**DEVELOP National Program Python Package (dnppy)**

**Software Description & NASA Software Engineering Classification**

**Introduction**

Teams in the DEVELOP program increasingly find themselves using some level of programming to manipulate data. Much of the time, this manipulation is performed in Python. The DEVLEOP Python package, referred to as “dnppy” is being created to improve institutional knowledge retention, open the DEVELOP toolkit for public contributions and use, and represent DEVLEOP in the public domain. It is a joint social media and programming capacity building endeavor.

**Applications and Scope**

The dnppy package will be used to functionalize common programming tasks in the geospatial community, specifically for working with NASA data products. It will include functions for processing satellite data and assist in structuring analysis to reduce the startup time for DEVELOP teams to learn programming and create tools for end users.

**Capabilities**

This software can streamline common processing tasks of NASA data products.

**Interfaces**

The package is a python module, written entirely in python.

**Assumptions, Limitations, & Errors**

* Many functions require the arcpy module, which comes with Arcmap
* There is presently no functionality built into the module that subjects the users to functionally inherent errors.

**Additional Information**

**Software Classification & Justification**

This software is considered to be Class E software per NPR 7150.2.

This software is used to:

* + Perform minor desktop analysis of science or experimental data.

The software is not used to:

* + make decisions for an operational Class A, B, or C system or to-be built Class A, B, or C system
  + support engineering development
  + test other Class D software systems
  + support mission planning or formulation
  + operate a research, development, test, or evaluation laboratory
  + provide decision support for non-mission critical situations
  + in a Major Engineering/Research Facility
  + perform research associated with airborne vehicles or systems

The software will not:

* + operate, directly support, or be flight qualified to support an operational system
  + be used in technical decision concerning an operational system
  + directly affect primary or secondary mission objectives
  + adversely affect the integrity of engineering/scientific artifacts
  + have an impact on operational vehicles

Additionally, if the software had anomalous behavior, that behavior would not cause or contribute to a failure of a system function:

* + resulting in a minor failure condition for the airborne vehicle
  + with an effect on airborne vehicle operational capability or pilot workload

When these criteria are no longer valid, categorization/classification will be reevaluated and the project will start following the procedures for the higher class.

**Not Safety Critical**: The software does not:

* Reside in a safety-critical system with at least one of the following being applicable to the software:
  + Causes or contributes to a hazard
  + Provide control or mitigation for hazards
  + Controls safety-critical functions
  + Processes safety-critical commands or data
  + Detects and reports, or takes corrective action, if the system reaches a specific hazardous state
  + Mitigates damage if a hazard occurs
  + Resides on the same system (processor) as safety-critical software
  + Process data or analyze trends that lead directly to safety decisions (e.g. determining when to turn power off to a wind tunnel to prevent system destruction)
  + Provide full or partial verification or validation of safety-critical systems, including hardware or software systems