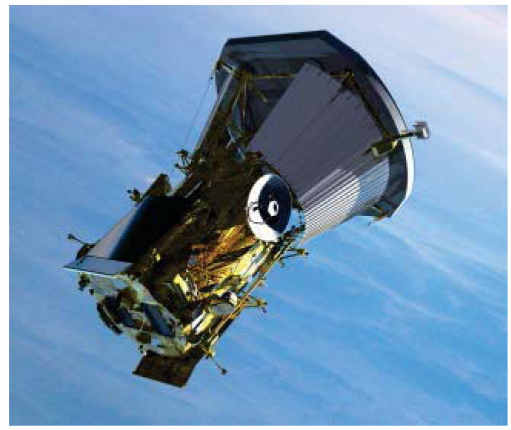
Confidence in Spacecraft Software: Continuous Process Improvement in Requirements Verification

M. Annette Mirantes
Johns Hopkins Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723
443-778-8215
Annette.Mirantes@jhuapl.edu

Maria Spezio
Johns Hopkins Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723
443-778-4410
Maria.Spezio@jhuapl.edu

Kristin A. Wortman
Johns Hopkins Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723
443-778-9634
Kristin.Wortman@jhuapl.edu

Van Allen Probes



Solar Probe Plus

Abstract—**The Space Department's (SD) Embedded Applications Group at the Johns Hopkins University Applied Physics Laboratory (JHU/APL) continues to apply its experience with independent requirements verification to achieve high confidence levels in the quality of Spacecraft Software. NASA's Van Allen Probes observatory, which was launched in August 2012, provides another stepping stone for continuous improvement of our independent software requirements verification process to support future spacecraft missions.**

In 2005, a study was conducted within JHU/APL's Embedded Applications Group that summarized lessons learned, cost efficiency and effectiveness of independent Spacecraft Software requirements verification across four supported NASA spacecraft missions [1]. The outcome of the 2005 study provided recommendations to improve how we conduct requirements verification on Spacecraft Software for JHU/APL SD supported missions. Many of the 2005 study's recommendations were directly applied to the Van Allen Probes mission and this recent experience provides another opportunity to analyze the cost efficiency and effectiveness of our independent Spacecraft Software requirements verification process.

The follow-on paper will highlight the 2005 study's recommendations and elaborate on the Embedded Applications Group's implementation of these recommendations during the requirements verification program for the Van Allen Probes Spacecraft Software.

A variety of metrics can be used during development to monitor Spacecraft Software maturity patterns, to assess confidence level in Spacecraft Software prior to launch and to guide the focus of the Spacecraft Software's requirements verification program. Elaboration on these metrics will be discussed in this paper and use cases will be presented for the Van Allen Probes mission. The paper will also highlight lessons learned during the Van Allen Probes Spacecraft Software requirements verification program. The results of the analysis and the lessons learned will yield additional recommendations for continuous improvement of our independent Spacecraft Software requirements verification process to be followed on the next NASA mission, Solar Probe Plus, which has a planned 2018 launch date.

TABLE OF CONTENTS

978-1-4799-1622-1/14/$31.00 ©2014 IEEE

1. INTRODUCTION

The Embedded Applications Group in the Space Department (SD) at Johns Hopkins University Applied Physics Laboratory (JHU/APL) has strived since 2001 to continually improve our Spacecraft Software Requirements Verification (SSRV) process over the course of five NASA spacecraft missions. Increased complexity of Spacecraft Software functionality and shrinking budgets over the years have driven the need for an efficient and effective SSRV process to ensure a high level of confidence in Spacecraft Software. In 2005, a study was conducted on four NASA spacecraft missions to summarize findings on our SSRV process and make recommendations for future JHU/APL supported missions [1]. The four missions that were included in the study were COmet Nucleus TOUR (CONTOUR), MEcury Surface Space Environment, Geochemistry, and Ranging (MESSENGER), Solar-TErrestrial RElations Observatory (STEREO), and New Horizons. Descriptions of these missions can be found in [1].

Results of the 2005 SSRV Study yielded recommendations for the next mission, Van Allen Probes (previously known as Radiation Belt Storm Probes). Van Allen Probes implemented many of the 2005 SSRV Study's recommendations and provides another opportunity to summarize lessons learned and make recommendations as we move forward on the current JHU/APL supported NASA mission, Solar Probe Plus.

Mission Overview

Van Allen Probes mission is part of the NASA's Living With a Star Geospace program. The Van Allen Probes observatory consists of two spacecraft, each hosting an array of science instruments that are providing measurements needed to characterize and quantify the processes that produce relativistic ions and electrons. Van Allen Probes are helping us to understand the Sun's influence on the Earth and near-Earth space by studying the planet's radiation belts on various scales of space and time [2]. Van Allen Probes was launched in August 2012 and the SSRV program was completed in April 2012.

Solar Probe Plus, JHU/APL's next NASA Living With the Star mission, will come closer to the Sun than any spacecraft has ever flown. Solar Probe Plus will study the streams of charged particles the Sun hurls into space from an unprecedented vantage point which lies inside the Sun's corona, its outer atmosphere, where the processes that heat the corona and produce solar wind occur [3]. Solar Probe Plus has a planned launch in the summer of 2018 and the SSRV program will commence in fall of 2014.

Terminology Use

Throughout the writing of this paper several terms are used and the meanings of those terms need to be clarified for understanding of the concepts. Two terms, *SSRV process* and *SSRV program*, are frequently used in the discussions. The term *SSRV process* is used to reference the specific sequence of activities, including the use of tools, to produce the test documentation and test artifacts. The term *SSRV program* is used to reference all SSRV process activities, and the management of the SSRV process activities, that were performed cumulatively for all Spacecraft Software releases. Another term *Spacecraft Software* is used to refer to the on-board flight software that includes the source code for one or more Computer Software Configuration Items (CSCIs). A Spacecraft Software CSCI can be considered as software to perform functionality for either boot, guidance and control, or command and data handling or a combination of software (e.g., guidance and control and command and data handling). CSCIs were organized differently for most of our heritage missions. The term *Acceptance Test* is synonymous with software requirements verification. Acceptance test is considered part of JHU/APL's Spacecraft Software development process.

Spacecraft Software Overview

As with previous NASA missions, JHU/APL intends to reuse Spacecraft Software components implemented on Van Allen Probes and apply them to the Solar Probes Plus mission. The Van Allen Probes Spacecraft Software provides Command and Data Handling (C&DH) functionality and was developed by the JHU/APL's SD Embedded Applications Group. The C&DH software is designed to work with Goddard's core Flight Executive (cFE) architecture. Components of the Van Allen Probes C&DH software will be reused in the design of the Solar Probe Plus C&DH software. Unlike Van Allen Probes, Solar Probe Plus Spacecraft Software will have an additional component to perform Guidance and Control (G&C) of the spacecraft. The G&C component will add a layer of complexity to the Solar Probe Plus Spacecraft Software that will increase the scope of the SSRV program as compared to Van Allen Probes.

Background Information

The 2005 SSRV Study [1] noted varying differences and complexities that affect the number of Spacecraft Software requirements, the size of the Spacecraft Software and the percentage of code reuse for four previous JHU/APL supported missions.

Table 1 lists the spacecraft system attributes, the number of external interfaces and the number of science instruments, for the four missions studied in 2005 with the addition of Van Allen Probes and Solar Probe Plus.

Table 1. Mission System Comparison

Mission	External Interfaces	Science Instruments
CONTOUR	12	4
MESSENGER	19	7
STEREO	15	4
New Horizons	12	7
Van Allen Probes	8	5
Solar Probe Plus	17	4

Implementation of the G&C system for each mission is important when evaluating the complexity of the overall spacecraft system. Table 2 provides background information for each mission and describes the mission's G&C system as either *open-loop* or *closed-loop*. An *open-loop* G&C system refers to the guidance and control operations being performed with ground system intervention. A *closed–loop* G&C system refers to the guidance and control operations being performed on-board the spacecraft without ground system intervention.

Table 2. Guidance and Control System Across Missions

Mission	Guidance and Control
CONTOUR	Closed-loop
MESSENGER	Closed-loop
STEREO	Closed-loop
New Horizons	Closed-loop
Van Allen Probes	Open-loop
Solar Probe Plus	Closed-loop

Table 3 lists several Spacecraft Software attributes for the four missions studied in 2005 [1] with the addition of Van Allen Probes and the current number of requirements and estimated percentage of code reuse for Solar Probe Plus. The actual launch year for the five heritage missions and the planned launch year for Solar Probe Plus provide a timeframe reference for each mission related SSRV program.

Table 3. Mission Spacecraft Software Comparison

Mission	Number of Requirements	% Code Reuse
CONTOUR 2002	690	30%
MESSENGER 2004	1035	30%
STEREO 2006	1422	15%
New Horizons 2006	1074	35%
Van Allen Probes 2012	462	45%
Solar Probe Plus 2018	612	65%

Similarities and differences across the six missions make it difficult to accurately compare the Spacecraft Software complexity using the attributes shown in Table 3. Several factors that affect the number of requirements attribute can be contributed to differences in software and hardware architecture, type of G&C, percentage of code reuse, number of CSCIs, procurement of vendor boot code, and organization of requirements (e.g., separation of requirements into interface control documents). Another factor that makes it difficult to compare the complexity of these missions is the evolution over time to improve our Spacecraft Software requirements process.

Independent SSRV was required to reduce risk and was used to assess confidence in the Spacecraft Software of our heritage missions. The Solar Probe Plus mission also requires independent requirements verification be performed on the Spacecraft Software. The spacecraft system and software attributes summarized in Tables 1, 2 and 3 are useful in understanding the scope and composition of each mission supported over the years by the Embedded Applications Group. The information is provided as background to support the SSRV discussions in the remaining sections of this paper.

Test Like You Fly Philosophy

JHU/APL's SSRV process follows the "Test Like You Fly" (TLYF) philosophy [4]. TLYF specifies the use of flight-like commands in a flight-like test environment to verify the Spacecraft Software's functional requirements. Verification of functional requirements must be conducted by a team of test engineers who are independent from the code development. The Van Allen Probes SSRV test designs were developed by an independent team consisting of several test engineers. Test engineers conducted the test executions using a test simulation environment with science instrument emulators and utilizing a variety of JHU/APL developed software test tools.

2. VAN ALLEN PROBES SSRV PROGRAM

The 2005 SSRV Study summarized the lessons learned from previous working groups, actual project experiences and metrics for the CONTOUR, MESSENGER, STEREO and New Horizons missions. The lessons learned and recommendations resulting from the 2005 SSRV Study are listed for easy reference below in Table 4 and Table 5.

Table 4. 2005 SSRV Study Lessons Learned [1]

Lesson Learned
Invest in testing: training, tools, communications
Increase focus on cost-effectiveness
Use metrics for Process improvement
Manage resources purposefully
Scope the Effort
Involve the development team
Leverage all test efforts
Track changes to test documentations and tools
Plan for test re-use
Plan “waiting” time

Table 5. 2005 SSRV Study Recommendations for Future Missions [1]

Future Mission Recommendation
Involve experienced testers heavily in requirements review, and make their membership on the requirements review panel mandatory.
Carefully review requirements to assure testability and to filter out extraneous design information
Perform risk analysis to determine whether functional areas will be tested via scenario testing or traditional requirements-based testing.
Build up scenario-based tests iteratively and incrementally.
Maintain all test plans in Requirements Tracking Tool, link them to requirements, and capture test methods.
Use scenarios as the basis of regression test: use automation for cases to be included in regression; and do the remaining test manually.
Build and review test cases and documentation incrementally. Track changes using a COTS version control system.
Gather and use metrics throughout the test cycles to allow dynamic process adjustments when needed.
Plan for re-use when developing test cases, and look for ways to simplify them.
Use verification matrix and test methods to identify where other test efforts can be leveraged to reduce duplication across CSCs.

The lessons learned (Table 4) and recommendations (Table 5) extracted from previous SSRV programs provide the foundation for the discussion in the following subsections on the Van Allen Probes SSRV experience.

Test Resources and Management Commitment

One valuable lesson learned from past missions was to invest more in the training of a core group of test engineers and to involve them in earlier phases of the development process (e.g., requirements). Investment in training includes the availability, knowledge and use of off-the-shelve and custom software tools to enhance the overall effectiveness and efficiency of the SSRV process.

The core team for the Van Allen Probes SSRV program consisted of four test engineers. In addition to the core team there was a technical lead and two part-time test engineers. Two of the four core test engineers worked full-time for the majority of the SSRV program. The remaining two core test engineers worked part-time (25%-50%) on the SSRV program. The two part-time test engineers were strategically used during periods of the SSRV program.

Technical background of the overall test team varied in the number of years’ experience in software development and testing. Retention of the core test engineers was achieved throughout the SSRV program. A stable team reduced the need for training time to bring new test engineers up to speed with the core test team. Commitment of resources to the SSRV program was well supported by all levels of Van Allen Probes program and line management.

Van Allen Probes SSRV program did not fully staff test engineers during the Spacecraft Software development requirements phase. Even though this is highly desirable and recommended in the 2005 SSRV Study, it was difficult from a project funding perspective. Test engineers were brought onto the program prior to Critical Design Review (CDR), but the test team was not fully staffed until after CDR. However, the technical test lead for the Van Allen Probes SSRV program was heavily involved in the requirements review phase and did serve as a peer review board member for the requirements review and CDR.

A variety of commercial off-the-shelf (COTS) tools were used throughout the Van Allen Probes SSRV program and these tools are listed in Table 6 along with their functional descriptions.

Table 6. Van Allen Probes SSRV Program Tool Set

COTS Tool	Description
IBM Rational DOORS	Test plan and test specifications with test case mappings to software requirements.
GForge Subversion	Configuration and version control of test artifacts (e.g., scripts, logs).
PassMark’s TestLog	Test execution records.

Atlassian JIRA	Software issue reporting and tracking.
Windchill Product LifeCycle Management	Test documentation configuration management.

Training was provided to the test team on the use of the COTS tool set to support SSRV activities. Additionally, customized in-house developed test tools were provided to work with the test simulations environment. These test tools provided functionality to enable control over the spacecraft command format and to simulate science instrument telemetry and were used extensively in the test designs.

Test Scope and Cost Effectiveness

JHU/APL software is categorized according to criticality, use, risk and size. Categorization of the software determines the required SSRV process activities to commensurate with the assessed risk. Spacecraft Software is considered a critical CSCI and therefore requires all requirements to be verified by an independent test team. An internal test team is used to verify the Spacecraft Software requirements for each incremental Spacecraft Software release before its delivery for use during spacecraft Integration and Test (I&T) and mission operations. The focus of the SSRV program is to implement an effective SSRV process to efficiently reduce software risk to the mission.

Van Allen Probes SSRV process was tailored to use a criticality rating to distinguish the most to least critical Spacecraft Software requirements. The criticality assigned to each Spacecraft Software requirement was based on JHU/APL's definitions that were established and used to classify software defects on heritage spacecraft missions. Table 7 contains the criteria description used to determine the criticality rating for each Van Allen Probes Spacecraft Software requirement.

Table 7. Software Requirement Criticality Rating

Rating	Criticality Criteria
1	Prevent the accomplishment of an essential capability or jeopardize safety, security, or other requirement designated as "critical". Results in **mission failure or the inability to meet a primary mission objective.**
2	Adversely affect the accomplishment of an essential Mission capability and no workaround solution is known or adversely affect technical, cost, or schedule risks to the project or to life-cycle support of the system, and **no workaround solution is known.**
3	Adversely affect the accomplishment of an essential Mission capability but a workaround solution is known or adversely affect technical, cost, or schedule risks to the project or the life-cycle support of the system, but **a workaround solution is known.**
4	Result in **user or operator inconvenience or annoyance** but does not prevent the accomplishment of the responsibilities of those personnel.
5	Any other affect, usually **a bug with no impact**.

The ratings range from 1 (most critical) to 5 (least critical) depending on the criticality of the functionality required in the Spacecraft Software for the target mission. Requirements rated as a 1 or 2 are considered the most critical functionality to be provided by the Spacecraft Software to achieve mission success and to meet the mission's objectives. Requirements rated as 3 through 5 are considered the least critical functionality to be provided by the Spacecraft Software to achieve mission success and to meet the mission's objectives.

The Van Allen Probes SSRV strategy varied according to the criticality rating assignment for each Spacecraft Software requirement. Assignment of a requirement criticality rating (1-5) was done by the Spacecraft Software Lead and approved by the Mission Software Systems Engineer. The criticality rating was stored as a readily accessible attribute (C1-C5) in the requirements management system (IBM Rational DOORS). The verification strategy stated that any requirement with a criticality rating of 1 or 2 (most critical) required positive and negative functional testing. Requirements with a criticality rating of 3 through 5 (least critical) could be considered for a less rigid verification approach (e.g., exploratory session-based tests). The main difference between the functional and exploratory test approach was the level of required test documentation. A test case specification mapped to critical requirements must contain the details (or steps) to be followed for execution of the test. In contrast, a test case specification mapped to the least critical requirements does not require pre-planned test steps at the same level of detail. The exploratory test steps will evolve during the test execution session and the results of each test execution will determine the next test step. In both cases, the test case specification was taken to a peer review for approval and each test case was mapped to at least one Spacecraft Software requirement. Adopting this test strategy for Van Allen Probes SSRV proved effective in finding issues, and meeting the planned budget and schedule milestones. Supporting issue metrics for Van Allen Probes Spacecraft Software for each test approach will be provided in Section 3.

Tracking Changes and Use of Metrics

Official releases of SSRV documentation for Van Allen

Probes were maintained using the Windchill's Product Management Lifecycle (PLM) configuration management tool. Baseline versions of the test plan and test specifications were created along with any revision to the baseline based on the 'as run' or snapshot in time of the test case specifications that were officially executed to verify each Spacecraft Software release.

JHU/APL's Spacecraft Software development process requires that the SSRV documentation (test plan and test specifications) be peer reviewed. Issues resulting from each peer review are to be recorded, resolved, and approved by the person who initiated the issue before performing an official test execution. For Van Allen Probes, issues collected from test plan and test specifications peer reviews were tracked to closure by the technical test lead with an Excel spreadsheet. Memos were generated to document minutes for each peer review and to document associated issue resolutions along with concurrence provided by the person who initiated the issue. Copies of these memos were placed in the PLM system.

The Van Allen Probes SSRV process used Spacecraft Software issue metrics to monitor growth and to focus regression test for each Spacecraft Software release. The Spacecraft Software issue management process, the tracking tool (Atlassian JIRA), and the use of resultant metrics in the SSRV program will be discussed in detail in Section 3.

Development Team Involvement and Communications

Communication with the Spacecraft Software team was effective throughout the Van Allen Probes SSRV program. The SSRV team met on a weekly basis to discuss test status and to report any roadblocks or software-related questions that needed to be resolved in order to keep the SSRV team moving in a forward direction. The Spacecraft Software Lead was invited (and regularly attended) the weekly SSRV team meetings and was an active participant in the discussions. If questions surfaced between the weekly meetings that inhibited SSRV progress, the technical test lead and/or test engineer would initiate communication with the Spacecraft Software Lead or responsible developer through email communications. Ad hoc face to face communication during high-pressure development times was discouraged as it was disruptive to the developers. The developer who was responsible for the Spacecraft Software component, and the Spacecraft Software Lead were invited to test specification peer reviews.

Release notes containing valuable information on the use of the delivered Spacecraft Software functionality and any resolutions to reported issues (if applicable) were delivered to the SSRV team. Inherent to the development process and schedule constraints, the lag time between a Spacecraft Software release and updated user documentation made it extremely important to maintain effective open communication between the Van Allen Probes SSRV and development teams.

Test Effort Leverage

An objective of our SSRV program is to add value to the overall Spacecraft Software verification and validation program. An important aspect of the SSRV process used for Van Allen Probes was to leverage on the verification performed by the development team to reduce duplication of effort. Many scripts were developed and used for unit and system tests. Capitalizing on reuse of these existing scripts, by extending the scripts, was essential for cost effective and efficient SSRV.

In addition, combining the test scripts developed by both teams forms a comprehensive regression test suite that can be used to build and to maintain the necessary confidence level in Spacecraft Software to reduce risk to the mission.

The Van Allen Probes Spacecraft Software development team implemented a *Test Framework* [4] during the later stage of system-level testing. The *Test Framework* was implemented for all system-level test script development. The test scripts developed during the SSRV program were merged, after formal test execution, with the system-level tests to form a comprehensive regression test suite. One advantage of the comprehensive regression test suite was the automation feature. The comprehensive regression test suite is currently being used to verify Van Allen Probes post-launch Spacecraft Software releases. Details of the Van Allen Probes *Test Framework* and the organization of the comprehensive test suite can be found in [4].

Plan for Test Reuse

Several components of the Van Allen Probes Spacecraft Software to support C&DH functionality were reused from heritage missions. Van Allen Probes Spacecraft Software was designed to work with a different architecture (NASA Goddard's cFE) than our heritage missions with the objective to reuse the C&DH CSCI on future missions. Code changes were sprinkled throughout the C&DH components to support cFE libraries. The Van Allen Probes test designs that were developed to verify the Spacecraft Software C&DH components were also designed to promote reuse. The plan is to use these existing test designs for requirements verification of the corresponding Solar Probe Plus Spacecraft Software C&DH components.

Conventions for SSRV test designs for Van Allen Probes were established to maximize the potential for reuse on future missions. Operational details of the test simulations environment were centralized in a section of the test specifications and then used as a reference point in all test cases. Centralizing the test environment details eased maintenance and replacement of the test step details as they evolved during the SSRV program. Adding a title to describe each test step using a feature of the requirements management tool promoted viewing test descriptions at a high-level without viewing the supporting mission-specific details. Use of these test design conventions has resulted in a high-level of reuse in the test case specifications for the core C&DH components being used for the Solar Probe Plus Spacecraft Software.

Van Allen Probes was the first JHU/APL mission to use Telemetry-West L3 InControl for the core ground software component. The InControl software required use of a new scripting language as compared to heritage missions. There was a steep learning curve endured by the SSRV team and no reuse of heritage mission test scripts. The advantage to the SSRV program for Solar Probe Plus is the test scripts developed for Van Allen Probes can be reused.

Plan Waiting Time

As mentioned earlier in this paper, SSRV is the last step in our development process before a Spacecraft Software release is delivered to our user community (Integration and Test and Mission Operations teams). Test designs and peer reviews, dry run executions and formal test executions are time and resource demanding. Occasionally there is lag time in the deliveries of Spacecraft Software, Ground Software and the test environment features to support test executions. Once the required software deliveries and test resources are made available, there is limited time available for the SSRV team to use the test environment to prepare and to execute the tests. On Van Allen Probes, the strategy was to staff test engineers at a steady level and then to switch focus (as needed) on tasks related to different Spacecraft Software releases. Tasks alternated between test design preparation and peer reviews for future Spacecraft Software releases, and test script preparation and test executions for the current Spacecraft Software release. A key element to the success of this alternating task strategy was assignment of multiple Spacecraft Software components to each test engineer. In theory, test engineers should be able to keep moving forward on SSRV activities at all times during the SSRV program to reduce "waiting" time. The alternate task strategy did prove to be efficient in reducing "waiting time" during the Van Allen Probes SSRV program and was effective in achieving schedule milestones.

3. Use of Metrics in SSRV Process

Metrics collection can be a valuable tool used in the SSRV process. Metrics provide a vehicle to assess a confidence level in Spacecraft Software and also to provide a mechanism to improve requirements verification. However, metrics are only valuable when the data are collected properly and evaluated with the intent to use the metrics to improve the SSRV process; whereby helping an organization meet its objectives. In SSRV, collecting metrics on issues (defects) is a valuable endeavor as it provides insight into how the Spacecraft Software is maturing and stabilizing.

The Van Allen Probes Spacecraft Software development team used a COTS issue tracking tool called Atlassian JIRA. Each issue was classified with a severity classification and a type. Table 7 in Section 2 of this paper defines the severity classifications used for each issue in the same way it defines the criticality rating for each Spacecraft Software requirement. An issue "type" was classified as follows: software bug, planned development, or documentation. The combination of issue severity and issue type provided the data necessary to evaluate each issue to determine the prioritization for resolving the issue in a future Spacecraft Software release.

In addition to the severity and the type classifications, each issue found during the development process had the option to be tracked with the following data:

- Date Issue Found
- Test Case ID - a uniquely coded name used to track formal Acceptance Test Cases
- Computer Software Component (CSC) (or component) with its corresponding Computer Software Configuration Item (CSCI)
- SW Release – version number where the issue was found
- Detecting Activity (development, system test, acceptance test, Integration & Test (I&T), Mission Simulations (MSIM), Internal Independent Verification and Verification (IV&V), External IV&V, Mini-MOC, and Operations).

Though these additional details of a JIRA issue were not required when submitting a new Van Allen Probes issue, these fields are extremely useful for extracting metrics which may help improve the SSRV process.

Figure 1 illustrates the trend of the issues found per Spacecraft Software Release during the Van Allen Probes SSRV program. It should be noted that there were multiple minor releases that resolved issues and for simplicity these issues are bundled and indicated in each major Spacecraft Software Release depicted in Figure 1.

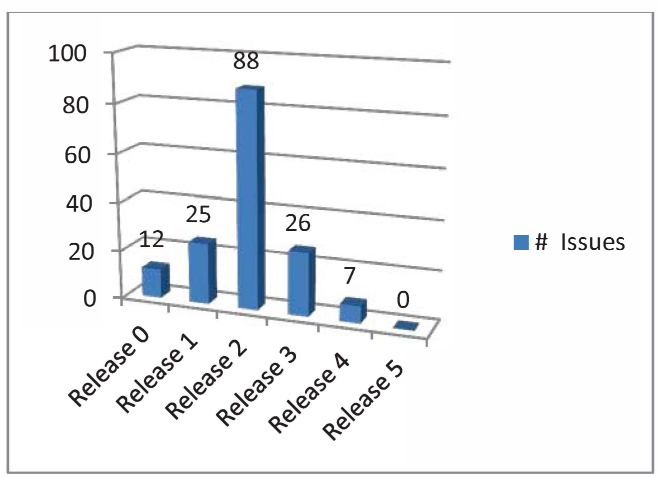


Figure 1. Issues by Release

During the development of the Van Allen Probes Spacecraft Software, the metrics obtained from the JIRA issue tracking tool show that the majority of issues were uncovered in Release 2. Release 2 of the Van Allen Probes Spacecraft Software was designated for use during I&T; thus, the bulk of the C&DH functionality was implemented and delivered for requirements verification (or Acceptance Test). Figure 1 also shows that as issue resolutions and the remaining functionality were delivered, fewer and fewer issues were found in subsequent releases of Spacecraft Software. The Van Allen Probes Spacecraft Software launch Release 4.x was delivered with zero defects being found during testing (Release 4.x was a regression test). To simplify the analysis of the metrics, it should be noted that an "x" indicates an incremental (minor) release within the development life cycle. The "x" incremental releases are discussed in this paper as if they were all verified under the primary release number.

In addition to viewing the issues as Total Issues Found per Release, the Issue Data was broken down further and is represented in Figure 2.

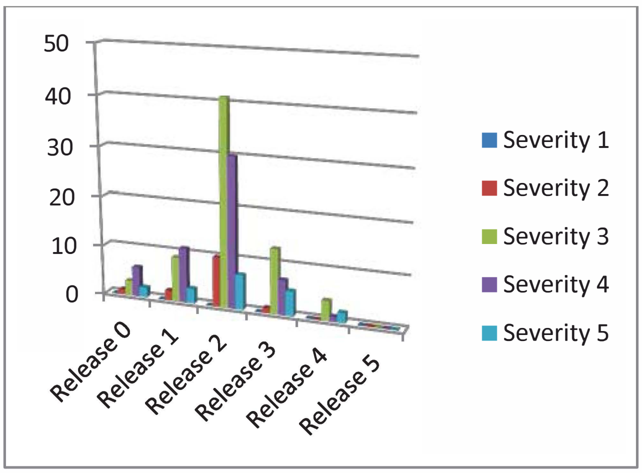


Figure 2. Issues by Release and Severity

The Issue Data in Figure 2 illustrates the total number of issues found in each Spacecraft Software Release and represents the breakdown of issues by severity. As the SSRV program moved forward from one Spacecraft Software Release to the next, the number of higher severity issues diminished; whereby providing further data which is evidence that the Spacecraft Software was maturing and becoming more stable for final operational deployment. The trends show as the number of releases are delivered, the number and severity of the issues diminishes as well. As the Issue Data illustrated, there are no high severity (Severity 1 or Severity 2) issues remaining open in the issue tracking system.

In the 2005 SSRV Study Recommendations for Future Missions listed in Table 5, a recommendation was made to incorporate Exploratory Testing in the SSRV process for verification of Spacecraft Software requirements rated with a lower criticality (Severity 3 through Severity 5). Functional and Exploratory Testing were both used during the SSRV program on Van Allen Probes. The Other Tests category includes the system-level tests conducted by the development team, scenario tests conducted by the Integration and Test team, mission simulation tests conducted by the Mission Operations team and various other tests conducted by the subsystem teams. Figure 3 illustrates the total number of issues found by applying each testing type (Functional Tests, Exploratory Tests, Other Tests).

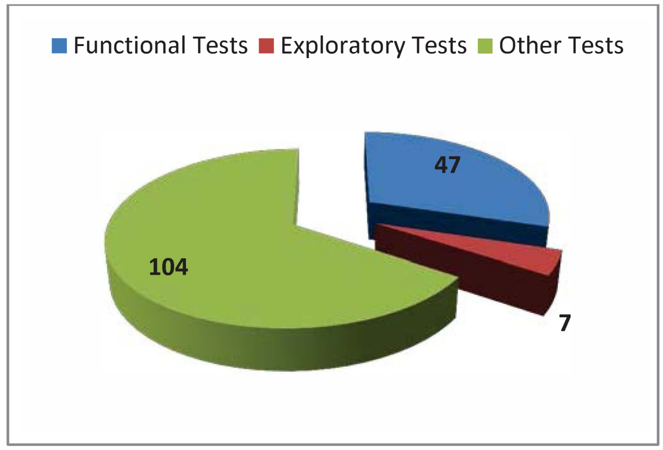


Figure 3. Issues Detected by Test Type

The Issue Data presented in Figure 3 can be broken down further. Figure 4 illustrates how issues of varying severity levels were uncovered using Functional Tests and Exploratory Tests.

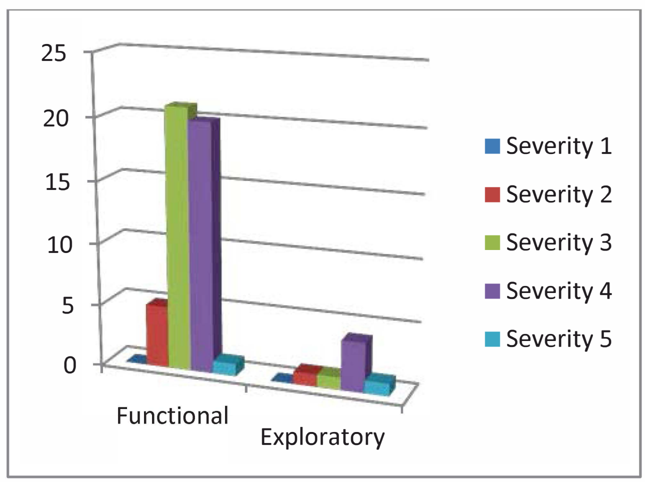


Figure 4. Issues Detected by Test Type and Severity

To put the Issue Data presented in Figure 4 into perspective, the total number of Functional and Exploratory Tests was 87. 72 of the 87 total tests were Functional Tests and the remaining 15 tests were Exploratory Tests. Functional Tests were designed to verify functionality most critical to the success of the mission and Exploratory Tests were used to verify functionality not critical to the success of the mission. Proportionally, the Functional Tests uncovered more issues overall with several issues having a high severity. Exploratory Tests also proved successful in uncovering issues, especially with one Severity 2 Issue being exposed in the Spacecraft Software.

Metrics for Van Allen Probes Spacecraft Software were also compiled using the Detecting Activity data provided in the JIRA issue tracking tool and is depicted in Figure 5.

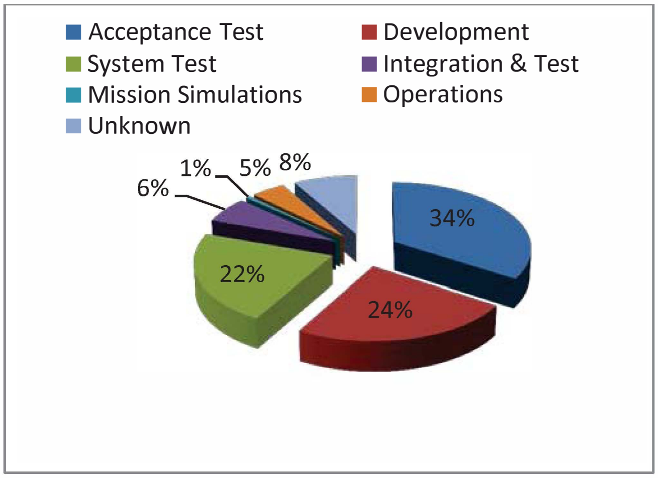


Figure 5. Percentage of Issues Found by Each Detecting Activity

Figure 5 illustrates the cumulative issues documented for all the Van Allen Probe Spacecraft Software releases for each Detecting Activity (Acceptance Test, System Test, Mission Simulations, Development, Integration and Test, Operations and Unknown). As the data clearly illustrates, the SSRV process was effective in uncovering more issues for Van Allen Probes Spacecraft Software than any other Detecting Activity.

The metrics tabulated for Van Allen Probes Spacecraft Software demonstrate how the data can be used in the SSRV process to track the stability and maturity of Spacecraft Software through development. In addition, these metrics provide the ability to perform a Spacecraft Software confidence level assessment.

For future missions, accurate and additional metrics would be beneficial:

- to focus regression verification,
- to determine supplemental testing, and
- to gain greater insight into the Spacecraft Software's level of readiness.

Several recommendations for metric collection for future missions are listed in Table 8.

Table 8. Recommendations for Future Metrics

Future Metric Recommendations
Make it mandatory that all fields be populated for each issue in the tracking tool (e.g., CSCI and component).
Issues found during SSRV should have a designated Test Case Identifier (if available).

Metric information extracted from the issue tracking tool is a valuable asset that can be utilized in the SSRV process. However, all data fields for each issue must be present in the issue tracking tool in order to make accurate assessments of the maturity of Spacecraft Software at any given time during development or post launch. To fully populate all the fields for each issue, it is necessary for several people to be responsible and involved in the issue tracking process. The person who submits the issue may have limited insight into the Spacecraft Software organization therefore it is the responsibility of the development team to fill in (or correct) any information relating to the component. Software Quality Assurance involvement should include monitoring the issues for document completeness throughout the issue's life cycle.

Accurate metrics throughout the development of Spacecraft Software can also provide information on possible areas of improvement which may be needed to the SSRV process. Metrics provide insight to help the SSRV team identify functionality (CSCI and component name) that may require stronger regression test and/or supplemental exploratory testing (e.g., scenario-based) in future releases. Adding CSCI and component level details provides a metric to further evaluate stability of specific functional areas.

4. Moving forward on Solar Probe Plus

Van Allen Probes provided the opportunity to implement many of the recommendations resulting from the 2005 SSRV Study. In Section 1, Tables 1, 2 and 3 presented system and Spacecraft Software attributes on four JHU/APL supported NASA missions and compared these missions with Van Allen Probes and Solar Probe Plus. Overall the complexity of the Van Allen Probes Spacecraft Software was simpler than the previous four missions and the current mission, Solar Probe Plus. For Solar Probe Plus, our evidence to the date of this paper is reflective that our development process for writing and organizing Spacecraft Software requirements has evolved since 2001. In parallel, our SSRV process has evolved and proved successful in verifying Spacecraft Software requirements and delivering a high level of confidence in our Spacecraft Software as we move forward on Solar Probe Plus.

Criticality ratings and assignment to each Spacecraft Software requirement proved valuable for use in the SSRV process implemented for Van Allen Probes. Using the requirement's criticality assignment resulted in an effective selection and application of an alternate test strategy. As a result of the exploratory approach to verification, several issues were uncovered with one issue being critical to mission success. Use of the exploratory approach for verification of requirements with lesser criticality uncovered issues in a timely manner which contributed to the Van Allen Probes SSRV program meeting schedule milestones. Building and retaining a core test team throughout the duration of the SSRV program proved worthwhile in reducing the time spent by the SSRV and development teams to train new test engineers. Establishment of conventions for documentation and organization of test designs reduced the maintenance burden and resulted in a high level of reuse for core C&DH components being used for Solar Probe Plus Spacecraft Software.

The transformation of SSRV scripts to implement a *test framework* and the consolidation of the scripts with system-level test scripts enabled automated regression test. Use of the regression test suite proved cost effective in verification and in assessment of a high confidence level in post launch Van Allen Probes Spacecraft Software releases that addressed new software functionality. Reuse of Van Allen Probes SSRV test designs, *test framework* and test scripts for core C&DH components is expected to benefit the Solar Probe Plus SSRV program. The *test framework* will be integrated into the SSRV process for Solar Probe Plus as a result of the Van Allen Probes experience.

New COTS tools were introduced and used in the Van Allen Probes SSRV process. Improvements were made to track test documentation and test results. As a result of this tracking, compilation of test reports and archiving of test artifacts were made more efficient. These COTS tools will be integrated into the SSRV process for Solar Probe Plus.

Section 2 of this paper described the improvements to the SSRV process that were implemented on Van Allen Probes in response to the 2005 SSRV Study. Table 9 lists the SSRV program lessons learned for Van Allen Probes.

Table 9. Van Allen Probes SSRV Lessons Learned

Van Allen Probes Lesson Learned
Use of mission-specific command and telemetry definition mnemonics in test case specifications is time consuming to maintain over the course of the SSRV program and reduces the level of reuse in test design.
Incomplete user documentation for a Spacecraft Software release impacts test team productivity and increases the need for effective communications with developers.
Prioritizing requirements by criticality and applying exploratory test strategy for verification of functionality implemented for requirements with lesser criticality was an effective and efficient way to uncover Spacecraft Software issues.
Following conventions (e.g., isolation of test environment setups) for test script development enhances maintenance through the course of the SSRV program.
Retaining a core test team through the duration of the SSRV program reduced the training burden on the test lead and test engineers.

In conclusion, recommendations resulted from the Van Allen Probes SSRV program and are listed in Table 10.

Table 10. SSRV Recommendations for Solar Probe Plus

Recommendations
Simplify test designs by using more generic detail in test steps (e.g., isolate command sequences and telemetry definitions to test scripts).
Establish and enforce test design conventions (e.g., use command and telemetry descriptions instead of mnemonics) to optimize test design reuse.
Use the test framework from the start of the SSRV program to support test script generation and to enable test automation.
Automate verification steps (e.g., cFE error event verification) as much as possible to reduce manual checks and halts in test scripts.
Continue test execution (when possible) to the end of the test script by capturing errors in logs.
Isolate test environment (ground and test simulation) details in test designs and test scripts to ease maintenance during development of the test environment (e.g., script libraries).
Merge system-level tests and requirements verification tests for each Spacecraft Software component as early as possible during development and use test suite for system-level regression test.
Reduction in the lag time between a Spacecraft Software release and supporting user documentation would ease the communication burden on the development team to provide information to assist the SSRV team in test design and script development.
Use Spacecraft Software issue-related metrics as an indicator to determine and monitor maturity and confidence level of Spacecraft Software during development.
Use Spacecraft Software issue-related metrics as a guide to determine the functional areas where supplemental scenario-based exploratory type tests can be used in the SSRV program.

The recommendations in Table 10 can be used to further improve the efficiency and effectiveness of JHU/APL Space Department's SSRV process to accomplish our main objective to ensure a high level of confidence in Spacecraft Software.

REFERENCES

[1] D.A. Clancy, B.A. Clyde., M.A. Mirantes, "The Evolution of a Test Process for Spacecraft Software", IEEE Aerospace Conference, Big Sky, MT, 2005.

[2] Johns Hopkins University, Applied Physics Laboratory Van Allen Probes web site: http://vanallenprobes.jhuapl.edu

[3] Johns Hopkins University, Applied Physics Laboratory Solar Probe Plus web site: http://solarprobe.jhuapl.edu

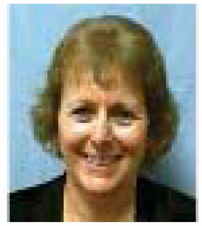
[4] M.N. Lovellette, J. White, "Test Like You Fly" Confidence Building for Complex Systems, IEEE Aerospace Conference, Big Sky, MT, 2005.

[5] T. Adrian Hill, W. Mark Reid, Kristin A. Wortman; "Command and Data Handling Flight Software Test Framework: A Radiation Belt Storm Probes Practice", IEEE Aerospace Conference, Big Sky, MT, 2013.

BIOGRAPHIES

M. Annette Mirantes *is a member of the Senior Professional Staff at The Johns Hopkins University Applied Physics Laboratory. She received a B.S. in Electrical Engineering from Purdue University and an M.S. in Systems Software Engineering from George Mason University. She previously developed flight software for numerous missions at Orbital Sciences Corporation including the ORBCOMM constellation, the Far Ultraviolet Spectroscopic Explorer (FUSE) and the Solar Radiation and Climate Experiment (SORCE). At JHU/APL, she has been the Main Processor flight software lead for the MESSENGER program and most recently the Mission Software System Engineer for the Missile Defense Space Layer work in the National Security Space area. She has authored and co-authored numerous papers, including "The Evolution of a Test Process for Spacecraft Software" by Debbie Clancy, Brenda Clyde and Annette Mirantes, which won Best Paper/Presentation at the 2005 International Conference on Embedded Software Testing.*

Maria Spezio *received a B.S. in Information Systems from The University of Maryland, Baltimore County in 1989and a M.S. in Computer Systems Management from the University of Maryland, University College in 1995.She has been with the Johns Hopkins Applied Physics Laboratory for more than 12 years. She has been a test engineer on the NASA New Horizons, MESSENGER, STEREO, and most recently the RBSP missions focusing on software V&V. Prior to JHU/APL she worked as a test engineer at Commerce One, Verizon, Lockheed Martin Aerospace, and Computer Science Corporation.*



Kristin A. Wortman *is a member of the Senior Professional Staff at The Johns Hopkins University Applied Physics Laboratory. She received a B.S. in Computer and Information Science from University of Maryland University College, in 1990, an M.S. in Software Engineering, a joint degree from University of Maryland and University of Maryland University College in 2000. She has been with JHU/APL for more than 5 years. Currently she is the software acceptance test lead for Solar Probe Plus. Previously she was the software acceptance test lead for Radiation Belt Storm Probes. Prior to joining JHU/APL she was a senior computer scientist for Computer Sciences Corporation (CSC) for two tenures totally twenty-two years. She supported a team of NASA Goddard scientists and developed instrument flight and data analysis software for two STEREO spacecraft science instruments. When not working for CSC she was a senior computer scientist for Computer Sciences Corporation (CSC) for two tenures totally twenty-two years. She supported a team of NASA Goddard scientists and developed instrument flight and data analysis software for two STEREO spacecraft science instruments. When not working for CSC she was employed by several software companies as a developer for another twelve years supporting data processing and analysis software development for various NASA missions. She is also an adjunct associate professor for University of Maryland University College in the Computer and Mathematical Sciences Department Undergraduate Studies since January 2001.*