AirFlow Processing Pipeline

Design Review

December 2nd, 2022

Who We Are

The Team

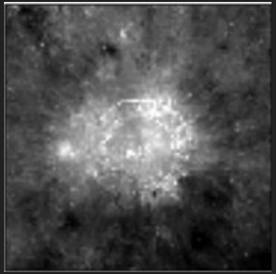
- Chris McCorkle
- Isaiah Raspet
- Richard McCormick
- Quinton Jasper
- Hunter Woodruff

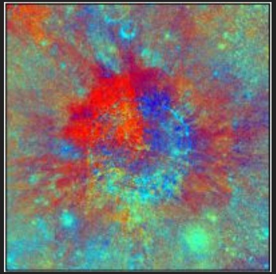
Our Mentor

Vahid Nikoonejad Fard

Our Sponsor

Trent Hare
USGS Astrogeology
Cartographer

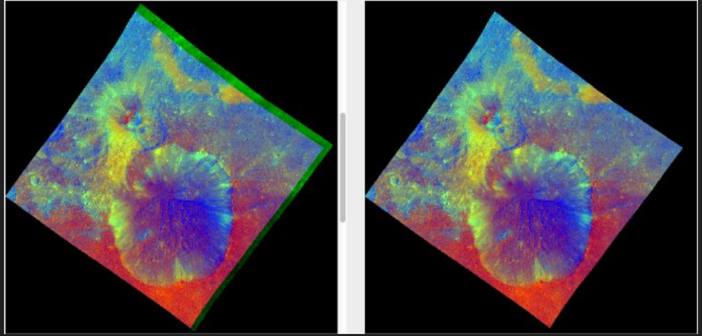


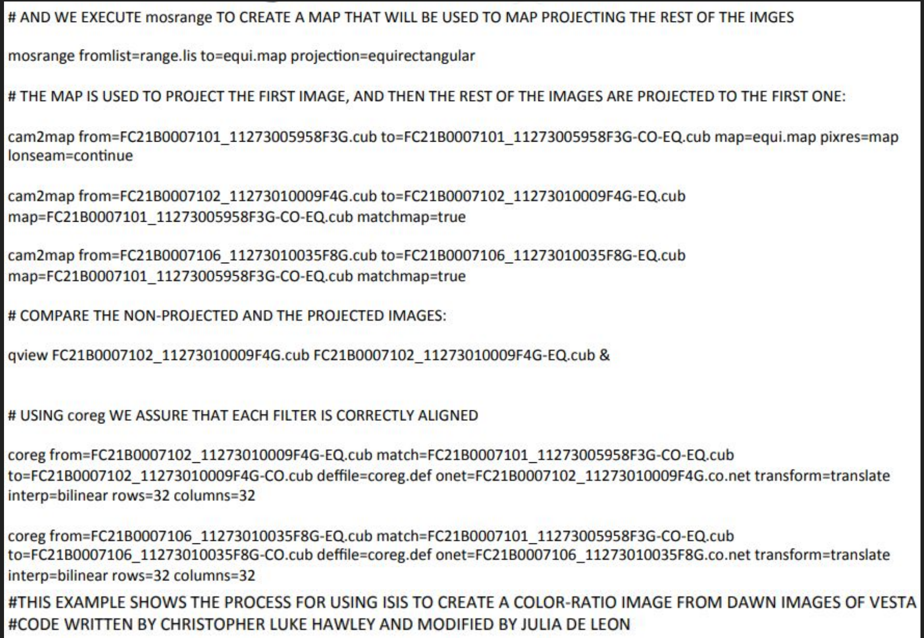


Our Client‘s Problem…

Main Problems

- ISIS3 has over 300 different applications
- Currently using terminal to run bash shells to compile images
- Researchers cannot save pipelines they want to use efficiently





```
# AND WE EXECUTE mosrange TO CREATE A MAP THAT WILL BE USED TO MAP PROJECTING THE REST OF THE IMGES

mosrange fromlist=range.lis to=equi.map projection=equirectangular

# THE MAP IS USED TO PROJECT THE FIRST IMAGE, AND THEN THE REST OF THE IMAGES ARE PROJECTED TO THE FIRST ONE:

cam2map from=FC21B0007101_11273005958F3G.cub to=FC21B0007101_11273005958F3G-CO-EQ.cub map=equi.map pixres=map
lonseam=continue

cam2map from=FC21B0007102_11273010009F4G.cub to=FC21B0007102_11273010009F4G-EQ.cub
map=FC21B0007101_11273005958F3G-CO-EQ.cub matchmap=true

cam2map from=FC21B0007106_11273010035F8G.cub to=FC21B0007106_11273010035F8G-EQ.cub
map=FC21B0007101_11273005958F3G-CO-EQ.cub matchmap=true

# COMPARE THE NON-PROJECTED AND THE PROJECTED IMAGES:

qview FC21B0007102_11273010009F4G.cub FC21B0007102_11273010009F4G-EQ.cub &


# USING coreg WE ASSURE THAT EACH FILTER IS CORRECTLY ALIGNED

coreg from=FC21B0007102_11273010009F4G-EQ.cub match=FC21B0007101_11273005958F3G-CO-EQ.cub
to=FC21B0007102_11273010009F4G-CO.cub deffile=coreg.def onet=FC21B0007102_11273010009F4G.co.net transform=translate
interp=bilinear rows=32 columns=32

coreg from=FC21B0007106_11273010035F8G-EQ.cub match=FC21B0007101_11273005958F3G-CO-EQ.cub
to=FC21B0007106_11273010035F8G-CO.cub deffile=coreg.def onet=FC21B0007106_11273010035F8G.co.net transform=translate
interp=bilinear rows=32 columns=32
#THIS EXAMPLE SHOWS THE PROCESS FOR USING ISIS TO CREATE A COLOR-RATIO IMAGE FROM DAWN IMAGES OF VESTA
#CODE WRITTEN BY CHRISTOPHER LUKE HAWLEY AND MODIFIED BY JULIA DE LEON
```

Our Client‘s Problem…

Minimum Viable Product

- Access to prebuilt commonly used pipelines or “Recipes”
- Ability to create and run unique workflow pipelines efficiently
- A user friendly interface
- A container to be able to efficiently deploy the project on any number of systems.

Stretch Goals:

- Minimal or non coding creation of pipeline
- Export support for cloud optimized metadata files
- Export support for main output directly to web services hosted by the U.S. Geological Survey

Possible Workflow Softwares

Apache Airflow

Luigi

Jenkins

Possible Environments

Conda

Docker

Our Proposed Solution

- Through the use of the ‘Airflow processing pipeline software’, we propose a system of python-wrapped ISIS3 ‘Nodes’ which will allow the scientists at USGS to accurately, efficiently, and quickly create actionable products.
- Python-based module wrapperers would allow for the creation of pipelines that can be automated using the AirFlow system.
- Pre-built pipelines can also be built to allow for frequently used configurations to be easily accessed.

Feasibility & Risks

- All things considered, our team believes that we will be able to implement our proposed solution within the allotted time frame.

- Several stretch goals outlined by the client are also obtainable within the time frame, such as exporting final products directly to a live web map.

- The risks for this project are relatively low. It is possible for malformed data or nodes to produce a final image which is not correct, but this will not result in significant or permanent damage to the system or data.

Key Requirements

How we got here:

- Collaboration and communication with our client
  - Multiple meetings with client
  - Discord and email communication
- Understanding the problem and reading provided documents
  - Going over the requirements set out by client
  - Doing personal research into the technologies
- Team brainstorming

Key requirements

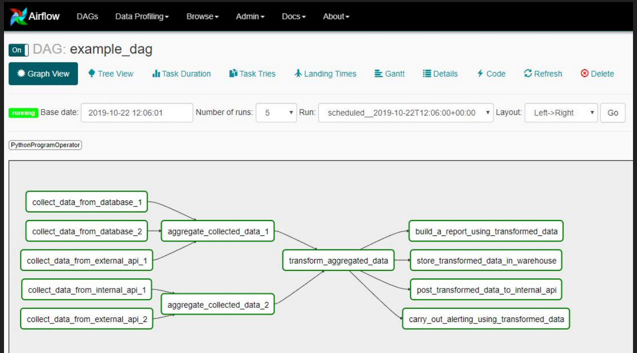
Our goals:

- User Visibility
  - Track and watch progress
- User interactivity
  - Environment that is adaptable and flexible
- Ability to save and create pipelines seamlessly
  - Ability to effortlessly create and save work
- Single instance environments
  - Run inside of a single instance
- Customizable DAGs
  - Directed acyclic graphs that are fully customizable

Key Requirements

In depth:

- Saving and Creating Pipelines
    - Functional: Apache Airflow, Elyra
        - In Airflow, users will be able to take individual ISIS3 commands, which our team will be wrapping in python to be accessible to Airflow, and create directed acyclic graphs.
        - These DAGs, or pipelines, will be able to take a base image input, then run through the pipeline’s commands in the order that the user chose, to generate an actionable image product.

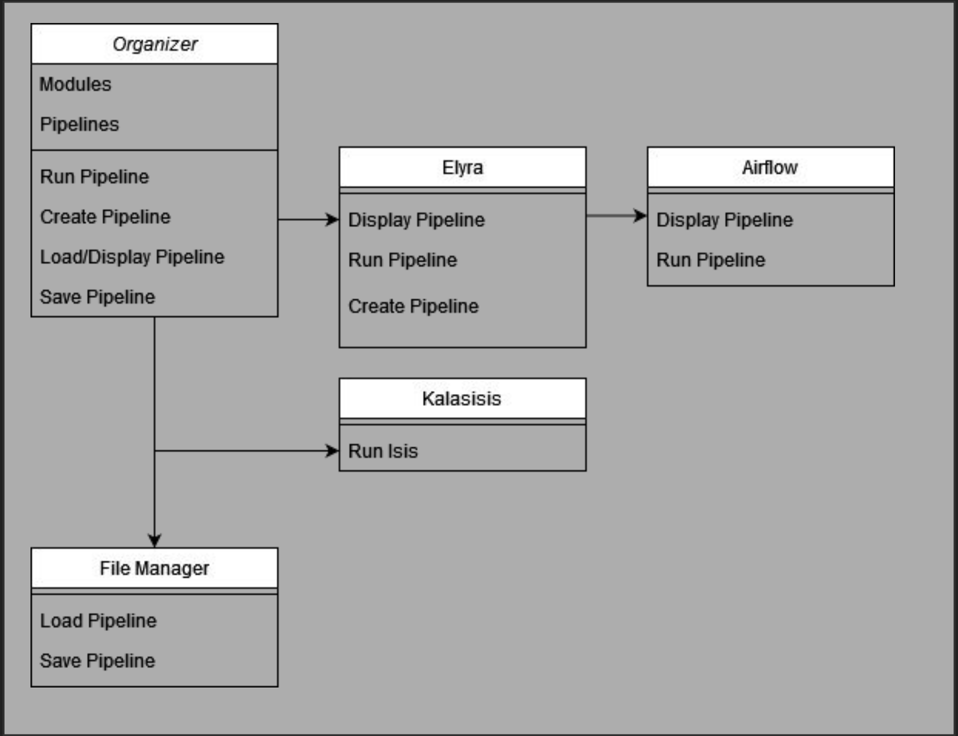

Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
On
Graph View
Tree View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Refresh
Delete
running
Go
PythonProgramOperator
collect_data_from_database_1
collect_data_from_database_2
collect_data_from_external_api_1
collect_data_from_internal_api_1
collect_data_from_external_api_2
aggregate_collected_data_1
aggregate_collected_data_2
transform_aggregated_data
build_a_report_using_transformed_data
store_transformed_data_in_warehouse
post_transformed_data_to_internal_api
carry_out_alerting_using_transformed_data

DAG: example_dag

Base date: 2019-10-22 12:06:01 Number of runs: 5 Run: scheduled__2019-10-22T12:06:00+00:00 Layout: Left->Right

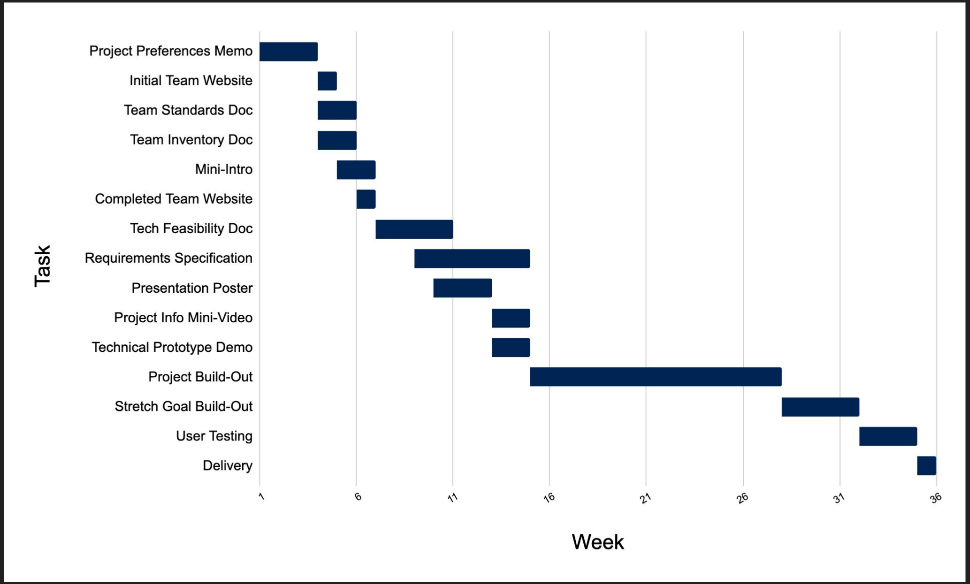
Architecture

- We are focused on keeping our architecture simple and straightforward
  - This project involves connecting many complicated modules in a specific manner to create a system that is easy to use and hopefully be extensible in the future



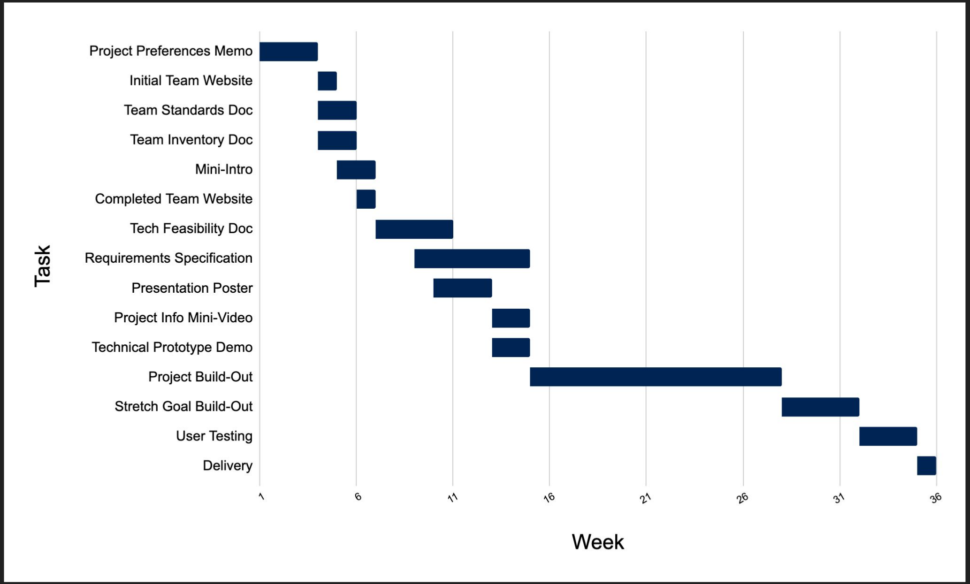
Development Schedule

- Previous Developments
  - Client Collaboration
  - Technology/Tool Analysis
  - Outcome Expectations



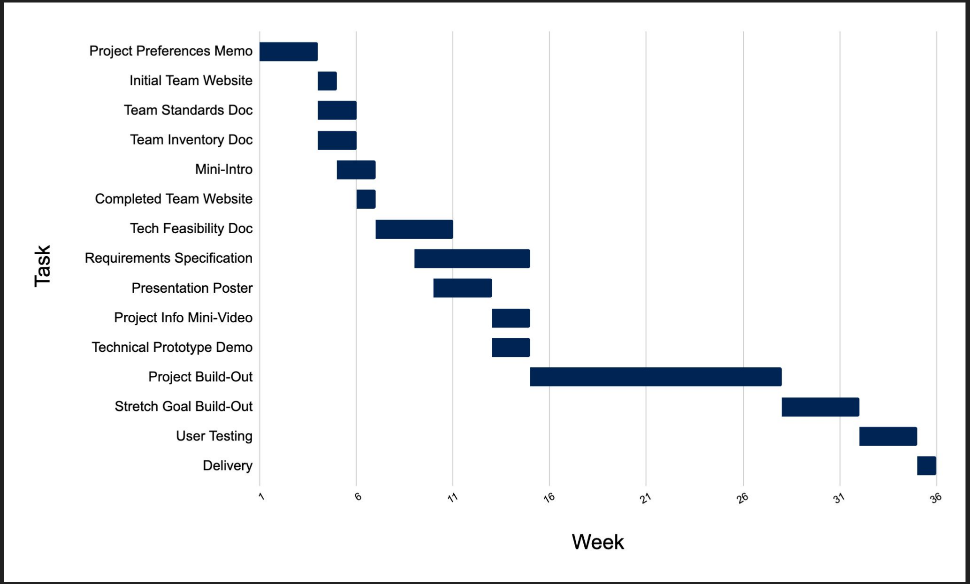
Development Schedule

- Previous Developments
  - Client Collaboration
  - Technology/Tool Analysis
  - Outcome Expectations
- Current Progress
  - Fast Approaching Week 16



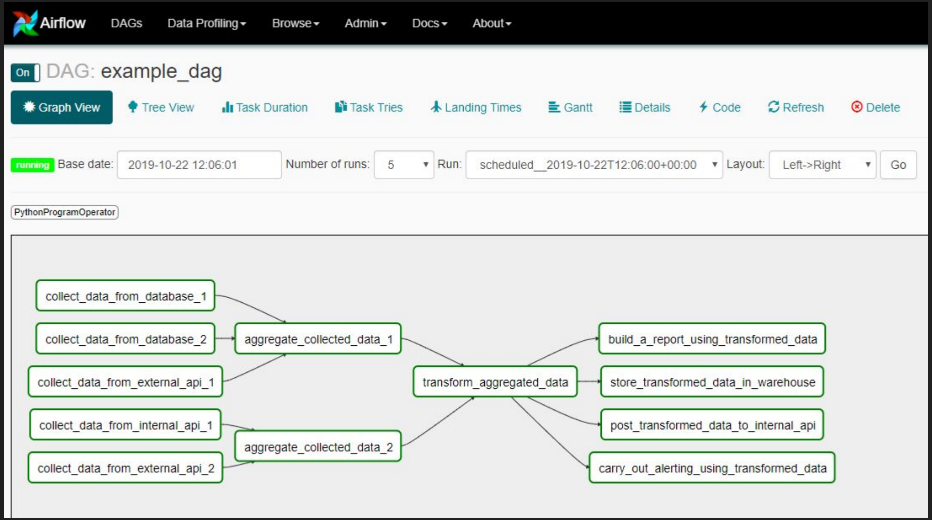
Development Schedule

- Previous Developments
  - Client Collaboration
  - Technology/Tool Analysis
  - Outcome Expectations
- Current Progress
  - Fast Approaching Week 16
- The Future
  - Spring 2023
  - Details to come



In Review

- USGS Astrogeology
  - Image and Data processing
  - Generate Accurate Planetary Maps
- A Pipeline Management System
  - Accessibility
  - Portability
  - Reusability
- Our solution
  - Develop a Base
  - Combine Technologies



DAG: example_dag

Base date: 2019-10-22 12:06:01 Number of runs: 5 Run: scheduled__2019-10-22T12:06:00+00:00 Layout: Left->Right

collect_data_from_database_1
collect_data_from_database_2
collect_data_from_external_api_1
collect_data_from_internal_api_1
collect_data_from_external_api_2
aggregate_collected_data_1
aggregate_collected_data_2
transform_aggregated_data
build_a_report_using_transformed_data
store_transformed_data_in_warehouse
post_transformed_data_to_internal_api
carry_out_alerting_using_transformed_data

Thank You!

We will now accept any questions