MAPSIT Special Action Team Report on Cartography and Image Processing Software at the Astrogeology Science Center

Ross A. Beyer
Sagan Center at the SETI Institute and NASA Ames Research Center
rbeyer@seti.org

K. Michael Aye
University of Colorado, Boulder

Nicholas M. Estes
Arizona State University

Carolyn Ernst
Johns Hopkins University Applied Physics Lab

Caleb Fassett
NASA Marshal Space Flight Center

Audrie Fennema
The University of Arizona

Dathon Golish
The University of Arizona

Cynthia Phillips
California Institute of Technology and the Jet Propulsion Laboratory

November 30, 2018

Contents

Chapter 1

Introduction

The National Aeronautics and Space Administration (NASA) Mapping and Planetary Spatial Infrastructure Team (MAPSIT) established a Special Action Team (SAT) consisting of the authors of this report. The objectives of this SAT were to execute a review of the present state of the U.S. Geological Survey's (USGS's) Integrated Software for Imagers and Spectrometers (ISIS) capabilities and software development. This SAT was to provide Findings relevant to future development of those software capabilities and others funded by NASA at the USGS's Astrogeology Science Center (ASC) in Flagstaff, Arizona.

The complete terms of reference for this SAT can be found in Appendix A on page 16.

The team met regularly from its initial teleconference on 26 June 2018 through the publication of this report, and had an in-person meeting at the ASC on 28-29 August 2018.

Approximately four years ago, under the auspices of the Planetary Cartography and Geologic Mapping Working Group (which has since dissolved, but whose mission carries on with MAPSIT), an External Technical Review Panel (ETRP) was formed. It reported Findings on the state of the ISIS software and ASC development practices (a copy of their Findings document can be found in Appendix B, on page 18).

As a part of the current SAT effort, we have reviewed the ETRP Findings. Some of those Findings have been addressed, and some were overcome by events. Specifically, the ASC has carried out successful remedies to the ETRP Present Development Methodology and Professional Development Findings, for which we commend them. Other ETRP Findings were also addressed, and we provide additional guidance on them in Findings 9, 12, and 23. We find that some of the ETRP Findings remain un-addressed, and we have revisited some of them (Findings 4, 8, 14, 15, and 22).

This SAT solicited feedback from the community of users and we summarize their responses in Appendix C on page 26.

Some Findings constitute tasks and activities that are important to the progress of the ASC software portfolio, which we think should be carried out within six months or a year. Where applicable, we have specified suggested milestones, and also summarize these suggested milestones in Appendix D on page 27. Other Findings are not about specific tasks, but their suggested remediation constitutes a set of guidelines or principles that should be adhered to. We have intentionally not ranked these Findings, and Findings without milestones are no less relevant than those with milestones.

It is important to note that this document and these Findings address the entire ASC software portfolio and are not limited to ISIS. In the past, the ISIS software was the primary development effort and primary software deliverable of the ASC, but that scope has broadened even over the last four years, and based on our conversations with ASC staff, that scope will continue to evolve. Except where these Findings are ISIS-specific, the intent is that they apply to all software development that the ASC performs.

Furthermore, this SAT understands that the ASC may be moving forward to discontinue use of the name ISIS for a variety of reasons. As future naming conventions have not yet been established, the terminology used in this document refers to both the current ISIS and any near-future incarnation.

Chapter 2

Findings

2.1 Planning

F1 Strategic Plan

We find that there is no publicly available long-term strategic plan for the ASC software portfolio.

An overall long-term publicly available strategic plan would support other Findings about community participation and transparency. It would would also help the ASC itself to understand the alignment between the direction of their software portfolio development and NASA missions, directions, and goals, and help them to ensure that their software portfolio is supporting the concepts of planetary spatial data infrastructure.

It would be reasonable for the ASC to develop such a plan within a year.

F2 Existing functionality and future deprecation decisions

We find that it is important to identify what the core ISIS functions are, to maintain existing versions of the software past the time when new versions are introduced, and to ensure that existing functionality not be deprecated unless alternate solutions are common, broadly accessible, and equally capable.

The SAT notes that significant changes are occurring or planned with the ISIS code base, including modularization of libraries, implementation of new interfaces, incorporation of new core libraries (e.g. ray tracing engines), etc. Given the broad user community, many of which have existing workflows and pipelines, we find that it will be important to maintain existing versions of the software for a period of time, even when new versions are introduced, until the user community has embraced the new versions.

In the context of libraries, it will be critical that core ISIS functions continue to exist and function as they do currently while their internal components are switched to use libraries. The definition of “core” should be defined by polling ASC developers, internal users, missions, and the community at large. It might be best to err on the side of a large pool of “core” functionalities, rather than one that is too limited. This would best smooth the transition to a library-based, modularized ISIS and toward an open-source community.

In the context of graphical user interfaces (GUIs), while there is broader expertise in the community outside of the ASC for creating GUIs that can be wrapped around core ISIS functionality, and the ASC may be considering this as an area from which to eventually step away from (so their resources can be deployed elsewhere), the existing level of GUI functionality (e.g. `qview`, `qmos`, etc.) should not be deprecated unless alternative solutions are equally capable.

2.2 Community Interaction and Documentation

F3 User base for ASC tools

We find that there is an uncertainty about whom the primary user base should be for the ASC software portfolio.

The ASC has largely developed ISIS with a posture clearly focused on experienced technical users comfortable in non-graphical environments, while providing some support for beginners in the form of tutorials and workshops. A continued primary focus on technical users is logical and appropriate, as this allows the ASC to focus their efforts on building software that solves technical problems needed to analyze planetary data that is unique to their skill set and essential for scientific analysis. For this reason, building new graphical applications that attempt to unify the current breadth of ISIS applications into a singular super-GUI that tries to offer a comprehensive interface for the majority of the existing tools for non-expert users is not an avenue that the ASC should pursue. The current level of per-application graphical interfaces is adequate.

The core functionality of ISIS should still ship as applications or tools usable without requiring extensive additional coding. In other words, just shipping libraries and application program interfaces (APIs) without some user applications is insufficient. Perhaps the Geospatial Data Abstraction Library (GDAL) example is a good one: primarily GDAL is a software library, and there are a large variety of operations that can be accomplished with the library for a user willing to write custom software. However, GDAL also ships with a number of command line applications that provide substantial functionality easily via the command line.

The ASC should continue its primary focus on technical users.

F4 Open Source, Community-Driven Development Model

We find that the ASC should continue to be the steward of ISIS and future planetary spatial data infrastructure software releases, but that more effort must be made to facilitate broad participation by external contributors.

The 2014 ETRP made a similar finding. While there is evidence that the ASC is moving in this direction, we expected to see more progress in this area after four years. The ASC development model should be as the steward of a set of open source packages, meaning that they should "embrace open exchange, collaborative participation, rapid prototyping, transparency, and community development" as was stated in their own aspirational documents in 2014.

We use the term 'steward' here instead of the term 'gatekeeper' used in the 2014 ETRP. 'Gatekeeper' implies defensiveness, while 'steward' invokes service to the community. While executive decisions about the course of the software should be made at the ASC, those decisions should be guided by a strong connection to the community of users. Recent discussions with the ASC indicate that they are considering assembling a technical committee consisting of external experts in order to address this need, we support and encourage that course of action.

F5 Community and User Support Forum

We find that there is currently no good mechanism in widespread use for user engagement with the ASC software portfolio and its developers.

The ASC had previously made available an online forum for its ISIS user community that served two functions. First, it enabled users to interact with ASC developers, ask questions, and report bugs. Second, it enabled experienced users to support fellow users (e.g. by suggesting tricks and strategies for certain tasks, answering novice questions, etc). This forum needed to be discontinued for understandable logistical reasons, but the unfortunate consequence has been a substantial reduction in communication both within the ISIS user community and between the users and the ASC developers. We note that the ASC has recently encouraged users to start using the GitHub Issues system to report issues with ISIS3, but it is not clear if the intended use is limited to bug-reporting, or if it is meant to also support discussions of broader topics.

The ASC should consider implementing a new platform to support user-user and user-developer interactions in the planetary data processing community and to advertise its use. Six months would be a reasonable timeline to develop such a platform. If such a platform cannot be hosted within the ASC compute environment for logistical reasons, ASC should implement it elsewhere (Google Groups e-mail list, GitHub Issues, many other possibilities exist with minimal overhead).

F6 Task-based Workflow Tutorials

We find that improved documentation for all programs would increase the utility of the software. Specifically, we find that more concrete, practical, modern workflow examples would be extremely useful for users.

While most ISIS applications have adequate documentation, the real power of using ISIS comes from chaining programs together into a workflow. While there are some tutorials that provide examples of these workflows, there are not enough to cover the broad applicability of ISIS. Additionally, more documentation about the general 'theory' of how and when to use certain options for a particular workflow or program would increase the utility of documentation over a simple, terse listing of program options.

F7 ISIS on the Windows Subsystem for Linux

We find that many users want a version of ISIS that is functional in a Windows environment, and that new technology available in Windows 10 (the Windows Subsystem for Linux) would allow it. Documentation for how to install ISIS on the Windows Subsystem for Linux is now referenced by the ASC.

It is important to note that there have been many requests over the years to provide a version of ISIS that runs on Windows. The historical problem with this request is that it would have required a substantial development effort to create a Windows-native version of ISIS that would have run on the Windows Command Prompt. Beyond that, creating a Windows program that could be double-clicked upon and operated from a comprehensive graphical interface–which is what many users mean when they request a Windows version–would require even more effort. Such an effort would have diverted resources from developing new planetary-science-relevant functionality, addressing bugs, and working on critical functionalities within the core codebase. We find that the historical decision not to develop a Windows version was correct, given the limitation of resources.

The new Windows subsystem for Linux allows ISIS (and its future modular descendants) to run on a Windows machine in an analogous manner to the way that it currently does on Apple Macintosh machines; not because a special new development effort was made to run ISIS on Macs, but because macOS provided a unix-like enough environment for ISIS to run on. The ASC is to be commended for publicly linking to a blog post which documents the process for getting ISIS installed and working in the Windows Subsystem for Linux within the ISIS documentation.

The ASC should incorporate the instructions from the external blog post to the ISIS documentation, though maintaining appropriate caveats about this installation method being not strongly supported is reasonable. As practicable, we encourage the ASC to test ISIS against and maintain at least installation support for ISIS on the Windows Subsystem for Linux going forward.

2.3 Core Functionality

F8 Code Modularity

We find that the ASC has not divided the ISIS functionality into a core, stand-alone library and a set of applications and utilities, as specified in the ETRP Findings. We find that developing such a strategy is of high priority and should be vigorously pursued.

We agree with the ETRP finding that this definition of a clear and concise API will encourage and aid outside developers using and contributing to the software codebase. Discussions with ASC indicate that they have developed a preliminary strategy for accomplishing this modularization effort, and the intent is to identify separable, foundational functionality within the codebase (camera models, SPICE interactions, etc.), write stand-alone libraries with a clean API to provide that functionality, and then replace the current functionality with calls to those APIs, which would address this ETRP finding.

The ASC is encouraged to develop a detailed, 5-year plan for this modularization including a specified order for which functionalities will be modularized and implementation milestones for that plan. One year is a reasonable timeframe to develop such a plan.

F9 Code Accessibility

We find that the ASC has placed their main development trunk for ISIS and most of their non-proprietary software development on the public GitHub site, as the 2014 ETRP finding on Code Accessibility recommended.

This placement allows the outside community to more actively participate and collaborate with ASC developers on the code. This is a major shift from their previous development methodology of ISIS, and they are to be commended for it.

We encourage the ASC to continue to develop all major, non-proprietary software projects in the open in this manner.

F10 Issue Tracking

We find that the issue-tracking system that was in place for some time at the ASC was opaque to external developers and users, and that their recent switch to GitHub Issues for ISIS is an improved issue-tracking and resolution system that will greatly benefit that project. However, not all ASC software projects do so.

Since ASC is using GitHub for development, using the GitHub Issues system would likely be the best solution. Doing so would reduce effort for all parties, because known issues can be found by simple search, which saves contributors from duplicating effort. Also, new error cases for known issues could be easily facilitated by adding to existing findable GitHub Issues. Additionally, the already opened GitHub pull requests are usually directly linked to issues being tracked. The sequence of opening GitHub Issues and solving them via GitHub pull requests is a well known and highly accepted procedure for software provenance.

The ASC should consider using the GitHub Issues system for issue-tracking with all ASC software portfolio projects (not just ISIS). Six months is a reasonable timeline to accomplish this.

F11 Software Development Standards

We find that the ASC does not have a set of consistent and logical software development standards for their software portfolio.

There are currently some loose standards and a style guide available for ISIS, but given the anticipated evolution of the ISIS codebase and other projects in the ASC software portfolio, a new comprehensive set of industry best-practice standards is needed. These standards are not just style guides, but also the guiding philosophies about how to write software for ASC projects. Once these standards are in place, all newly developed code should adhere to these standards. The existing large codebase is unlikely to conform to these new standards, and a massive ‘conversion’ project is not needed, but as older pieces of the codebase are modified and updated, they should be brought into conformance with the new standards. These standards must be followed by all external contributors, as well as ASC developers. The establishment of these standards will simplify maintenance, revision, and shared use of code, thereby supporting the move to an open source model.

Furthermore, a natural corollary of this Finding is an evaluation of how much of any particular codebase meets these established standards. The ASC should look into reporting a variety of code quality and coverage statistics for their software portfolio.

Establishing a set of such comprehensive software development standards and making them available to the community would support the move towards an open source model. A year is a reasonable timeframe to accomplish this.

F12 Universal Binary

We find that the ASC did not work to create a more universal binary under their old distribution system, but with the November 2018 move to their new `conda`-based distribution system, this goal is now realized.

The 2014 ETRP included a finding about creating a more universal binary distribution of ISIS. The concept was that such a ‘universal’ binary distribution would have fewer dependencies on the host system, at the cost of being larger to download, but with the benefit of being able to run on a wider variety of systems.

The new `conda`-based distribution system addresses the essential desire of the 2014 ETRP Finding to allow ISIS to be installed on a wider variety of systems. This is because much of the system-dependent library installation issues are handled by the `conda` system which allows installation on a much broader spectrum of systems than allowed under the previous binary distributions.

F13 Binary Installation

We find that the `rsync` installation process that had been in use for many years functioned well and was dependable for highly technical users, but that it had many pitfalls for less technical users. However, in November 2018, with ISIS 3.6.0, the ASC has removed it in favor of a `conda`-only installation process.

Under the `rsync` mechanism, warnings in the ISIS documentation about losing data when the installation would be performed incorrectly were frightening to potential new users. Incorrect use of the `rsync` mechanism caused accidental deletions or unintended data duplication for users. A Java-based graphical installer was supported in the past, but it is no longer functional and is not maintained. While that effort had the best of intentions, there are a number of practical reasons why it is not functional, and we are not suggesting that it be revived.

As of the ISIS 3.6.0 release, the ASC has discontinued the `rsync` mechanism for software download, and is using the `conda` package management system, making it a required installation in order to install ISIS. The new `conda` system does help users ensure that they get the right files in the right places, and that is ultimately a good thing. However, as it is a new installation system, more work needs to be done to streamline the user-experience. Such a mechanism could also enable installations of beta-versions or mission-specific versions for testing.

It is reasonable for the ASC to expect a larger number of user issues related to the new installation system. They should be especially vigilant for unforeseen issues that might cause difficulty with mission or instrument team installations, and be ready to support them. If absolutely necessary the ASC could fall back on the `rsync` mechanism.

F14 Development Builds

We find that the availability of development builds of ISIS (referred to as pre-release, weekly, nightly, edge or dev builds, not just the release-candidate builds that are available) continues to be limited.

The 2014 ETRP found that the existing weekly builds being generated at ASC should be released to external users; however, these builds are still not available. Access to such development builds allows users to verify bug fixes, investigate new functionality, and generally close the loop faster with developers leading to increased software quality and less pipeline downtime. The ASC stated that due to occasional proprietary code in the source tree, weekly builds cannot be automatically released and that work is ongoing via a new continuous integration platform to make development builds available to external users.

The ASC should consider how to make these development builds (at whatever cadence more frequent than official releases is practical) available in some form (`rsync`, `conda`, tarballs on GitHub, etc.). One year is a reasonable timeframe to accomplish this.

F15 Software Compilation

We find that there have been improvements to the ISIS build system, allowing external developers to compile the ISIS source code. However, it is still problematic for external developers to build and test the ISIS system.

The ASC has already opened up the development process for ISIS to be visible on GitHub and has stated that it wants to create an open-source community around the ASC software tools. For this to happen it is important that the general build process is stable and well documented, so that the community can participate, as indicated in a 2014 ETRP Finding.

There are many difficulties for compiling ISIS by external developers. The precise list of compiler flags and version requirements for different operating systems should be well documented. Additionally, availability of appropriate test data is needed, but seems to be partially addressed by the new ISIS 3.6.0 release which allows test data to be `rsync`ed. For successful integration and participation of an open-source community a test system that can run on a public continuous integration system like TravisCI or similar would be advantageous.

The ASC is encouraged to continue to work to make the primary development branch of ISIS and other ASC software easy to compile by external developers.

F16 Software Optimization

We find that improving current application performance, where practical, is warranted.

Applications with performance issues (e.g. `cam2map`) should be investigated and identified. Efforts should focus on applications with the greatest opportunity for performance improvement, prioritizing those that can be easily or quickly fixed. Planning for growth and performance should be a consideration in all future development.

2.4 Methodology

F17 Support for Active Missions

We find that the ASC's role in NASA mission support has been a cornerstone of the planetary science community, and it should continue and improve.

No other organization is in a position to provide live development and processing support to the wide variety of planetary missions being flown. This has long been one of ASC's most visible roles. The ability to continue to support mission and instrument teams depends on improving the responsiveness that ASC is able to offer those active teams. Early involvement and consistent communication with mission and instrument teams will help to maintain this support. Having a technical contact role between ASC and the team, rather than just purely scientific or administrative contacts, is crucial to this communication. We note that the open source framework toward which ISIS is headed would help to ease this tension, as mission and instrument teams would be able to develop and apply hot-fixes, or even additional features, themselves, with the understanding that those changes could be integrated back into the ISIS codebase. That framework requires clearly defined guidelines for code development, submission, and review to reduce friction for both ASC and mission developers as illustrated in other Findings in this document.

The ASC is encouraged to develop more flexible approaches to dealing with mission and instrument needs by clearly communicating to missions regarding scheduling and other resource constraints that allow both parties to plan and adequately budget for mission support.

NASA should consider ways to encourage and enforce more planning with respect to planetary spatial data infrastructure by mission and instrument teams early in their formulation, which could stimulate the formation of partnerships with the ASC (or other technical providers) earlier in their lifecycle.

F18 Support for small irregularly shaped bodies

We find that support for small, irregularly shaped bodies in the ASC software suite is important to the planetary science community, given the increasing number of current and pending small bodies missions.

That work has successfully started with the implementation of support for the Navigation and Ancillary Information Facility 's (NAIF's) Digital Shape Kernel (DSK), as well as additional ray tracing engines (Embree, Bullet), fundamentally enabling active missions. However that support currently has a number of known issues, including efficiency, accuracy, occlusion, limb regions, overhanging terrain, non-unique lat/lon points, additional projections for irregularly shaped bodies, and alternative shape model formats (e.g. `.obj`). We support the ASC's stated objective to expand their capabilities for small bodies. ASC is in a unique position, with its core set of expertise, to advance the community in these respects. We agree that coordination with missions, who may have active development of tools and techniques for processing images of small bodies, can help to make that effort more efficient. It is also important to note that there is expertise in the broader community for visualization of image data in 2D and 3D, e.g. geographic information systems (GISs), Java Mission-planning and Analysis for Remote Sensing (JMARS), Small Body Mapping Tool (SBMT), etc. The ASC does not need to reproduce that functionality. Given the variety of potential downstream users of small-body data that might be produced in ISIS, this is an area where export of data to standard 3D formats, as specified by a planetary spatial data infrastructure, is important.

F19 Automated image matching and feature recognition

We find that additional focus on improving the automated matching of datasets has clear value as datasets get larger.

ISIS currently has many powerful tools for building control networks manually and via area-based and feature-based image matching. The ASC has made investments in solutions and tools to improve control network creation and management with humans-in-the-loop (e.g. the Integrated Photogrammetric Control Environment, IPCE). We note that the optimal tool or workflow for automated matching in a given situation is often not evident to users at present. In addition, some of the tools that are presently released are difficult to use in practice (e.g. `findfeatures`). We recognize that the ASC already has some projects related to this finding underway, and encourage developments of improved automated matching capabilities in ISIS to continue.

F20 Broadening the SPICE web service

We find that it would be desirable to update existing ISIS routines that do not fully support the SPICE web service to function in all cases.

Currently, there is a limitation preventing the use of the SPICE web service for a few missions/instruments (e.g. MESSENGER, CASSINI, HiRISE) because instrument calibration routines rely on local SPICE kernels. For multiple reasons (user convenience, improving consistency of behavior across the ISIS code base, centralizing SPICE management, etc.), it would be an improvement to update existing routines to make the SPICE web service functional.

F21 Python API

We find that a Python API for ISIS would be of broad use by the community.

A Python API for ISIS was discussed during our on-site meeting at the ASC as one of the foreseen interfaces to the future ISIS core library. We encourage ASC to openly discuss the planning of this Python API soon, as the implementation of it will increase the ability of non-computer scientist application developers to create planetary science processing pipelines based on ISIS. Currently, developers rely on custom solutions, or third party Python wrappers (e.g. `pysis`) that create system calls to ISIS applications, and a fully supported Python API would be welcome.

2.5 Personnel

F22 Software Management Leadership

We find that the ASC has not identified a lead architect for the ISIS software base as specified in the 2014 ETRP.

We agree with the ETRP Finding that such a position is critical for this software, and the broader software suite that ASC envisions. We believe that this role could likewise be fulfilled by a steering committee with decision-making authority that meets regularly to guide development. A key component of this leadership is to ensure responsiveness and traceability to ASC and community scientific use cases, funded projects, and mission support.

The ASC should consider establishing a lead architect or steering committee for all cartography and image processing software development with overall technical responsibility for strategic direction of cartography and spatial data infrastructure software development. One year is a reasonable timeframe to accomplish this.

F23 Workload

We find that the ASC has followed the 2014 ETRP Finding on Workload by both hiring new developers and successfully outsourcing work to external contractors.

This mixed model of adding local talent and going to external groups has been a successful approach and should continue. It allows for expertise to be retained at the ASC. It also allows the ASC to be flexible if they either lack a particular expertise, or available on-site developer hours.

2.6 Oversight

F24 Continued Funding for Support and Maintenance

We find that continued funding for software support and maintenance are critical parts of ASC's role as stewards of ISIS and their NASA-funded software portfolio.

The initial lack of funding for software support in the FY2019 Inter-Agency Agreement (IAA) budget curtailed ASC's ability to acknowledge and address bugs submitted by the community. The FY2019 budget under the IAA agreement allowed only for maintenance activities, which we understand to be mission-supported updates. This left the community at large, outside of funded missions, without support for a major piece of software. We understand that this oversight has been corrected, but it was a troubling omission.

We encourage NASA and the ASC to view the ASC software portfolio as fundamental infrastructure for planetary science that requires ongoing support and maintenance.

F25 External review of ASC annual funding requests

We find that it would be valuable for NASA management of ASC software and development funding under the annual IAA to include some component of external review.

The review that we imagine would not require an extensive review process (i.e. an in-person panel). Instead, a process whereby one or two external reviewers examine the priorities and levels of effort proposed by the ASC would have potential to improve outcomes.

NASA should consider soliciting this type of external feedback on the same cadence as new work effort is proposed under the IAA.

Appendix A

Special Action Team (SAT) Terms of Reference

The NASA Mapping and Planetary Spatial Infrastructure Team (MAPSIT) will establish a Special Action Team (SAT) with the objective of executing a review of the present state of the U.S. Geological Survey's Integrated Software for Imagers and Spectrometers (ISIS) software capabilities and software development to provide findings relevant to future development of those software capabilities and others funded by NASA at the U.S. Geological Survey's (USGS's) Astrogeology Science Center (ASC).

This SAT will be composed of individuals with scientific programming experience drawn from academia, industry, and the US Federal Government. The membership will be established by the MAPSIT Discipline Scientist, Dr. Sarah Noble, and the Chair of MAPSIT, Dr. Jani Radebaugh.

The final deliverable will be a written report containing findings and where applicable, discrete suggestions for potential improvements pertaining to future cartography and image processing software development.

The panel may choose to meet in person at the ASC in Flagstaff, Arizona. The final deliverable will be due to the NASA MAPSIT Discipline Scientist and the MAPSIT Chair no later than 30 November 2018.

High Level Questions defined by MAPSIT

This SAT will be tasked with providing findings to MAPSIT addressing the following topics:

Core Needs: The SAT will assess what the core functionality of ISIS should be.

Personnel: The SAT will provide findings on the needed development expertise to facilitate future ISIS development.

Methodology: The SAT will assess the present development methodology of ISIS and will provide the ASC with a viewpoint regarding where it stands with respect to current industry best practices. This includes the approach to software task planning, the organization structure, the development philosophy, version control, bug tracking, quality assurance, and testing.

Documentation and Community Interaction: The SAT will assess the state of ISIS public user documentation and training aids and provide findings about whether the content reflects future community needs.

Strategic Planning: The SAT will consider the feasibility of future plans for ISIS development, particularly the relationship between the current version of ISIS 3 and the proposed ASC roadmap for future cartographic software development and capabilities.

User Growth: The SAT will evaluate the present impediments towards the use of ISIS on alternative scientific computing platforms, and its ability to integrate with other software systems (as a library or part of a tool chain).

Governance: SAT will assess whether alternate ISIS governance models (for example, a USGS-led consortium approach) are feasible and could increase the long-term sustainability of ISIS development.

Review of ETRP: Many of these topics were addressed by the 2014 ISIS External Technical Review Panel, and this SAT will review those findings and evaluate how well the USGS has met them, and whether those findings remain relevant.

Relevance to PSDI: The SAT will assess how ISIS and future software development and capabilities envisioned by the ASC are relevant to Planetary Spatial Data Infrastructure.

Appendix B

External Technical Review Panel (ETRP) Findings

The pages that follow are the Statement of Findings from the 2014 ETRP.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

SCIENCE MISSION DIRECTORATE

PLANETARY CARTOGRAPHY AND GEOLOGIC MAPPING
WORKING GROUP

IMAGING SOFTWARE FOR IMAGERS AND SPECTROMETERS
EXTERNAL TECHNICAL REVIEW PANEL

STATEMENT OF FINDINGS

3 September 2014

Contents

Background

The NASA Planetary Cartography and Geologic Mapping Working Group (PCGMWG) established an ISIS External Technical Review Panel (ISIS-ETRP) with the objective of executing an external review of the present state of the ISIS software and its development to provide findings relevant to future ISIS development activity in a timescale commensurate to facilitate evaluation of the 2014 Cartography proposal.

ISIS-ETRP was composed of individuals with scientific programming experience drawn from academia, industry, and the US Federal Government. The membership was established by the Chair of the PCGMWG in consultation with the PGG Discipline Scientist.

Membership

Dr. Samuel Lawrence, Arizona State University [PCGMWG Chair]

Dr. Ross Beyer, NASA Ames Research Center and the SETI Institute

Mr. Ernest Bowman-Cisneros, Arizona State University

Mr. Joe Plassman, University of Arizona

Mr. Scott Turner, the Johns Hopkins University

Mr. Nicholas M. Estes, Arizona State University

Terms of Reference

The ISIS-ETRP was tasked with providing findings to the PCGMWG addressing the following topics:

Core Needs: The ISIS-ETRP will assess what the USGS considers the core functionality of the ISIS software.

Personnel: The ISIS-ETRP will provide findings on the needed development expertise to facilitate future ISIS development.

Methodology: The ISIS-ETRP will assess the present development methodology of ISIS and will provide the Astrogeology Science Center (ASC) with a viewpoint regarding where the ASC stands with respect to current industry best practices. This includes the approach to software task planning, the organization structure, the development philosophy (for example, Agile, SCRUM), version control, and bug tracking, quality assurance, and testing.

Documentation and Community Interaction: The ISIS-ETRP will assess the state of ISIS public user documentation and training aids and provide findings about whether the content reflects future community needs.

Strategic Planning: ISIS-ETRP will consider the feasibility of future plans for ISIS development, particularly the relationship between ISIS 3.4.5 and the proposed ASC ISIS refactoring effort.

User Growth: the ISIS-ETRP will evaluate the present impediments towards the use of ISIS on alternative scientific computing platforms, such as Microsoft Windows.

Governance: ISIS-ETRP will assess whether alternate ISIS governance models (for example, a USGS-led consortium approach) are feasible and could increase the long-term sustainability of ISIS development.

Findings

The panel met in-person at the ASC in Flagstaff, Arizona, on 12 August 2014, and had a productive bilateral interchange with ASC personnel consisting of presentations and interactive responses on the topic of ISIS development. Based on these exchanges, the ISIS-ETRP has developed the following set of Findings for the full PCGMWG.

THE ISIS PROJECT

The ISIS Project exists as a system that includes the ASC, the end users, and the active flight missions that use this software for a continuum of activities of critical importance to the future of the United States that includes (but is not limited to) basic planetary science research, applied exploration science, and mission operations. A unified ISIS presence has value for the community over a fractured landscape of planetary image processing applications.

We find that the development process associated with the ISIS project would greatly benefit from, and therefore must be performed in, a unified fashion that ensures a consistent strategic direction, responds to core audience needs (i.e., science users and missions), and thus continues to facilitate these vital activities. The Lead Software Architect finding (below) must be considered an enabling step to ensure sustainable progress for the ISIS Project.

PRESENT DEVELOPMENT METHODOLOGY

ISIS-ETRP finds that significant progress has been made towards using industry best practices for software development. We particularly point to the use of Redmine and SVN as excellent starting points for future development activity. The establishment of an automated build system that executes regression tests is another step in the right direction.

GOVERNANCE OF ISIS

ISIS-ETRP finds that the ASC should continue to be the gatekeeper of ISIS releases, but must facilitate broader participation by external contributors (as the ASC intends). The goal must be, as stated in the 2014 Cartography proposal, that ISIS development must "...function as a true open source package, meaning we embrace open exchange, collaborative participation, rapid prototyping, transparency, and community development." The inclusion of outside developers as named Co-Investigators in the full FY2015 Cartography proposal, as well as the formalized external review of the proposed new ISIS architecture, is strongly encouraged. If the proposed re-architecting effort proceeds, an open development model must be adopted.

This finding requires that the Code Accessibility section (below) be fulfilled. The ASC should be able to accept external ticket submission and external patch submission via their tools.

Lead Software Architect

ISIS-ETRP finds that there must be an identified lead architect for the ISIS code base with overall technical responsibility for strategic direction of the ISIS project, a key component of whose task is to ensure responsiveness and traceability to ASC and community scientific use cases, funded projects, and mission support.

Substantial internal variability over time in various parts of the ISIS codebase has been observed. These inconsistencies make developing external code against ISIS very difficult. A lead technical architect would help ensure consistency across releases.

Code Architecture

ISIS-ETRP finds that the current ISIS functionality should be split into a core, stand-alone ISIS library and then a set of applications and utilities. The definition of a clear and concise API will encourage and aid outside developers using and contributing ISIS code. The facade development pattern could be used to aid this transition, but should not be seen as the final implementation.

Proposed "Refactoring" Project

The scope of work encompassed within the D-2 Task proposed in the FY15 Cartography proposal, which proposes to develop a plan to re-architect the ISIS software, is commensurate with a new version of ISIS and should properly be described as such.

Compiling and Distribution

It is currently very difficult for external developers to compile ISIS source code. The current build system compiles software in such a way that it is too specific to particular architectures.

Changes must be made to make external compilation of ISIS source code simpler for external developers. Changes can also be made to the build system that allow a more 'universal' binary download, such that there would only be one or two downloads for a particular operating system. This does increase the size of the binary download, but should significantly ease maintenance of the code and installation on diverse user systems.

These changes would also facilitate pre-compiled packages (e.g. Debian and RPM), reducing end-user burden.

Workload

ISIS-ETRP finds that there is insufficient person-power at the Survey to achieve the work that they are currently undertaking. This is the case even if replacements for their retirements can be found.

Additional staff members are required to complete the work currently supported by Cartography and other programs. Additional outside contractors or government agency employees should be considered as a temporary measure to ensure near-term milestones are met.

If the USGS wishes to successfully execute additional ambitious projects, even more personnel resources will be needed, and could be achieved through additional local hires, contractors, or other government agencies.

Hiring

ISIS-ETRP finds that the ASC should hire additional experienced software developers. Experienced developers bring in not only needed hands but also bring in additional software industry experience and methods. At least one of these should be an application program interface (API) specialist.

The ASC has stated that they are looking for candidates who blend cartography and software development skills, but it is the considered professional opinion of the ISIS-ETRP panel that the available pool of such individuals is vanishingly small. Instead, ISIS-ETRP finds that the ASC should concentrate on hiring individuals with quality software development experience, and has confidence that the scientific and cartography expertise already existing at the ASC will be sufficient.

Code Accessibility

ISIS-ETRP finds that public access to the active ISIS source tree would be beneficial. The ASC must participate in a more open, collaborative engagement with the outside community. Opening Redmine to the public is strongly encouraged and should be executed at the earliest practical opportunity.

The ASC has indicated that they would like to provide public access to their SVN repository on their DOI servers. However, history has shown that getting this kind of access from the DOI is a major stumbling block and given the ASC's desire to extend this public interaction to code submission, SVN repository access may not be possible even if public read-only access is made available.

Accordingly, ISIS-ETRP finds that hosting the code in an external, 3rd party, public repository would be a superior solution to attempting to negotiate access on DOI servers. One viable implementation pathway for this finding is to use the NASA-facilitated repositories on GitHub that are designed to store NASA-funded software projects. NASA has already begun hosting many of its software projects (including PDART-funded software products) on GitHub.com, and it is in keeping with industry best practices.

MAKE WEEKLY BUILDS ACCESSIBLE

ISIS-ETRP finds that the weekly “Production Builds” should be made available to the public. This would primarily be useful to a small group of users engaged in active missions, but could have other benefits, and should be straightforward for the ASC to accomplish.

PROFESSIONAL DEVELOPMENT

ISIS-ETRP finds that there are insufficient avenues for the professional development of software developers at the ASC. The software industry is a rapidly changing field, and the ASC developers need to be kept current.

This could be achieved by

- attending conferences
- offsite training
- onsite training (as the ASC has done)
- onsite one-day talks or presentations by outside experts
- developer exchange with other similar groups

ISIS-ETRP finds that developer exchanges between the ASC and other “superuser” groups would potentially be the most valuable short-term professional development activity, and the one with the most usefulness for the proposed re-architecture effort.

Appendix C

Community Feedback

The SAT solicited feedback from the general user community by creating an online form through which individuals could provide input into our process. There was a long-form text box soliciting questions for individuals for which the SAT should get answers, comments about their previous user experiences with ISIS, and suggestions for future NASA-funded software development in this area.

The SAT took all of the gathered information into account during our review, but did not specifically respond to each of the 30 responses.

In general, the public feedback was extremely positive, indicating that the software was very useful, and crucial for remote sensing data analysis from many spacecraft missions. The developers were praised for being responsive to user inquiry. There was also sentiment that funding and support should continue for the software development efforts at the ASC.

Several people asked for more instructive documentation and mission or instrument specific workflow tutorials, which we have addressed in Finding 6 on page 6.

Finally, there were requests for a Windows version of ISIS. This topic is specifically addressed in Finding 7 on page 6.

Appendix D

Suggested Milestones

Several of the Findings indicate a suggested timeline for accomplishment. We have gathered them together in this table to provide an overview:

	Finding	Milestone
F5	Community and User Support Forum	Summer 2019
F10	Issue Tracking	
F1	Strategic Plan	Winter 2019
F8	Code Modularity plan	
F11	Software Development Standards	
F14	Development Builds	
F22	Software Management Leadership	

Appendix E

Citing this Report

Please cite this report as:

Ross A. Beyer, K. Michael Aye, Nicholas M. Estes, Carolyn Ernst, Caleb Fassett, Audrie Fennema, Dathon Golish, and Cynthia Phillips. 2018. MAPSIT Special Action Team Report on Cartography and Image Processing Software at the Astrogeology Science Center. DOI: 10.5281/zenodo.2280984

Acronyms

API application program interface

ASC Astrogeology Science Center, https://astrogeology.usgs.gov

DSK Digital Shape Kernel

ETRP External Technical Review Panel

GDAL Geospatial Data Abstraction Library, https://gdal.org

GIS geographic information system

GUI graphical user interface

IAA Inter-Agency Agreement

ISIS Integrated Software for Imagers and Spectrometers, https://isis.astrogeology.usgs.gov

IPCE Integrated Photogrammetric Control Environment

JMARS Java Mission-planning and Analysis for Remote Sensing, https://jmars.asu.edu

MAPSIT Mapping and Planetary Spatial Infrastructure Team, https://www.lpi.usra.edu/mapsit

NAIF Navigation and Ancillary Information Facility, https://naif.jpl.nasa.gov

NASA National Aeronautics and Space Administration

SAT Special Action Team

SBMT Small Body Mapping Tool, http://sbmt.jhuapl.edu

USGS U.S. Geological Survey