Requirements Document v1.3

7 December 2018

Team Pypline

Sponsor Scott Akins
Dr. Jay Laura

Mentor Isaac Shaffer

Team Members Nicholas Anderson
Austin Collins
Connor Schwirian
Abdulaziz Zarie

Accepted as baseline requirements for the project:

For the client:

For the team:

----

1. Introduction **2**

2. Problem Statement **2**

3. Solution Vision **5**

4. Project Requirements **8**

4.1 Functional Requirements 8

4.2 Non-functional Requirements 11

4.3 Environmental Requirements 12

5. Potential Risks **13**

6. Project Plan **14**

7. Conclusion **16**

1. Introduction

The mission of the United States Geological Survey Astrogeology Center (USGS) is to further our knowledge of the solar system through their research into planetary cartography, geoscience, and remote sensing. The center was founded to assist with lunar mapping and astronaut training for the Apollo program and continues its work helping chart and understand all planetary bodies in the Solar System. Among their responsibilities are the development of a software toolkit for working with planetary images called the Integrated Software for Images and Spectrometers (ISIS), participation in mission planning, and the archive of all NASA planetary image data.

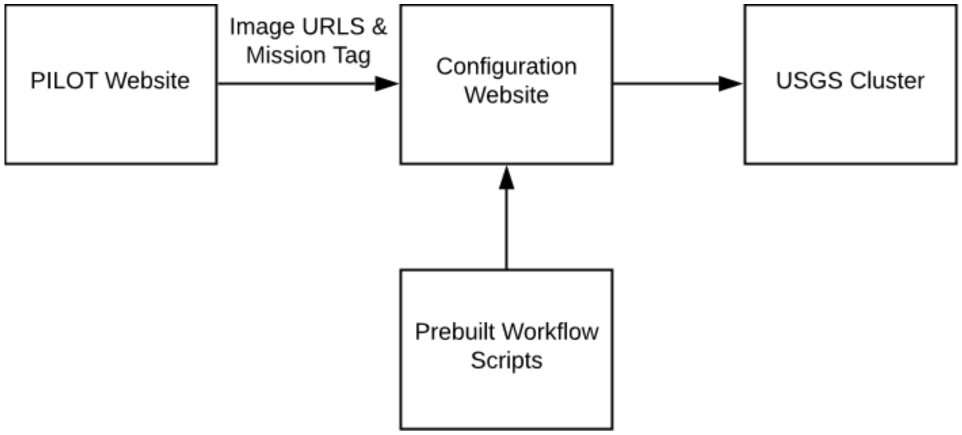
USGS maintains the Planetary Image Locator Tool (PILOT) website to facilitate the distribution of planetary image data to the scientific community and the general public. The site allows for the selecting of images from the planetary database and utilizes their in-house processing cluster to apply desired ISIS processing tools onto the images. These tools convert archived images into useable forms that can be applied to a variety of use cases: scientific research, mission planning, and even high school science projects. This entire process is encapsulated in the Map Projection On the Web (POW) pipeline.

Our sponsors, Scott Akins (USGS IT Specialist) and Dr. Jay Laura (USGS Research Scientist) work in a number of areas for USGS, including general software development, spatial data analytics, and planetary data infrastructure. They have brought Pyline on to upgrade the current POW implementation, which lies between the user selecting images on the PILOT website and processing on USGS's processing cluster.

2. Problem Statement

Given the importance of the planetary images in scientific research and future NASA mission planning, their distribution is an area that needs to be made as user-friendly as possible. The current imaging pipeline, seen in Figure 1, presents several issues from both the user and the USGS perspective.

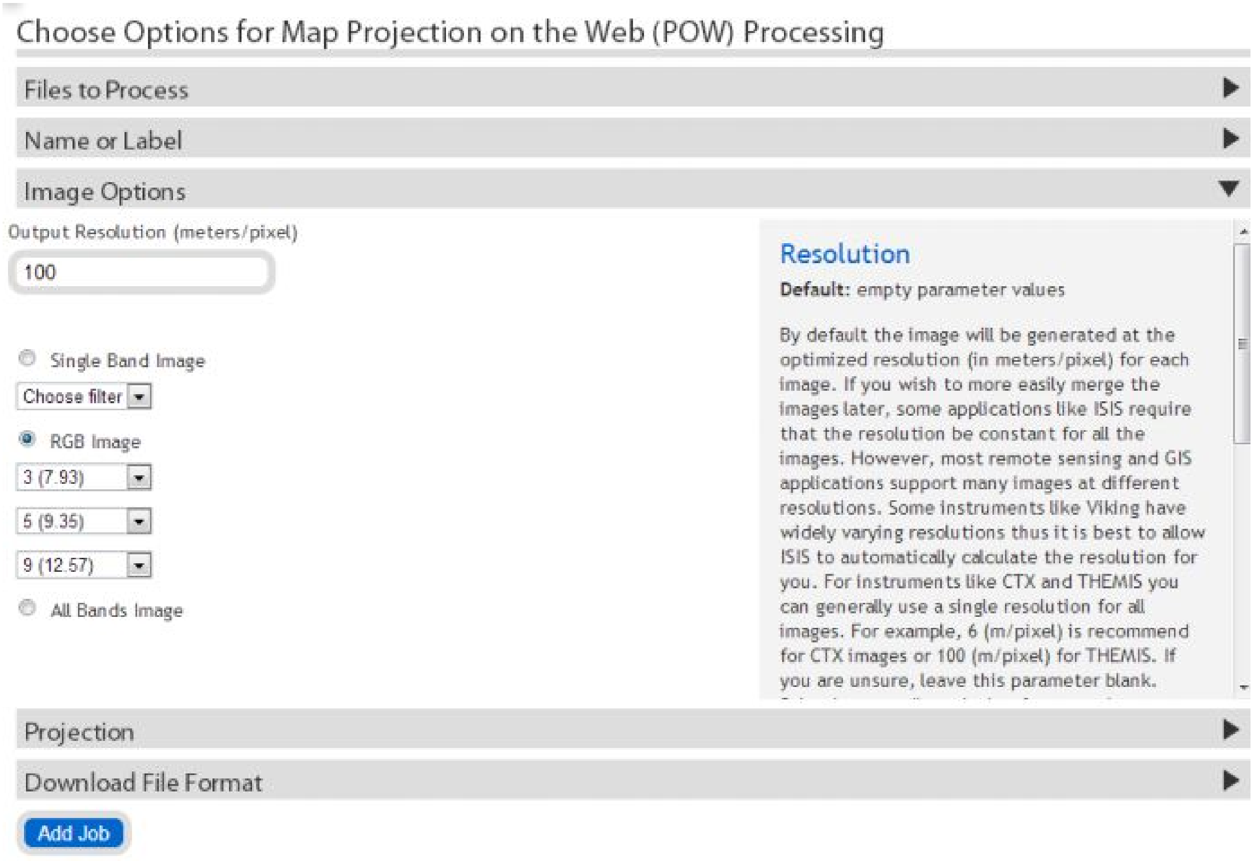
Figure 1. Current USGS Imaging Pipeline



First, the user logs in and selects images via the Planetary Image Locator Tool's (PILOT) web-based interface. PILOT allows for the selection of up to 50 images at a time from one of several dozen instruments. The instruments range from cameras aboard the current Mars Reconnaissance Orbiter (MRO) all the way back to the Mariner missions. Each instrument is assigned a unique tag that is used to identify it. Once the images are all selected, the user will choose to download the images directly or send to POW cluster for processing. If the user selects POW, the image URLs and their associated instrument tag is handed off from the PILOT website to a configuration page on the POW website, pictured in Figure 2. The user is then walked through a limited set of options, such as job name, what output size and format, and what type of map projection is preferred.

One of the main options is which prebuilt workflow they would like to use. These workflows are a set of ISIS commands to be performed on the images and the order they will be performed. A general workflow will take the image and convert from mission specific format into the ISIS compatible cube image format. Next, SPICE metadata (Spacecraft & Planetary ephemerides, Instrument C-matrix and Event kernels) will be add into the images. This data contains information about the spacecraft position, the direction the instrument is pointing, and any mission activities being performed during the image capture. Last a map projection is applied to correct for skewing within the images. The user selected options from PILOT and POW are used to generate an XML document summarizing the processing job. This document is sent to a processing script that translates the data into the appropriate ISIS commands and parameters. The job is then executed on a Moab/Torque processing cluster that USGS maintains. An email is sent to the user letting them know once the images are ready for download from the POW website.

Figure 2. The configuration page for the POW website



Choose Options for Map Projection on the Web (POW) Processing

Files to Process ▶

Name or Label ▶

Image Options ▼

Output Resolution (meters/pixel)

100

○ Single Band Image

Choose filter ▾

● RGB Image

3 (7.93) ▾

5 (9.35) ▾

9 (12.57) ▾

○ All Bands Image

Resolution

Default: empty parameter values

By default the image will be generated at the optimized resolution (in meters/pixel) for each image. If you wish to more easily merge the images later, some applications like ISIS require that the resolution be constant for all the images. However, most remote sensing and GIS applications support many images at different resolutions. Some instruments like Viking have widely varying resolutions thus it is best to allow ISIS to automatically calculate the resolution for you. For instruments like CTX and THEMIS you can generally use a single resolution for all images. For example, 6 (m/pixel) is recommend for CTX images or 100 (m/pixel) for THEMIS. If you are unsure, leave this parameter blank.

Projection ▶

Download File Format ▶

The main issue with the current processing pipeline is the reliance on prebuilt scripts to generate the workflows. These scripts present a problem for upkeep in addition to a lack of flexibility in presenting the user with the full ISIS toolkit. The main issues surrounding the current pipeline that we will be addressing are:

- **Lacks Flexibility**
  The POW website configuration page currently relies on static scripts to generate the workflows executed on the cluster. There is a large variety in what ISIS tools can be applied to an image depending on the instrument selected. The script contains many conditional statements to handle the myriad of possible options available. Currently, users can only select options that are built into the script and cannot exit the workflow early. Some feedback from users that USGS has received is a desire for the option to allow execution of only the first steps of the prebuilt (image conversion & SPICE data) workflow, which would allow full control of any additional processing. Users of the POW website have a variety of use for the generated images from scientific research to their use in art projects. It is difficult with a unified script to cover all potential uses. Last, the workflows

cannot be edited by the user to remove any unwanted optional steps. The current script tries to support the most common workflows but does not have the flexibility to generate all possible workflows that a user might need.

- **Script Upkeep**
  The ISIS toolkit is an active project that is constantly being updated for new missions. The reliance on prebuilt scripts means that any updates to the toolkit must be integrated into the script. Additionally, new missions require edits to the workflow script to make sure it is compatible with any mission specific requirements.

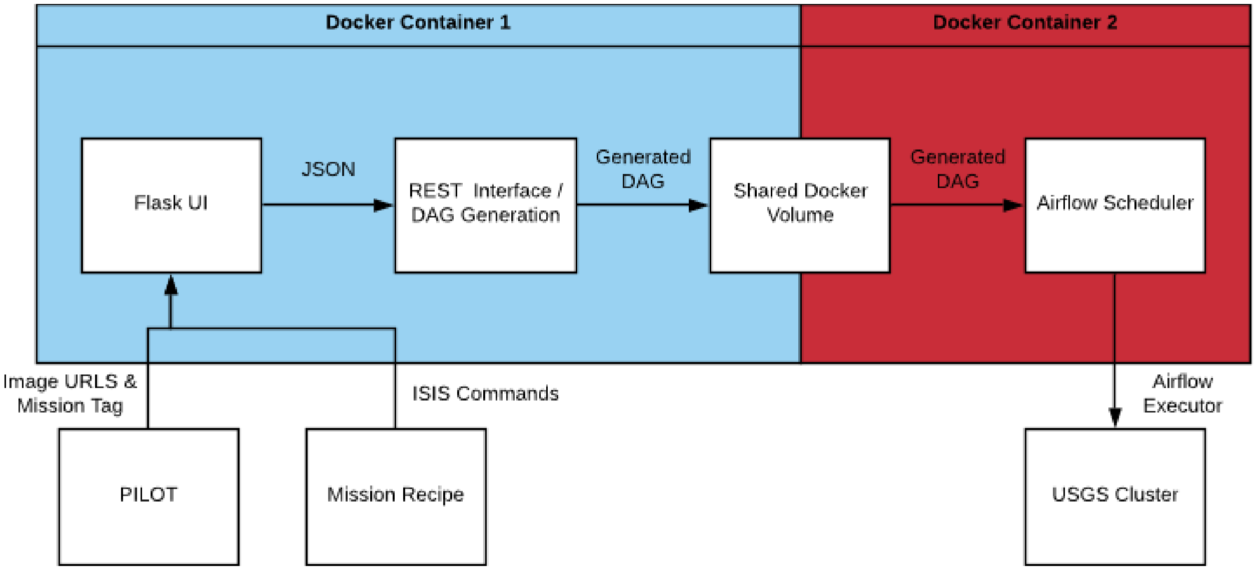
- **Download Only**
  The POW website is used extensively at USGS for their research. Currently, it lacks an option to specify an output location of the finalized images and requires employees to download the images. This is a waste of bandwidth when the images are located on servers already accessible by USGS staff. An ideal upgrade will allow users to specify their assigned folder on the USGS for image output.

The current image processing pipeline's lack of flexibility is preventing some workflows from being possible without the manual use of the ISIS toolkit. Replacement of the processing pipeline will help USGS and the end users alike.

3. Solution Vision

Pypline will replace the existing image processing pipeline with software to dynamically create job-specific workflows for execution on the USGS processing cluster. A new UI for presenting options to the user based on mission-specific recipes will allow for a greater flexibility in workflow creation. The user input will then be taken by the job-generating script to dynamically create a workflow unique to that user's processing job. Management of submitted jobs and their execution will be performed by a new workflow management tool deployed at USGS. The new pipeline image processing pipeline seen in Figure 3 will provide for the flexibility needed by USGS and end users.

Figure 3. Diagram of the replacement image processing pipeline



The heart of the new pipeline will be the new workflow management software and job scheduler. Airflow was selected by USGS to provide both of these services. Airflow allows for workflows to be broken down into the individual processing steps and submitted as python scripts that represent workflows as directed acyclic graphs (DAG). A DAG is comprised of individual tasks and the order in which they need to be executed. Airflow allows for these tasks to be built using already generated templates. Individual ISIS commands can be represented as a templated task to be included within a DAG script. This will allow for dynamic job generation that is unique to each image processing job. Additionally, the included Airflow UI will allow for viewing submitted jobs in DAG format, providing easier monitoring and troubleshooting of generated workflows.

A new UI will utilize the existing mission recipes to generate a visualization of the workflow. The recipes contain relevant mission specific mission data including what initial ISIS tools must be used to convert from mission image format to the ISIS cube format and what ISIS tools are available for that mission. From the UI, a user will be able to manipulate the workflow order, what toolkit steps are included, and what parameters those tools will use. That workflow will be translated to a JSON object for use in the generation of a DAG that is submitted to the Airflow scheduler. DAG generation will be handled by an interface capable of receiving the JSON object and outputting an Airflow compatible DAG to be executed.

Finally, all components of the final solution will be containerized in order to allow for easier maintenance post project completion. Application containerization allows for deployment and isolated execution of applications through OS-level virtualization without the overhead of running an entire virtual machine. The replacement pipeline will rectify issues seen in the current solution with these key features:

- **Dynamic Workflows**
  The flexibility issues seen in the current pipeline will be solved through the elimination of prebuilt workflow scripts and creation of dynamically generated workflows. This will allow for the full menu of relevant image processing options to be presented to the user. The new UI will allow for workflows to be visually manipulated and provide a better overall user experience. Additionally, it will provide end users with the ability to stop workflows at any step in the previous pre-generated workflows or eliminate individual steps within them. Overall the new pipeline will offer an increased ability for users to tailor the output images for their needs.

- **Easier Maintenance**
  The new pipeline will separate out the portions of the pipeline potentially requiring maintenance to allow for easier upkeep of the pipeline. Specifics to the mission will be contained within the instrument recipes and the implementation of specific ISIS tools will be implemented as templated Airflow tasks. This isolation will support the maintenance of existing mission data in addition to the inclusion of future missions over the lifetime of the new pipeline.

- **Additional Image Output Options**
  One of the inclusions into the new UI will be an image output location to be specified by the user. For users outside of USGS, this will always be for the images to be packaged for download (like the current process). Internally USGS employees will be given an option to specify a folder path on the server to place the finalized output. This will prevent any wasted bandwidth downloading the images just to be placed back on USGS servers.

The new image processing pipeline adds in the flexibility to generate dynamic workflows and provide a better experience for end POW website user. The focus on dynamic job creation will allow for easier future modifications to the pipeline and its ongoing maintenance. Next, we will detail the individual requirements that the project and each component entails.

4. Project Requirements

The project requirements were gathered through the client interviews onsite at USGS. They were developed through an iterative process where Pypeline was prototyping the basic functional and non-functional requirements of the project during the technological feasibility document. Afterward, we discussed the findings with the client during several meetings until we confirmed the specific requirements mentioned in the following sections.

4.1 Functional Requirements

In this section, we will be detailing the specific functions of our solution that will satisfy our client's requirements. The solutions will provide a user interface alongside a REST API. It will handle dynamic workflow generation and will also include a catered job scheduling solution. Last the solution will make use of Docker to wrap all the previously mentioned functions in two containerization environments.

1. **User Interface**
   The user interface will need to allow a user to create, edit, and submit jobs over the web and work with the existing PILOT website.
   - 1.1. Accept existing output from the PILOT website including user-selected images as a list of URLs along with an instrument unique tag that identifies the mission.
   - 1.2. The user interface will be required to parse existing mission recipes as well as any future mission recipes added by USGS that maintain the same format.
     - 1.2.1. The correct mission recipes will be retrieved based on the tag provided from the PILOT website.
     - 1.2.2. All ISIS commands provided in the recipe will be retrieved for presentation to the user.
     - 1.2.3. Parameters for the command from recipe will be used generate options presented to the user.
   - 1.3. Parsed recipe data will be presented to the user for modification prior to execution.
     - 1.3.1. A default workflow to be presented to the user as individual processing steps (based on all possible commands found with the mission recipe).

1.3.2. Individual processing steps can be deleted by the user from the workflow if marked as optional within the mission recipe.

1.3.3. The user will be able to make changes to the order in which the steps are executed

1.3.4. A user can choose to stop a workflow at any point and the rest of the workflow removed from execution.

1.3.5. The user will able to input parameters for each of the commands involved in the workflow.

1.4. The UI will make a call to the RESTful Interface based on user input at the completion of their modifications to the workflow.

1.4.1. The output will be a GET request to the formatted to the specification of the RESTful interface

1.4.2. The request will contain a JSON object that comprises the modified workflow from the user.

1.5. The UI Will allow for user selection of the location for the finalized images.

1.5.1. The packaging of the images for downloads will always be an option and the existing structure for generating the notification will be maintained.

1.5.2. USGS employees will be able to enter a file path on the internal USGS server.

2. RESTful Interface

A RESTful interface will need to accept formatted user data and generate a workflow script for use by the workflow management system.

2.1. The included user interface can access and utilize the RESTful interface through API calls.

2.2. The interface will be front end agnostic and function detached from the included UI.

2.2.1. The pipeline can be operated without the included UI through command line API calls.

2.3. Successful calls will initiate creation of user-defined processing jobs.

2.3.1. Will call the Dynamic Workflow Generation library to create a workflow.

2.3.2. The output will be stored and accessible in a shared folder to the job scheduler.

3. **Dynamic Workflow Generation**

A library containing functions for parsing mission recipes and converting supplied workflows into format compatible with the job scheduler.

3.1. Generator will be built as a library accessible to the RESTful interface and any additional front ends.

3.2. The generator will encapsulate all actions performed during job generation.

3.3. Library will include functions for parsing mission recipes and returning available ISIS commands.

3.4. Generate dynamic workflows based on the user's process specifications.

3.4.1. Able to generate workflow of specified mission and ISIS commands.

3.5. The library will include functions to interact with pipeline's scheduler.

3.5.1. Generate workflows capable of being parsed and executed by the scheduler.

4. **Pipeline Job Scheduling**

The workflow management software will need to constantly detect the creation of new user jobs, schedule them for execution on the processing cluster, and monitor their progress.

4.1. Detect new image processing jobs created by the job generator.

4.1.1. Monitor a specific folder mounted through the containerization platform for newly created workflows.

4.2. Scheduler will quickly parse and process newly created image processing jobs for execution on the USGS processing cluster.

4.3. The scheduler will send image processing jobs to the USGS processing cluster for execution.

4.4. Jobs will be able monitored by USGS personal via scheduler control panel.

4.4.1. Included scheduler UI will present a list of currently processing workflows to USGS staff.

4.4.2. Workflows can be selected to display the current task along the workflow will be illustrated.

5. **Containerization**

All of the provided components of the new pipeline will be containerized and function independently.

5.1. All functionality must be housed in containerization software for hosting.

5.2. Breaking the containerization must function through the included features within the containerization platform.

4.2 Non-functional Requirements

The Non-functional requirements of our project will be complementing the functional requirements so that the project will be up to the standards our client is expecting. The solution should be easily maintainable for their IT staff. Additionally, our solution should handle updates to the ISIS3 toolkit and new NASA missions. The solution should also be able to integrate with current existing architecture at USGS. Last, the solution we are developing should have similar execution time when compared to the current solution.

- **Maintainability**
  Our solution must be easily maintained by employees of the USGS following the completion of this project. Maintainability will be implemented by using a combination of coding style and documentation. Our solution will be implemented with the Black code formatter. Documentation will consist of a combination of general code documentation and diagrams that describe aspects of our solution, such as type assumptions, expected behaviors, and component organization. In order to ensure future contributions to our project are well-developed we will develop unit and integration tests. These tests will check the integrity of our project and its individual components after future changes are made.

- **Extensibility**
  The solution should be able to handle new updates to the ISIS toolkit. Updates to the toolkit will be able to added into the solution by editing or creating new Airflow task templates used by the workflow generator. The pipeline will accommodate new mission instruments through the addition of a mission recipes. The workflow generation library will be able to parse the new mission recipes without the staff working on the pipeline having to modify the UI or workflow generation library.

- **Integrates Within Existing Architecture**
  Our solution will integrate with the Docker swarm currently hosted and managed by USGS. The Docker containerization will isolate the new pipeline from other containers running on the server. As for the USGS cluster, the solution should easily send the jobs to processing cluster and schedule them through the use of Airflow. When the solution is deployed it will not affect the cluster’s other processes.

- **Performance**
  The new pipeline will have overall similar execution time when compared to the current pipeline solution. Airflow will be utilized to manage and monitor the execution of processing jobs.

4.3 Environmental Requirements

This project needs to fit in within the existing USGS system architecture. Additionally, the software solutions utilized need to be compatible with the future project plans and needs of USGS. Two main requirements presented to us by USGS were the containerization solution and the desired workflow management software.

- **Server Operating Environment**
  Our new pipeline is required to operate in a server environment that will be hosted by USGS. The containerization of the pipeline software itself will allow for its execution on a wide variety of host operating systems supported by the containerization platform. USGS other software projects are all Unix based and supported multiple Linux distribution and OSX. To continue with this standard, the pipeline software will run utilizing a minimal linux kernel within the containers themselves.

- **Docker**
  The new pipeline was required to be containerized to allow for isolation of the project from the rest of the USGS system. USGS already has in place a Docker swarm which is utilized for other USGS projects and service hosting. The new pipeline is required by USGS to be contained within one or more Docker containers. This will allow for the new project to be integrated within the swarm without any additional software or maintenance needed. Utilization of the shared volumes feature found within Docker was approved as a solution by USGS for breaking the containerization. This feature is built into Docker itself and requires no outside software for implementation.

- **Airflow**
  The replacement pipeline will need to use Airflow for workflow management, scheduling, and, execution. Airflow’s use of Python scripts as the basis for workflows will provide the ability for them to be dynamically generated as required by the project. The included Airflow UI will provide for visualization of currently running workflows and prebuilt views for metrics on previously run image processing jobs. The included scheduler will be used to execute the

workflows on the processing cluster. The scheduler will fit into plans at USGS to utilize it for execution of not only the image processing jobs but additional future projects.

Ensuring compatibility and simple integration into existing USGS architecture is a high priority for the project. This focus will help minimize the risks to the project that will we will face during its deployment at USGS.

5. Potential Risks

The biggest risks to the project will occur over long-term use of the software and the new pipeline needs to be designed their mitigation in mind. Specifically, the inclusion of new NASA missions and future alterations to the ISIS toolkit need to be taken into consideration when developing the new pipeline. Failure to do so could result in the misuse of mission-specific ISIS programs or incorrect execution order of tasks within a workflow. Additionally, we will need to account for changes to the USGS architecture or alteration of the POW website.

- **Future Updates to ISIS**
  The ISIS toolkit is updated about four times a year to support new missions and provide additional functionality within the toolkit. The workflows generated by the new pipeline will be directly utilizing ISIS via the command line and any changes to the syntax of those commands will prevent the successful completion processing jobs. Due to the certainty of future updates, the new pipeline will need to allow for modifications to be made in tandem with ISIS releases. The ISIS commands themselves will be mirrored as task templates within the workflow generation library. This isolation within the code for the pipeline will require only minimal changes to support a new version of the toolkit.

- **Changes to the USGS System Architecture**
  The new pipeline must function within the USGS system architecture and alterations the environment have the potential to prevent its operation. Due to the nature of USGS's work, system changes are infrequent but can significantly alter the hosting environment. To mitigate the impact of environment changes to the pipeline, it will be designed to make use of an isolated running environment. Containerization will allow for the pipeline executables and their required dependencies to be stored together and remain independent of changes to the hosting environment.

- **Alterations to the POW website**
  One of the potential risks we considered when developing our solution is alterations of the POW website. Based on discussions with USGS, the PILOT and POW websites are likely candidates for future development. Following a change of the website, the user interface may produce incompatible output to the new pipeline. The use of a RESTful interface will allow for changes or replacement of the front end of POW without affecting the workflow generation abilities of pipeline itself. The only requirement of the updated UI is that it functions though documented methods implemented by the interface.

6. Project Plan

We have divided our tentative plan for project execution into two parts, Fall and Spring semesters. These two semesters logically separate the planning and development phases of the project. We have designed our project plan to reflect the different milestones that will need be accomplished during each phase.

There two major project milestones in the Fall semester are the finalization of our project requirements and the implementation of our prototype. Figure 4 depicts the overall structure of our process during the fall semester. Finalization and sign off of the project requirements allows our team to be confident in our project vision. The completion of our prototype at the end of the Fall semester will mark our shift of focus from planning to implementation. The prototype will demonstrate the key technical components of the project and how they enable us to translate our project requirements into a final product. Our prototype will be an end to end implementation of the pipeline designed to function with a single mission camera. The image processing workflow generated by the prototype will utilize a subset of ISIS commands and demonstrate the creation of a minimal but complete workflow.

An important part of this project is ensuring that the product we deliver will be usable by the USGS. To ensure a smooth transition for our project, we have allocated ample time in the Spring semester for testing and integrating our project at USGS. However, in order to take advantage of that allocated time, we must closely follow our set milestones for the Spring as shown in Figure 5.

The first Spring milestones are the creation of our project's RESTful interface and the dynamic workflow generation library. The finalizing of the interface is important due to the reliance of all other project components on its operation and design. Dynamic workflow generation is a vital part of our project and must be completed before any meaningful testing beyond the prototype can take place. Due to the inter-dependence of these two components, their development will need to be executed in tandem. We plan to accomplish this by dividing the team into pairs that will focus on each component. Once the interface is completed, that development pair will transition to the implementation of the user interface, our third milestone. Afterwards, the full team will turn to internal testing of our completed pipeline to ensure it is properly implemented and satisfies all project requirements. The completion of this testing is the fourth milestone of our project. Finally, the integration of our project into USGS's systems will be the final milestone and represent the completion of the project.

Figure 4. Fall Semester Gantt Chart

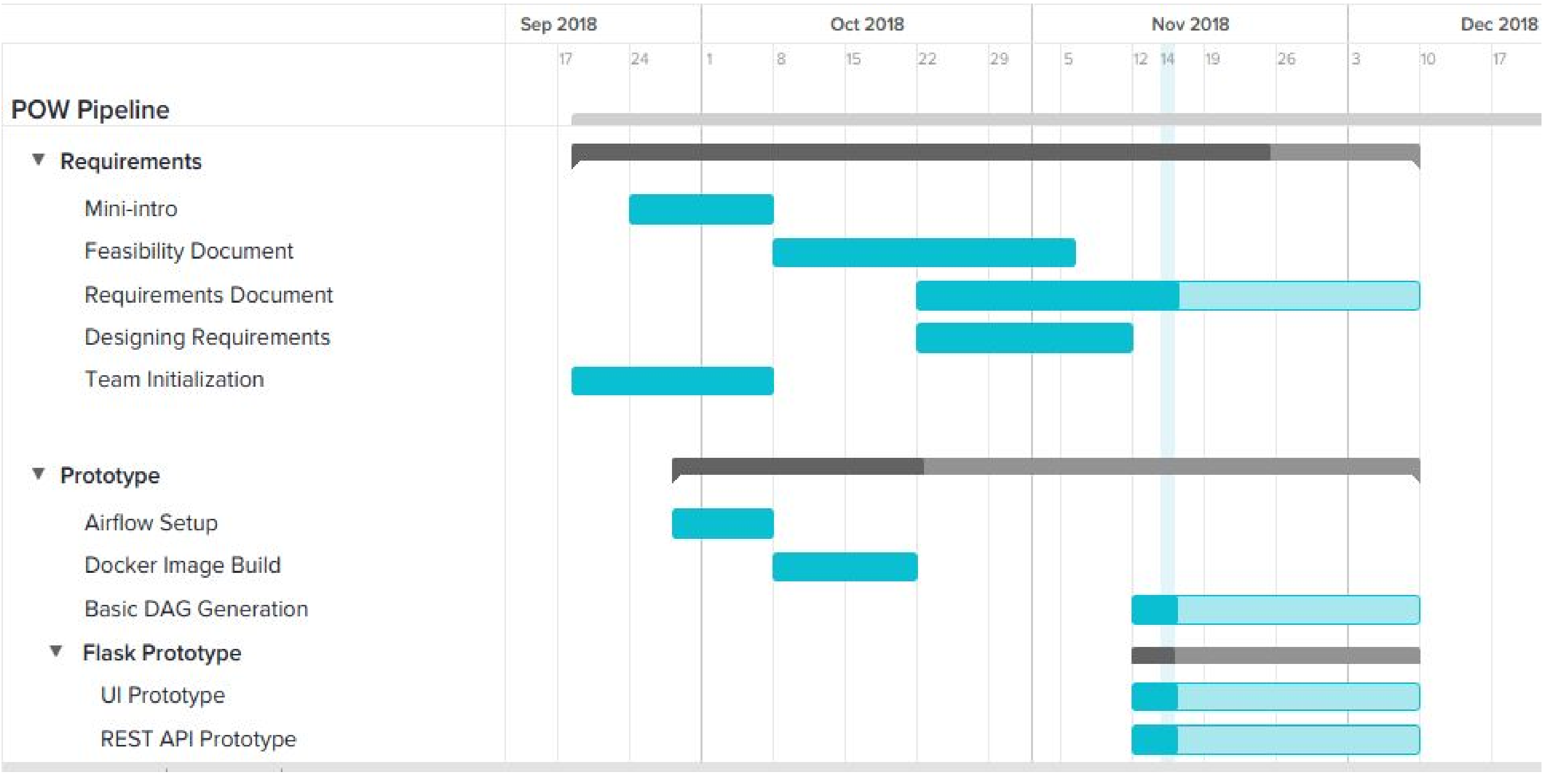
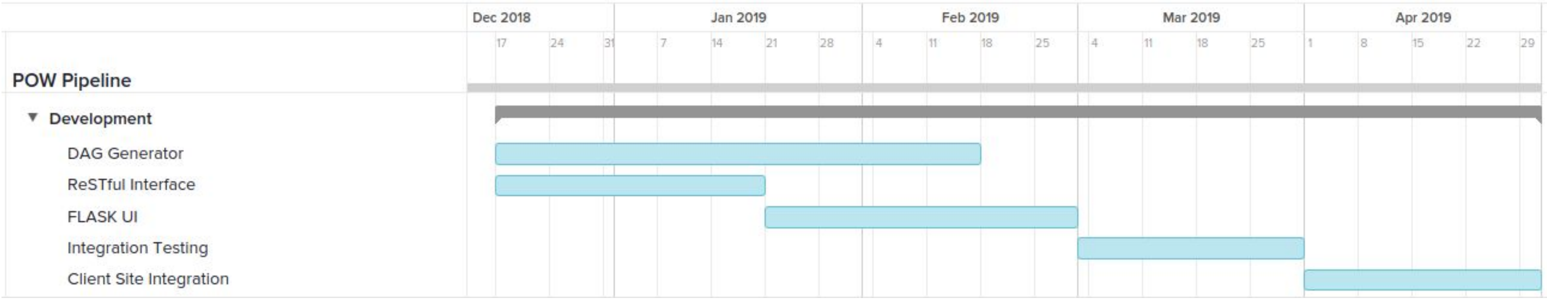


Figure 5. Spring Semester Gantt Chart



7. Conclusion

In addition to its scientific research, USGS serves an important role for the scientific community and the public by providing access to all of NASA's planetary image data. Distribution of the archived images is handled through the PILOT website maintained by USGS. The accompanying POW website provides accessibility to the ISIS toolkit itself by allowing their execution on the USGS processing cluster. However, the current POW processing pipeline behind PILOT fails to meet the modern standards of the USGS.

The current pipeline lacks the flexibility needed to generate some of the workflows requested by its users and requires too many modifications in order to be kept updated with the ISIS toolkit. Our project will replace the workflow generation that lies between the PILOT website and the ISIS toolkit as outlined in Figure 3. The new pipeline will provide a greater degree of customization by dynamically generating workflows that are unique to each processing job. It will provide easier maintainability by requiring only minimal changes to the system to order in support new missions and ISIS toolkit updates.

The purpose of this document is to define the requirements of our project and the risks associated with them. Additionally, it describes the finalized structure and flow of our project. Now that we have done this, we will be able to design and implement our project to meet clear operating objectives.

This document represents the culmination of our work in the Fall semester. With it, our group can continue into the Spring semester confident in our ability to deliver a well-developed, high quality product. Our solution will provide a flexible user experience alongside a modern backend, supported by dynamic workflow generation and scheduling. We look forward to continuing to work with our client and to serve the mission of the USGS.