Unstructured CFD (Computational Fluid Dynamics) Graph Operations Mini-application

Software Assurance Classification Report

Prepared by:

Name	Date
Leslie J. Johnson, NASA LaRC Software Assurance Engineer Mission Assurance Branch	LESLIE JOHNSON Digitally signed by LESLIE JOHNSON Date: 2021.08.10 13:22:47 -04'00'

NASA Langley Research Center
Hampton, VA 23681

Revision and History Page

Revision No.	Description	Release Date
-	Initial Release	8/10/2021

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.

2 REFERENCE DOCUMENTS

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

Document No.	Document Title
NPR 7150.2**C**	NASA Software Engineering Requirements
NASA-STD-8739.8**A**	NASA Software Assurance and Software Safety Standard
LAPD 5300.1	Product Assurance Program
LPR 5300.1	Product Assurance Requirements
LMS-CP-4754	Software Assurance (SA) for Development and Acquisition

3 BACKGROUND

The application is two tasks needed of a fluid dynamics simulator for the purposes of measuring performance of those tasks on different high performance computing (HPC) hardware. The application can formulate two sides of the conservation equations of mass, momentum, and energy. The application has no mechanism to solve the equations.

The purpose of the software is very narrow. It is to understand how graph traversal and reductions can be coded to run faster. These tasks are commonly used in "real" Computational Fluid Dynamics simulation codes. This application will help find strategies on how to make these algorithms efficient on newer HPC hardware. The software does not solve equations, nor is it capable of making any scientific or engineering predictions. The output of the software is timing data and other metrics that pertain to performance.

Inputs: A file containing the graph of where communication needs to happen in the system, and a text file containing information about what the number of equations to be constructed.

Outputs: The output is performance information: how long each function took to complete, and how much memory was moved. This information is printed to the screen.

The miniapp can run on laptops, desktop workstations, and high performance compute nodes, particularly those with GPGPUs (General Programming Graphics Processing Units, for example nVidia V100).

The software does not interface with any other hardware except the computing machine it runs on.

The outcome of this software is to learn what programming techniques should be used to write high performance software.

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.

4 SUMMARY

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

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

4.1 Software Classification According to NPR 7150.2C, this software is classified as Class E – Design Concept, Research, Technology, and General Purpose 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.

2. Software used to perform minor analyses of science or experimental data. Class E software cannot be safety-critical software. If the software is classified as safety-critical software, then it has to be classified as Class D or higher.

3. A defect in Class E software may affect the productivity of a single user or small group of users but generally will not affect mission objectives or system safety.

4. Class E software runs in a general-purpose computing environment or a board top environment. Class E software does not support ground tests, flight tests, or operations.

4.2 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.

4.3 Results

The Project shall follow the instructions/requirements in NPR 7150.2 and complete the Requirements Mapping Matrix, which applies to Class E software.

The Requirements Mapping Matrix with associated documentation shall be reviewed, approved, and signed by the Project Management/Branch Head and the Software Assurance Engineer in the Mission Assurance Branch.

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