Geospatial Data Abstraction Library

Software Assurance Classification Report

Prepared by:

Name	Date
Leslie J. Johnson, NASA LaRC Software Assurance Engineer Mission Assurance Branch	12/16/2016

NASA Langley Research Center

Hampton, VA 23681

Revision and History Page

Revision No.	Description	Release Date
-	Initial Release	12/16/2016

Table of Contents

1. INTRODUCTION

This report contains the software assurance classification assessment which identifies and evaluates the characteristics of software in determining the software's classification, software safety-criticality, and level of software assurance to be applied to a Project.

1.1 Background

The Geospatial Data Abstraction Library (GDAL) is an open source, translator library used by geographic information system (GIS) software to access Earth Observing Satellite (EOS) data products, which are in raster and vector geospatial data formats. As a library, GDAL presents a single raster abstract data model and vector abstract data model to the calling application for all supported formats. Many geospatial tools - including ArcGIS, GeoServer, MapServer, and Quantum GIS (QGIS) - rely on GDAL. Nonetheless, EOS data products have been difficult to consume by GIS tools, whether commercial or open-source, as GIS applications are frequently unable to read files or unable to properly interpret the internal data structures necessary to be visualized or analyzed. As a result of data products being difficult to consume by GIS applications, there has been a respective reduction in the acceptance of these data products by GIS and general user communities, emphasizing the need for a solution that can be adopted by all NASA Distributed Active Archive Centers (DAACs). To address these challenges and make EOS data products more accessible and interpretable by GIS applications, a collaborative approach was taken by the NASA Atmospheric Science Data Center (ASDC) at the Langley Research Center (LaRC), George Mason University (GMU), and The HDF Group through a project titled "GDAL Enhancements for ESDIS (Earth Science Data and Information System)" (GEE) to identify specific problems as well as causes of those problems then develop a framework and plug-ins to offer solutions.

This software will not be used in a safety-critical system, to monitor a safety-critical system, to verify or validate a safety critical system, or to make safety decisions.

2. REFERENCE DOCUMENTS

The following documents were used or referenced in the development of this report:

Document No.	Document Title
NPR 7150.2B	NASA Software Engineering Requirements
NASA-STD-8739.8	NASA Software Assurance Standard
NASA-STD-8719.13	NASA Software Safety Standard
LAPD 5300.1	Program/Product Assurance
LPR 7150.2	LaRC Software Engineering Requirements
LPR 5300.1	Product Assurance Plan
LMS-CP-4754	Software Assurance (SA) for Development and Acquisition

3. SUMMARY

The following paragraphs summarize the results and describe the details used to determine the software classification assessment for this report.

3.1 Software Classification

According to LPR 7150.2, this software is classified as Class E – Design Concept and Research and Technology Software which is defined as

1. *Software developed to explore a design concept or hypothesis, but not used to make decisions for an operational Class A, B, or C system or to-be built Class A, B, or C system, or*
2. *Software used to perform minor desktop analysis of science or experimental data.*

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

3.2 Software Safety

The Software Safety Litmus Test below is applied to all projects with software to determine if the software is safety-critical. If the software is determined to be safety-critical, then the project must adhere to the NASA-STD-8719.13, NASA Software Safety Standard.

A software component is considered safety-critical if it meets **any** of the following criteria:

Criteria:	**Software components**
a. Resides in a safety-critical system (as determined by a hazard analysis) **AND** at least one of the following apply:	No
(1) Causes or contributes to a hazard	
(2) Provides control or mitigation for hazards	
(3) Controls safety-critical functions	
(4) Processes safety-critical commands or data	
(5) Detects and reports, or takes corrective action, if the system reaches a specific hazardous state	
(6) Mitigates damage if a hazard occurs	
(7) Resides on the same system (processor) as safety-critical software	
b. Processes data or analyzes trends that lead directly to safety decisions	No
c. Provides full or partial verification or validation of safety-critical systems, including hardware or software subsystems.	No

The software components in this Project do not reside in a safety-critical system; process data or analyze trends that lead directly to safety decisions or provide full or partial verification or validation of safety-critical systems.

3.3 Software Assurance Effort

The software assurance effort is based on the software class and impacts from potential failure. In accordance with LMS-CP-4754 software assurance is not applicable for non-safety critical Class E software developments.

The Project shall follow the instructions and complete the compliance matrix in LMS-CP-7150.6, *Class E Software*, which applies to all Class E software that is not safety-critical.

APPENDIX A: ACRONYMS

CP	Center Process
LaRC	Langley Research Center
LAPD	Langley Policy and Directives
LMS	Langley Management System
LPR	Langley Procedural Requirements
NASA	National Aeronautics and Space Administration
NPR	NASA Procedural Requirement
SA	Software Assurance
STD	Standard