, the team recognized the categories were ineffective at communicating the specific changes one might expect to see in R13, and introduced topic areas.

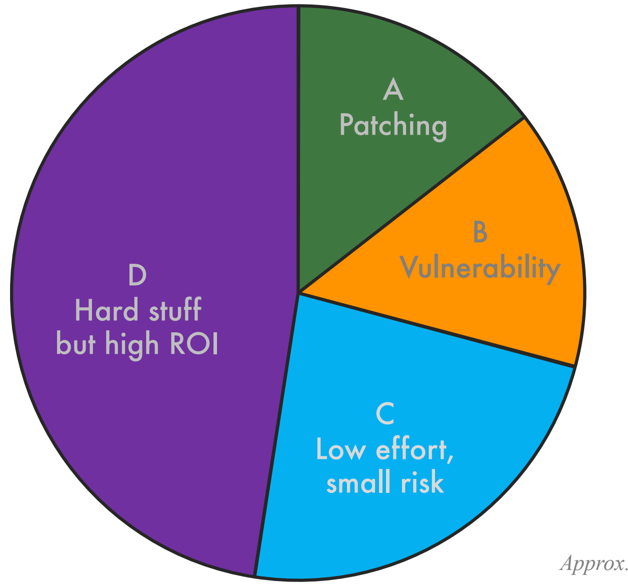

Figure 2. Patching and vehicle vulnerability fixes together were about a third of all proposals. One-quarter were one-line changes. Most proposals were "hard stuff", i.e. changes to existing modules that would require substantial resources to test and validate, but would yield high impact results for future mission operations.

4. Topic Areas

Development began following CCB approval for work for each proposal. Software development eventually involved 27 developers, mostly borrowed from other projects: with MSL in extended mission, MSL's original set had moved on to other projects. As new issues surfaced and new capabilities were brought forward, the number of distinct approved changes approached 70, even after the triage process.

The flight software team corralled the changes into nine subject matter packages, such as changes related to the way the arm operates, changes related to files and the file system, improvements to the automated on-board targeting tool AEGIS, and the new visual odometry thinking-while-driving (VTWD) capability.

Dividing the changes among nine packages facilitated communication at higher levels of abstraction between teams about the changes going into R13, and allowed for easier bookkeeping of the development of each change and its testing status. The package divisions also enabled the team to identify an individual point of contact (subject matter expert), and oversee the progress of the package through development, V&V testing, and subsequent data review after being approved for inclusion by the CCB.

VO Thinking While Driving

Integration of the new visual odometry thinking-while-driving feature (VTWD) [3], which was previously being developed as a hot-patch, was expected to allow the vehicle to drive farther while saving power. Without VTWD, the vehicle must stop moving, take images, and remain stopped while processing images to determine whether it is safe to continue the drive; with VTWD, the rover starts moving for its next step *before* visual processing begins, and performs

VO processing in parallel with the actual driving. The result is expected to increase science return due to about 50% faster drive steps and more efficient use of power.

AEGIS improvements and integration

Autonomous Exploration for Gathering Increased Science (AEGIS) is a capability which allows for opportunistic, autonomous science targeting without requiring a ground-in-the-loop cycle. The AEGIS component was integrated into the monolithic build for R13, thereby making the feature available even when all other components are not (such as in the case of a bootup which does not automatically load components, e.g., safe mode operations). Moreover, AEGIS capabilities were expanded and improved, making the science targeting easier to use.

Arm and sample processing operation improvements

The MSL sample processing subsystem is one of the most complicated pieces of machinery on Mars – and one whose operation changes frequently due to time and wear. The team wrote several forward-looking features and bug fixes to address some of the concerns about aging mechanisms. An arm inverse-kinematics fix previously implemented as a hot patch was integrated into the R13 monolith [4]. These improvements resulted in the opportunity to remove some operational restrictions (known as Flight Rules) and workarounds for behaviors that were known to be broken or fragile.

Instrument operation

Improvements to instrument handling will allow for more efficient instrument operations through exposed interfaces, and the removal of restrictions on the instruments' use. Two hot-patches were integrated into the flight software monolith, removing the need to install them on each boot and possible conflicts between these patches and any future ones.

Data management, sequencing, and patching improvements

The R13 release introduces new patching infrastructure intended to ease the burden of hot-patching with less chance of user error [1]. Hot-patching requires multiple steps, such as lifting and resetting write protection, checking memory addresses to ensure byte-alignment, and validating memory contents before and after write. With the new infrastructure, the patch author will have that procedure followed for them by onboard FSW. Additional changes and bug fixes in this package result in reduced accumulation of un-needed data and increased efficiency in ground operations.

Mobility improvements

Wheel wear has been an ongoing challenge for MSL rover drivers [13]. The terrain-adaptive speed control software that was implemented as a hot patch [2] has been integrated into the monolithic build for R13. In addition, new commands were created to address the issue with wheel wear, using arc motion to reach waypoint goals while reducing steering [14] [15]. As a result, we expect a future decrease in the rate of wheel wear due to decreased steering and odometry resulting from commanding fewer turns in place.

Power and thermal improvements

A change to the thermal subsystem improved tolerance to preheat faults. Heating which comes close to a temperature target (within a parameter), but fails to reach it, will be seen as a successful preheat. Additionally, changes to how engineering power data are generated will result in more efficient data product creation and reduced downlink size.

Fault protection improvements

In order to achieve greater resiliency in fault scenarios where current parameter values fail to be restored from nonvolatile memory, the team audited all spacecraft parameters and updated those which had dangerous default values.

Rover planner improvements

Improvements specifically targeting rover planner operations and operability include more arm target storage and expanded arm placement capability. These improvements are intended to result in simpler drill campaign target management, and less time needed reviewing complicated plans.

5. DEVELOPMENT

In effect, development began the day R12 was delivered and installed on the vehicle in January 2015. The first R13 check-in directly followed R12 installation, integrating a hot-patch into the monolith. R13 development began in earnest after its approval in 2017 with 27 on- and off-project developers contributing source code, unit tests, and documentation to the R13 package.

The team followed a looser-than-usual software development process to accommodate the distributed nature of the work.

Off-project developers' time was limited, and several of the R13 developers were opportunistically recruited from the MSL engineering operations team in order to meet development deadlines. Junior developers were paired with off-project experts who provided consultation and thorough peer review, thus ensuring adherence to strict MSL, JPL, and embedded coding standards. Development across the R13 release was tracked with Jira software. Effective unit tests were required for all changes and any new unit test development was peer reviewed along with unit test output. The topic area point-of-contact was notified once all development, unit testing, and peer review was completed for their topic area. This was a good indication that all changes for a module in the topic area package had been integrated and the module could be closed out and its changes V&V tested.

In July 2020, the first candidate release of R13 was generated and V&V testing commenced. Despite several Vehicle Systems Testbed (VSTB) hardware challenges, the COVID-19 pandemic, and the need to have the VSTB relocated to an indoor test facility during a portion of the test campaign, the test team made steady progress and completed V&V testing in May 2022. Next, a regression test campaign was executed to verify existing system capabilities were not inadvertently impacted by R13 changes, and it completed in September 2022. Finally, an R13 Software Review and Certification Record (SRCR) review was held on September 29, 2022, and R13 was approved for use in flight onboard Curiosity. Section 6 describes the iterative V&V and subsequent regression testing performed on R13 candidate builds.

Figure 3 summarizes the process flow for the R13 flight software, starting with triage (1) and ending with approval for use in flight (6). The nine subject matter packages are listed under V&V (4) and the twelve flight software modules modified by R13 are listed under the regression campaign (5). The following sections discuss why R13 became so big, what new features and bug fixes it contained, and how the team organized and kept track of all of the pieces throughout planning, development, testing, and regression.

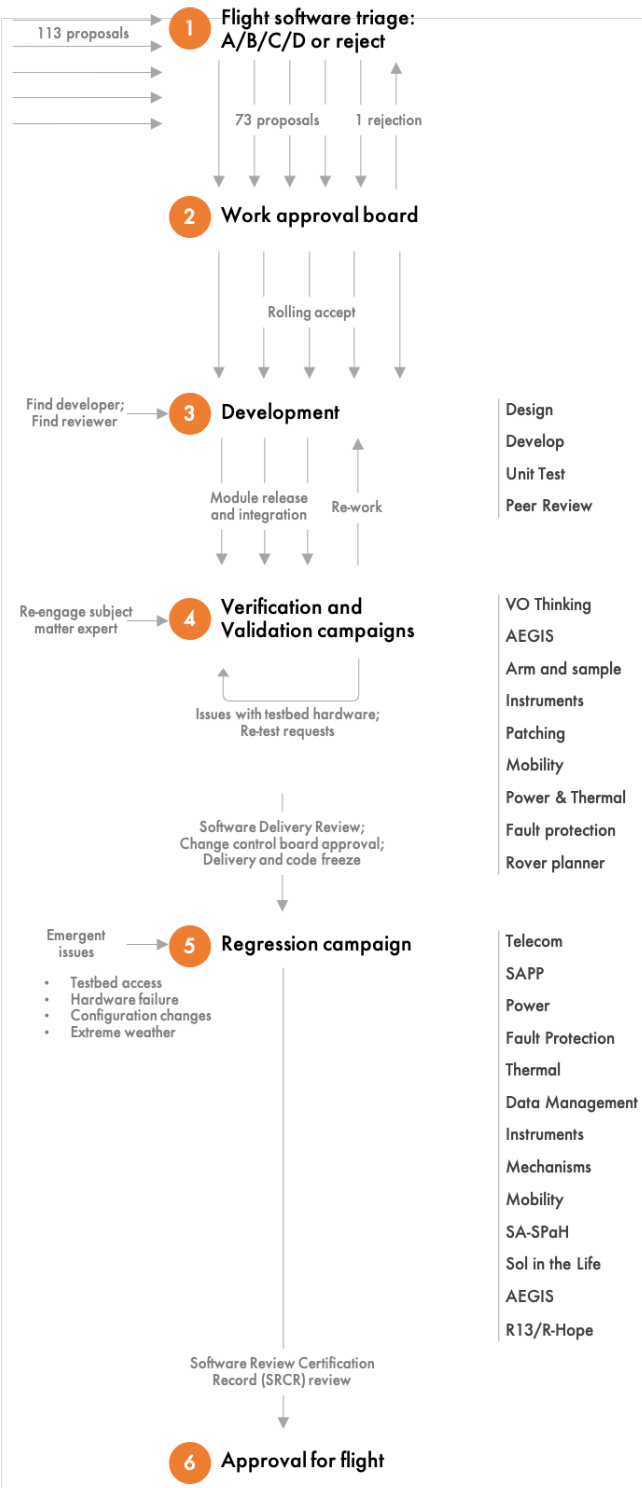

Figure 3. The R13 flight software path from triage to approval for flight

6. TESTING

Verification and Validation

After several years focused solely on development, the first candidate release of R13 was generated in July 2020. With only minor known commits still outstanding, the team began focusing on the V&V Testing phase. During this phase, detailed test procedures were designed and executed that were crafted to prove each of the R13 changes worked as expected. Hence, the V&V phase was also known as "ticket testing" in that each of the implemented Jira development tickets was tested. Since many of the testers and reviewers that would take part in these activities were not FSW team members, a V&V Lead was appointed to coordinate testing and serve as a resource for testers to get assistance with test configuration, procedure design, and review preparation.

As described in the development section, each R13 change was organized into one of nine topic areas. This structure was also used to divide responsibilities during V&V. Each topic area had a main point of contact identified who would be responsible for the development of testing procedures, test execution, and review preparation. To ensure that the test procedures were adequately designed and the data captured during test shifts met expectations, each topic area had their procedure and test data formally reviewed. Approvers for these reviews included the MSL Engineering Operations Team Chief, the FSW Team Lead, the V&V Lead, the Team Lead from the most relevant subsystem, a representative from the Systems Team, a representative from the Testbed Systems Engineering Team, and a Software Quality Assurance representative. In some cases, off-project subject matter experts were invited to participate in reviews to help ensure more complicated or higher-risk changes received appropriate scrutiny.

An important part of testing for each team member was selecting the appropriate test venue. Typically, the options available to MSL are the VSTB, an engineering model of the full Curiosity rover with flight avionics and many flight-like instruments (Figure 4); the Mission System Testbed (MSTB), a benchtop venue which has flight avionics but limited motors and instrument capabilities; or a Workstation Test Set (WSTS) running VxSim FSW emulation with simulated motors and instruments on Linux workstations. Those same options were available to us for R13 testing, but several unique complicating factors needed to be considered for this test campaign. One venue that continued to be available to us without exception was WSTS. This software simulator is capable of running on a variety of Linux machines; but while it is simple to set up and can be used in parallel by multiple testers, it is the least flight-like of all the test venues.

The MSTB consists of all major avionic components, however, actuator and instrument capabilities are simulated. Furthermore, the rover avionics have a dual string architecture and the MSTB accurately represents the redundant hardware in the loop. Importantly, the MSTB is the only testbed with actual Telecom hardware in the loop as well as the necessary support equipment to close the Radio Frequency (RF) links. This testbed is primarily used for avionics-heavy testing and is especially useful for aspects related to Earth-Mars Cruise and Entry, Descent, and Landing (EDL) phases of the mission. While the MSTB can be a great venue for

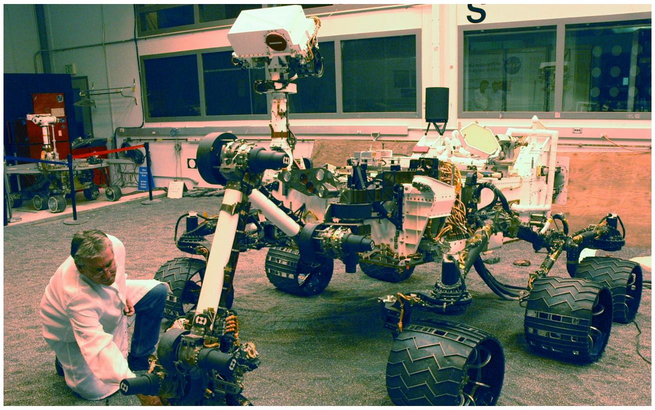

Figure 4. An operator kneels beside Curiosity's twin, the Vehicle Systems Testbed, in the In-Situ Instruments Laboratory at JPL [16].

Figure 5. The Vehicle Systems Testbed performing testing at sloped attitudes inside the constraints of the In-Situ Instruments Laboratory at JPL

FSW V&V testing, it had been re-configured to serve as an MSTB for both the MSL and M2020 projects. Additionally, the M2020 project was the primary owner of the testbed with an agreement in place to provide the MSL configuration only upon negotiated request. As much of this R13 testing occurred during the Cruise, EDL, and early Surface missions for M2020, our team had to be strategic in requesting use of the MSTB to minimize potential off-project impacts. This meant that the VSTB would be the ideal testbed for much of our campaign.

Nominally, the VSTB contains almost all major avionics, like the MSTB, with the added benefit of functioning instrument models and the actuated systems of the rover. It can be thought of as a *near*-one-to-one model of the Curiosity Rover. This makes the VSTB the ideal testbed for the Surface phase of the mission and the only venue available to test Mobility and Sampling related functions. Even though the VSTB was ideal and reasonably flight-like, several challenges were present throughout testing that made it difficult to make progress. Testing began during the early days of the COVID-19 Pandemic, so limitations on personnel access to the laboratory and other concerns impacted our testing [3]. Also, the VSTB only had one of its two RAD7500 processor-supported Rover Compute Element (RCE) for much of the testing campaign. V&V testing started in July 2020, but it was not until September of 2021 that we re-installed the second RCE into our VSTB. This other RCE had been on loan to the M2020 project during their critical pre-launch developments. Additionally, the VSTB only had one of its Rover Power Analogue Modules (RPAM) remaining as the redundant unit was permanently transferred to the M2020 project for their own VSTB. The lack of a fully redundant set of these critical avionics meant that certain types of testing - like system fault protection - had to be performed on the MSTB venue.

Other challenges included failing hardware on the testbed. While testing in April 2021, the Left NavCam became unresponsive. This component is an important part of testing mobility with stereo imaging. While we do have a second Left NavCam on the VSTB and on the flight vehicle, it is only connected to the backup RCE avionics. This meant we had no stereo imaging capability from April 2021 until September 2021 when we gained the use of a second RCE and its set of cameras. One final caveat of the VSTB venue was the physical location of the rover. While the VSTB is typically free to roam the Mars Yard at JPL (pictured in Figure 6), from August 2019 through November 2021 it was located indoors in the In-Situ Instruments Laboratory (ISIL) in JPL's building 317 (pictured in Figure 5). The significant space discrepancies between the two locations had especially challenging impacts to our Mobility related testing [3]. We used building 317 as a VSTB testing location to accommodate both the anticipated re-installation of our missing RCE and the renovations that were occurring in the Mars Yard to make room for an M2020 VSTB.

Despite all of these challenges with venue selection, location, capability, and availability, our team was able to make steady progress on testing. Our team accounted for these complications in approving each test procedure for a particular test venue and ensured that all data was valid given the testbed's condition at the post-test data review. The project concurrently planned testbed related developments such as the return of our VSTB's spare RCE and return to the Mars Yard test environment in a way that our R13 testing would be least impacted in terms of lost shift time, and most benefited: by returning key capabilities to testers just as they were necessary to make progress. In May of 2022, the last V&V Testing Data Review was completed and the project convened a Software Delivery Review (SDR) that ultimately decided the software was complete, the changes had been tested, and the release was ready for Regression Testing.

Regression

The Regression Testing Campaign would focus on ensuring existing system capabilities had not been inadvertently impacted by the changes made with R13. As with V&V Testing, a Regression Testing Lead was identified to coordinate testing, reviews, and documentation. The approvers involved with procedure and data reviews were also similar to the V&V campaign. While V&V testing was structured around each topic area and the changes made to certain capabilities, the Regression Testing would be organized around subsystem teams or domain areas much like how we organize our EO Downlink Operations Team. Many subsystem teams already had previously reviewed and approved regression procedures from previous FSW releases that could be used as a starting point or in some cases used unaltered as the R13 procedure. One special aspect to note is that the last time a system regression campaign was performed for MSL it was for the R-Hope FSW specially designed for the backup computer. As such, the team added a new regression test category called R13/R-Hope Interaction since R-Hope fundamentally changed the way the backup string behaves and it was considered critical that we check for any signs of incompatibility with the prime FSW. As we wrapped up the V&V campaign we began to update, review, and approve these regression procedures such that we could begin regression testing as soon as the SDR was passed. By the May 2022 SDR many subsystems were ready and waiting for their regression campaign while some of the more complicated subsystems such as Sample Acquisition, Sample Processing, and Handling (SA-SPaH) would follow soon after. All of the Regression subsystems as well as some information about the respective campaigns is provided in Table 2.

Heading into the regression campaign our testbed status was much improved. While the MSTB was still a shared resource with M2020, the most relevant phases of the mission for their use of this testbed had long passed with a successful Mars landing in 2021. Additionally, our VSTB was back in the Mars Yard and had a set of dual RCEs which was critical to mobility for driving space considerations and the use of stereo camera capabilities after the failure of one of our NavCams in early 2021. Unfortunately, experiencing testbed venue challenges was not unique to the V&V campaign. Since multiple subsystems required functionality only available in the MSTB (e.g., Telecom) for regression testing, the Regression Lead coordinated with M2020 Project Personnel to convert the MSTB to the MSL configuration for a short amount of time. The first time such a conversion was attempted several configuration issues were discovered along the way that delayed testing. As MSL is multiple extensions into the mission, many past subject matter experts are no longer available and various institutional infrastructure changes (e.g., cybersecurity requirements) have changed in the long periods in between MSL uses of the MSTB. These factors combine to make the already complicated conversion and troubleshooting process much more difficult than in the past. Ultimately, our team was able to get around these issues with MSTB configuration and enable our regression campaign on this venue.

In the VSTB, multiple new hardware failures froze progress on testing and resulted in unplanned VSTB downtime while our Testbed Teams diagnosed the issues. In an almost cruel occurrence, mere months after the A-side RCE was returned to the VSTB and within days of the R13 SDR, the B-side RCE suffered a failure that prevented loading FSW in the typical manner and rendered the computer useless for regression. Luckily, the aforementioned failed NavCam was also tied to this B-side RCE. This meant that our regression campaign on VSTB could continue unhindered on the A-side RCE. Had the failed components been of some other pairing the team may have still been in the midst of working the problem at the time of this writing. Next, during the first attempt of our Sol-In-the-Life Test (SITL), which serves as a regression for the system at large and covers a typical Sol's worth of sequenced cross-system functional activity, the VSTB experienced an apparent issue with the Left Mastcam. Upon further investigation it was discovered that the issue was in fact somewhere in the following set of hardware components: the MARDI, MAHLI, Mastcam (MMM) Digital Electronics Assembly (DEA), cabling, or the RCE card that interfaces with the MMM instruments. Luckily, the relevant sections of MMM regression within the Instruments topic area had already been completed in a valid regression run and the team determined the MMM capabilities were not required for the SITL test to be considered a success.

In addition to all of the hardware related issues, the team had to deal with the Southern California summer heat when considering the VSTB's excursions into the Mars Yard. The testbed has a gaseous nitrogen-based cooling system (unique to the engineering model), but it is not always sufficient to keep the VSTB below safe temperatures under certain circumstances. Despite these challenges, the team completed the campaign in September 2022 after 4 months of testbed activity and data reviews. With all 12 regression tests complete (see Table 2), the project convened a Software Review Certification Record (SRCR) review resulting in approval to use R13 in flight.

7. Installation and Usage In Flight

Since the software release has finished testing and review as of this writing (October 2022), the team will embark on a months-long campaign to prepare for its installation and use in flight. This effort will likely complete in 2023 and will require not only the uplink and installation of the new FSW image, but sweeping changes to parameters and sequences on the vehicle and software tools on the ground.

For the rover, one impact that will need to be considered before R13 is used in flight are updates to FSW parameters. Using R13 the first time in flight will cause modules that have parameter schema changes in R13 to reset those parameter values to their original default values. While the FSW default parameter values are safe, many will need to be changed in order for us to maintain all of our previous capabilities and enable the new features promised as part of R13. In addition to planning for parameter changes, we will also need to review reusable sequences built for the R12 FSW that we repeatedly invoke in daily operations without re-uplinking. When activities are first planned on Earth, they are carefully designed to make use of reusable sequences that can be left onboard the vehicle indefinitely. This vastly simplifies planning and reduces risk by eliminating the need to continually re-transmit the same commands. With R13 representing such extensive changes to FSW, we will need to update, test, and uplink new versions of some reusable sequences before we can take advantage of R13's new capabilities.

On the ground, we will need to make updates to Flight Rules, SEQGEN, RP-check [17], and various other scripts that help us automate daily operations. The project maintains a database of Flight Rules that restrict commands or activities that could be detrimental to the vehicle or operations. While rules can be waived with appropriate review and approval, the team relies on careful definition and enforcement of these rules to ensure successful commanding. R13 will significantly impact our project's set of Flight Rules. Some R13 changes that simplify commanding or remove vulnerabilities will result in the removal of Flight Rules. Other rules will need to be updated with new information based on changes to R13. We will also need to author new rules in some cases based on the intended usage of the added capabilities within R13. Flight rules vary significantly in nature. Some rules are checked for compliance manually before each uplink, while others are checked by a JPL multi-mission tool called SEQGEN that each mission adapts to their own operations.

Table 2. Cumulative System Regression Summary. MSTB is the Mission System Testbed, which includes flight-like avionics but not actuators. VSTB is the Vehicle System Testbed, including flight avionics, actuators and many instruments. WSTS is the Work Station Test Set, a pure simulation running VxSim and motor and instrument simulators on Linux workstations.

Regression Test Subsystem	**Venue (Shifts) MSTB**	**VSTB**	**WSTS**	**Certified for R13**
Telecom	7			Yes
SAPP (Attitude)		2		Yes
Power		1	2	Yes
System Fault Protection	4			Yes
Thermal	1		4	Yes
Data Management		3		Yes
Instruments & AEGIS		10		Yes
Mechanisms		6		Yes
Mobility		11		Yes
SA-SPaH		6	2	Yes
Sol In the Life (SITL)		2		Yes
R13/R-Hope Interaction	1			Yes
Total shifts	13	41	8	—

Figure 6. The M2020 VSTB pictured in the Mars Yard at JPL as seen from the JPL Virtual Tour.

SEQGEN is designed to model the state of the ground system and spacecraft throughout a specific period of time. The input required by SEQGEN to accurately model events includes Flight Rules and a Spacecraft Model. Changes made by R13 to these areas will require accommodations within SEQGEN. Finally, the team uses a vast expanse of scripts and tools that have been built by team members as we continue to add automation into our operations. New R13 capabilities and extensive changes to command and telemetry dictionaries necessitate a review and update to repositories in order to ensure the transition to R13 doesn't lead to unexpected tool issues.

Even with all of these items addressed, physically installing software requires careful coordination. First, the FSW image must be prepared for uplink. The image itself is almost 22 MB and cannot be sent all at once. The FSW image is compressed and split into roughly 50 smaller files in preparation for uplink. These files can be sent either directly from the Deep Space Network to the rover's High-Gain Antenna or by requesting a forward-link from other Martian orbiter teams that can have files sent during one of their spacecraft's regularly scheduled ultra-high frequency communication sessions. When all of the files are onboard and the team is ready to proceed, the rover payload will be powered off for the four main FSW transition plans.

The rover avionics have several areas of memory available for use. The ones designed to hold the FSW images that are loaded on boot are regions of NOR-style memory. There are four NOR zones, each large enough to hold one FSW image. At any given time, a group of two NOR zones are selected via ground command for use while the other two are not. A group of NOR zones being selected means that the vehicle can attempt to boot from an image in those two zones. There is no way for the rover to autonomously attempt to boot from a zone in an unselected group, even in a fault scenario. Typically, each of the two selected zones contains identical copies of the FSW version currently in use. If a boot is unsuccessful from the first zone in the selected group, a boot will be attempted with the second zone. This design helps reduce risk during FSW transitions.

During the first of the four main transition plans, the new FSW image will be saved into the first NOR zone of the unselected group while the current version of FSW is saved into the second zone of the unselected group. This will later allow us to change the selected group and first try several boots with R13 and still have the capability for the rover to boot with R12 if something goes wrong. The second plan saves the image in a specific area of RAM that is checked on startup for a valid FSW image before the NOR zones are checked. This allows us to next command a reset and boot into R13 using the imaged saved in RAM without really committing to that FSW version by switching the selected group. The third plan involves changing the selected group so the vehicle will now try booting from R13 first. As mentioned before, if something goes wrong during this plan the second group has R12 which would allow the vehicle to still use the older version of FSW even if a serious issue was experienced. Finally, the fourth plan would write R13 into the second NOR zone in the newly selected group, fully committing us to the use of R13.

The installation of R13 is expected to occur in early 2023.

8. Discussion

The scope of R13, MSL's ultimate flight software release, was immense, with 184 proposed individual changes to 55 different flight software modules. The previous R12 release contained just 27 individual changes in all.

Of R13's 184 changes (summarized in Figure 7):

- 56 were bugs which were fixed;
- 53 were related to new features or capabilities;
- 63 were bugs which MSL decided not to fix, but to mitigate instead with other means such as process improvements and creation of new Flight Rules; and
- 12 were new features which the project decided not to pursue developing.

Thus, 109 individual changes were incorporated into the flight software and released as R13, resulting in R13 having four times as many changes as R12.

Table 3 summarizes the measurable changes to resources in the R13 release. The overall flight software image size increased by 2.9% over the previous R12 releases, totaling 21.921 MB for the compiled monolithic binary image. Usage of the RAD750 RAM increased from 29.303 MB (R12) to 30.144 MB, as calculated by telemetry channels. The processor's RAM usage on each RAD750 board is hardware-limited to 32 MB (more RAM is available on other boards). The number of non-volatile parameter memory (NPM) records increased by nine from R12 to 9,153 records, well below the 15,000-record limit for R12 and R13. One additional data product application identifier (APID) was defined. There were 42 new flight software commands, bringing the total number to 3,912; and eight new hardware commands were added in R13. An additional 200 data channels were introduced, for a total of 19,800 channels in R13. Finally, 224 new text message event reports (EVRs) were added, and 11 were removed.

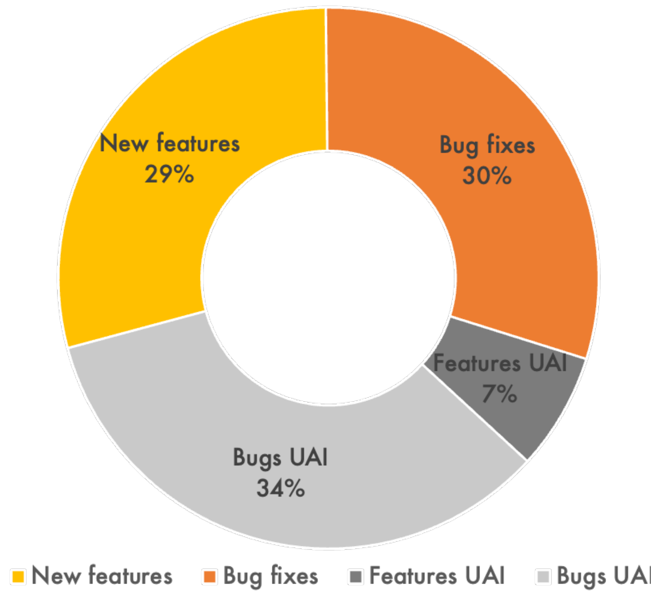

Figure 7. In all, 184 changes were proposed for R13. The final version includes 53 (34%) new features and 56 (29%) bug fixes. However 63 (30%) bug fixes and 12 (7%) features were proposed but ultimately rejected by the change board with the disposition "use-as-is" (UAI).

Table 3. Comparison of key resources in R12 and R13

Resource	R13	R12	Delta
Flight image size (MB)	21.921	21.304	+0.617
RAM usage (MB)	30.144	29.303	+0.841
NPM records	9,153	9,144	+9
APIDs	606	605	+1
FSW commands	3,912	3,870	+42
Hardware commands	135	127	+8
Data channels	19,800	19,600	+200
EVRs	26,642	26,429	+224, –11

9. Conclusion

With the realization that Curiosity had the potential for years of future operations and that there will always be a need for patches through Curiosity's lifetime, in April 2016 the MSL flight software team proposed developing an R13 full flight software release with the primary motivation of making patching flight software less onerous and safer in the future. In early 2017, the project decided to pursue a feature-rich R13 that would include changes to make future patching easier and a set of high-value and low-cost changes.

Since R13 is expected to be the last full flight software update for Curiosity, the project decided to cast a wide net and allow subsystems to propose changes. Initially, 113 changes were proposed. To screen which proposed changes to bring to CCB for approval, the flight software team binned the proposed changes into four categories related to vehicle health and safety and future operability. Of the initial proposals, 40 did not fall into any of the four categories and were rejected by the flight software team. Most of the others were taken to the CCB where only one was rejected. Such a low CCB rejection rate was a testimony to the effective job the flight software team had done in screening the proposed changes.

During the initial development phase of approximately three years, proposed changes continued to be brought forward by the subsystems as new issues surfaced. Ultimately, 109 changes were approved by the CCB. The flight software team binned the change requests into nine subject matter packages, and each package had an individual point contact to oversee its progress all the way through development, V&V testing, and data review.

R13 V&V testing and data reviews started in July 2020 and lasted nearly two years due to several challenges that were encountered, including *a*) limited in-person testbed staffing due to the lingering COVID-19 pandemic, *b*) the lack of dual-RCE capability in the VSTB for 15 months, *c*) the lack of a functioning Left NavCam for nearly 6 months in 2021, and *d*) the VSTB relocation from the Mars Yard to a small area in an indoor test facility until November 2021. R13 regression testing was then performed over the next four months with its own set of challenges that included technical difficulties in converting the MSTB from the M2020 to MSL configuration and unplanned VSTB downtime due to hardware failures. Despite these challenges, the test team made steady progress and completed the V&V testing data review in May 2022 and the regression testing data review in September 2022.

When R13 was first proposed, Curiosity had been operating on Mars for about 4.5 years and many of the original MSL flight software developers had transitioned to the M2020 project. Curiosity had completed its two-year prime mission and was in extended mission with a reduced budget. Extra funds were not allocated to MSL for the development of R13. Rather, R13 was incorporated into the existing MSL operating budget by the EO team allocating 4.5% of their staffing budget to additional flight software personnel for support of R13 development. That amounted to the addition of 1.5 full-time employee equivalents (FTE) spread out over 27 developers, mostly from outside of MSL working part-time on R13.

Development and testing of R13 came to a successful close in September 2022 when R13 was approved for use in flight at its SRCR. Of all the full release upgrades approved for use in flight onboard a NASA Mars rover, R13 has the largest number of change requests and the longest combined development and test period. It is remarkable that a project on extended mission could achieve a flight software update within its normal operating budget with so many bug fixes and new features.

The development and testing of R13 spanned three MSL project managers (plus one interim), four EO team chiefs (plus one acting), and three Flight Software Leads. The smooth transfer of leadership at all three levels has enabled R13 development and testing to continue to make progress and overcome the challenges that were encountered. Upload and installation of R13 onboard Curiosity is expected to occur in early 2023. The MSL operations team eagerly awaits the elimination of numerous nuisance bugs, the improvements that will make patching flight software easier, and the opportunity to use new features which will all come in R13.

ACKNOWLEDGMENTS

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration (80NM0018D0004). The authors would like to thank the Mars Science Laboratory Program for supporting this research.

We acknowledge the work of Mars Science Laboratory developers: Neil Abcouwer, Ron Baalke, Eddie Benowitz, Jeff Biesiadecki, Anna Boettcher, DJ Byrne, Joseph Carsten, Kevin Davis, Jonathan Denison, Gary Doran, Dan Gaines, Alexandra Holloway, Dan Levine, Todd Litwin, Mark Maimone, Deep Mukherji, Steven Myint, Marc Pack, Nick Peper, Gregg Rabideau, Arturo Rankin, Mark Reid, Anna Sabel, Robert Steele, Sloan Swieso, Igor Uchenik, and Clayton Williams.

We would also like to thank to all Mars Science Laboratory staff who directly or indirectly supported the development, testing, and documentation of R13: Yanhua Anderson, Keri Bean, Jeff Biesiadecki, Ron Baalke, Charlie Bell, Anna Boettcher, Lea Chandler, Matthew Dailis, Jonathan Denison, Harel Dor, Deirdra Fey, Brian Franz, Dan Gaines, Rishab Gangopadhyay, Matthew Gildner, Evan Graser, Amy Hale, Alfonso Herrera, Evan Hilgemann, Raymond Ho, Alexandra Holloway, Lauren Kafadarian, Brian Kahovec, Matthew Keuneke, Reidar Larsen, Mark Maimone, Matthew Menten, Camden Miller, Sophia Mitchell, Ryan Mukai, Emily Newman, Neel Patel, Nikunj Patel, Betina Pavri, Nicholas Peper, Flynn Platt, Kimberly Rink, Peter (PJ) Rollins, Anna Sabel, Dane Schoelen, Ashley Stroupe, Sloan Swieso, Matthew Van Kirk, Freddy Wang, James Wang, Monica Wang, and Emme Wiederhold.

REFERENCES

[1] M. Maimone, A. Holloway, "Load and patch: Improving hot patch capabilities in Curiosity's flight software," Flight Software Workshop, February 2022

[2] O. Toupet, J. Biesiadecki, A. Rankin, A. Steffy, G. Meirion-Griffith, D. Levine, M. Schadegg, and M. Maimone, "Terrain-adaptive wheel speed control on the Curiosity Mars rover: Algorithm and flight results," *Journal of Field Robotics*, Aug. 2019.

[3] M. Maimone, N. Patel, A. Sabel, A. Holloway, A. Rankin, "Visual Odometry Thinking While Driving for the Curiosity Mars Rover's Three-Year Test Campaign: Impact of Evolving Constraints on Verification and Validation", IEEE Aerospace Conference, Big Sky, Montana, March 2022.

[4] A. Rankin, A. Holloway, J. Carsten, M. Maimone, "Integration of an Arm Kinematics Hot Patch Onboard the Curiosity Rover," IEEE Aerospace Conference, Big Sky, Montana, March 2021.

[5] E. Benowitz & M. Maimone, "Patching Flight Software

on Mars," Workshop on Spacecraft Flight Software, Laurel, MD, USA, 27 October 2015

[6] D. J. Mudgway, "Telecommunications and Data Acquisition Systems Support for the Viking 1975 Mission to Mars," JPL Publication 82-107, May 15, 1983.

[7] A. Holloway, DJ Byrne, N. Peper. "R-Hope: Development approach to extreme non-volatile memory reuse onboard the Curiosity Rover". *8th IEEE International Conference on Space Mission Challenges for Information Technology* (SMC-IT 2021), July 2021.

[8] A. Holloway, N. Peper, A. Anabtawi J. Quade, DJ Byrne. "Building a Lifeboat: MSL's Uplink and Installation Campaign to Restore a Failing Backup Computer". *IEEE Aerospace Conference*, Big Sky, Montana, March 2022.

[9] Where is the rover? https://mars.nasa.gov/msl/mission/where-is-the-rover/

[10] MSL Sol-200 Anomaly. *JPL Lessons Learned.* 2014. https://llis.nasa.gov/lesson/11201

[11] A. Rankin, M. Maimone, J. Biesiadecki, N. Patel, D. Levine, O. Toupet, Mars Curiosity Rover Mobility Trends During the First Seven Years, *Journal of Field Robotics*, January 2021.

[12] SMAP Team Investigating Radar Instrument Anomaly, July 9, 2015, updated August 5, 2015 https://smap.jpl.nasa.gov/news/1244/smap-team-investigating-radar-instrument-anomaly/

[13] A. Rankin, N. Patel, E. Graser, J.-K. F. Wang, and K. Rink, "Assessing Mars Curiosity rover wheel damage," in IEEE Aerospace Conference, Big Sky, Montana, USA, Mar. 2022

[14] N. Abcouwer, et al., "These Wheels Are Made For Arcing: Two New Mobility Commands to Reduce Wheel Wear," Flight Software Workshop, February 2022

[15] M. Maimone, et al., "These Wheels Are Made For Arcing: Two New Mobility Commands to Reduce Wheel Wear," IEEE Aerospace Conference, March 2023 (submitted)

[16] Testing Precision of Movement of Curiosity's Robotic Arm, February 22, 2012, https://mars.nasa.gov/resources/3812/testing-precision-of-movement-of-curiositys-robotic-arm/?site=msl

[17] M. Maimone, S. Maxwell, J. Biesiadecki, S. Algermissen, "RP-check: An Architecture for Spaceflight Command Sequence Validation," IEEE Aerospace Conference, 01 March 2018.

BIOGRAPHY

Alexandra Holloway *aims to design flight software for ground operability. She leads the flight software team for the Curiosity rover, chairs the data management subsystem in tactical operations, and serves as Assistant MSL Engineering Operations Team Chief. Prior to her work on MSL, Alexandra designed tools and processes for operators of the Deep Space Network as the system transitioned to Follow-the-Sun operations. Alexandra received her Ph.D. in Computer Science at the University of California, Santa Cruz, specializing in human-centered design and storage systems.*

Jonathan Denison *is the MSL Engineering Operations Team Chief and staffs tactical roles for the Systems and Data Management teams. Jonathan also served as R13 V&V Lead for the last 6 months of the testing campaign and was the Activity Lead for the R12.0.4 FSW Transition on Sol 2808. Jonathan earned his B.S. in Aeronautical and Astronautical Engineering and M.B.A. from The Ohio State University as well as a M.S. in Computer Science from the University of Southern California.*

Neel Patel *is Deputy MSL Engineering Operations Teach Chief and the MSL Testbed Systems Engineering Team Lead; he has staffed tactical operations roles for MSL Systems & Sampling, and served as the Regression lead for R13. He also supports the M2020 Project as part of the Robotic Operations team in the Sampling & Caching Role and previously supported V&V and transition to operations of the Adaptive Caching Assembly. Neel earned a B.S. in Aerospace Engineering from the University of Maryland and a M.S. in Astronautical Engineering from the University of Southern California.*

Mark Maimone *is a Robotic Systems Engineer in the Robotic Mobility group at the Jet Propulsion Laboratory. Mark designed and implemented the GESTALT self-driving surface navigation Flight Software for MER and MSL missions; during MSL operations served as Deputy Lead Rover Planner, Lead Mobility Rover Planner and Flight Software Lead; developed downlink automation tools for MER and MSL; and is now the Mars 2020 Robotic Operations Deputy Team Chief, and a member of the Rover Planner and Rover FSW development teams. He holds a Ph.D. in Computer Science from Carnegie Mellon University.*

Arturo Rankin *received his Ph.D. in Mechanical Engineering at the University of Florida in 1997 and has worked at the Jet Propulsion Laboratory since then. He is currently a Robotic Systems Engineer in the Robotic Systems Staff group and the Mars 2020 Robotic Operations Team Chief. Prior to becoming the Mars 2020 Robotic Operations Team Chief, he was the Deputy Robotic Operations Team Chief from 2019 to September 2022. Mars 2020 is the third Mars rover project he has work on. Prior to Mars 2020, he was a Mobility/Robotic arm downlink analyst on the Mars Exploration Rover project and the Mobility/Mechanisms Team Lead and Flight Software Team Lead on the Mars Science Laboratory project.*