Appendix A1: Software Management Plan (SMP)

1. Project Organization.

* *Cesar Munoz and Mario Consiglio from the Safety-Critical Avionics Systems branch will be leading the project. Cesar Munoz will serve as the technical lead. Swee Balachandran from the National Institute of Aerospace will be working on research, software development, and integration.*
* *Leslie Johnson of the Mission Assurance Branch performs the Software Assurance Classification Assessment and safety-criticality determination as defined in LMSCP-4754, Software Assurance (SA) for Development and Acquisition.*

1. Software Classification and Safety Criticality.

*The software system is Class D and not safety critical.*

1. Compliance Matrix.

*The completed compliance matrix is attached.*

1. Engineering environment.

* *Currently, Java and C++ implementations of ICAROUS are available. Java SDK 8 SE or greater will be required for compilation and execution of the Java implementation..*
* *Communication between ICAROUS and any external device is established via the MAVLink protocol. MAVLink is a communication protocol widely used for UAS applications (*[*http://www.qgroundcontrol.org/mavlink/start*](http://www.qgroundcontrol.org/mavlink/start)*).*
* *The JSSC library is required to enable communication over serial ports for the Java implementation (*[*https://code.google.com/archive/p/java-simple-serial-connector/*](https://code.google.com/archive/p/java-simple-serial-connector/)*).*
* *The software in the loop simulation application (*[*http://ardupilot.org/dev/docs/using-sitl-for-ardupilot-testing.html*](http://ardupilot.org/dev/docs/using-sitl-for-ardupilot-testing.html)*) is required to test the developed code.*
* *For flight tests, the Pixhawk flight controller (*[*http://www.pixhawk.org/modules/pixhawk*](http://www.pixhawk.org/modules/pixhawk)*) is required.*
* *For running software in the loop simulations, a desktop PC running Ubuntu 16.04 or greater is required.*
* *Communication with ICAROUS can be established via any MAVLink compatible ground station application. The MAVProxy ground station (*[*https://github.com/ArduPilot/MAVProxy*](https://github.com/ArduPilot/MAVProxy)*) is required for running tests documented in the Software Test Plan document.*
* *For flight tests, an embedded computer with an ARM processor running linux is required. A serial port on the embedded computer must be connected to the TELEM2 port on the Pixhawk controller (*[*http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html*](http://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html)*).*

1. WBS, Schedule, Effort, and Cost. [SWE-102.e] [SWE-016] [SWE-015]
   1. Document a list (e.g., Work Breakdown Structure (WBS)) of: [SWE-102.e]
      1. Software activities:

* Develop core algorithms to be used in ICAROUS:
  + Detect and avoid algorithms (DAIDALUS)
  + Geofence related algorithms (POLYCARP)
* Formally verify core algorithms
* Develop decision making algorithms that utilize core algorithms to enable complex missions.
* Develop communication modules.
* Develop test scripts.
* Run and evaluate simulation results
  + 1. Software products that will be produced:

*An API that will enable safe operation of UAS for a wide range of missions (including beyond visual line of sight missions).*

* + 1. Appropriate software documentation to be produced to support the operations and maintenance phases of the life cycle:

*Software documentation and use notes are provided in the Software Design Description document.*

* + 1. \*Software services and non-deliverable items to be performed.

NA

All the activities above are funded by SASO’s UTM Project under WBS: 154692.02.10.07.01.

* 1. Document the software schedule associated with the activities, services, and deliverables that satisfies the following minimum conditions:

*This project is the by-product of research activities.*

Document an estimate of effort (include both civil servant and contractor effort) that covers either: *This project is the by-product of research activities.*

* 1. \*Document an estimate of special costs that covers the entire software life cycle or period of performance: NA

1. Requirements Management. Describe how the project will collect and manage changes to the software requirements.

*Revisions to the software requirements specification document will be tracked by the file names. The version number will be indicated as part of the file name.*

1. Contractor management. Include the following: [SWE-102.h]
   1. The make-versus-buy decision (i.e., document the results of Step 1.b in Section 1 of this procedure).

*This software project is a by-product of research activities and hence software modules have to be designed and implemented from scratch.*

* 1. \*The acquisition planning decisions and the activities to be performed to gain insight on the supplier

NA

1. Software acceptance.
   1. Document the acceptance criteria that will be used to confirm that the software fulfills its intended use (and any necessary conditions under which the acceptance criteria must be met).

*The tests described in the Software Test Plan document will be executed. On successful completion of these tests (more details on the expected outcome of each tests are available in the Software Test Plan document), it will be deemed that the software fulfills its intended use.*

* 1. Describe how the project will ensure software products meet acceptance criteria and any conditions

*The developer will execute all the test cases after changes to the software and verify it meets the acceptance criteria.*

* 1. Describe how the project will record results of software acceptance, and address, and track issues to closure

*During testing, if expected results are not obtained, the developers will be notified either via the revision control management system or via email/phone. The developers will then fix the relevant issues and close the relevant tickets or inform concerned authorities regarding the bug fixes.*

* 1. \* If applicable, describe how the project will perform any additional activities (other than testing) such as demonstration, analysis, or inspection to confirm the product fulfills its intended use and requirements; and how the project will record results of those activities, and address, and track issues to closure.

*Flight tests will be conducted to evaluate real world application of the developed software. Any issues that arise during flight tests will be conveyed to the developers. The process described in h(3) will apply to this case as well.*

1. Reviews. [SWE-018]
   1. Define the reviews that the project will hold regularly of software activities, status, and results with project stakeholders and the frequency or schedule for the reviews.

*Self-review of code through testing and inspection will be conducted throughout the development process. Flight test demonstration will be conducted at regular intervals to obtain and assess real world performance.*

* 1. Describe how the project will track issues from reviews to resolution.

*All testing and inspection issues/defects will be tracked to closure by the developer as encountered.*

1. \*Approvals. If applicable, any additional approval required by such means as regulations, required certifications, proprietary, usage, ownership, warranty, and licensing rights.

NA

1. Plan Tracking. Describe how the project will ensure that actual milestones and deliverables are tracked against the SMP.

*The developers, technical lead and project managers will hold regular meetings, perform flight tests and discuss results to track milestones and deliverables against the SMP.*

1. Software development life-cycle phases.

*Requirements phase, design phase, coding phase, integration phase, test phase, acceptance phase, and maintenance phase.*

1. Configuration Management.

*See Appendix A2*

1. Maintenance:

The project will follow Section 1: Perform Software Activities of LMS-CP-7150.5 and the Software Configuration Management Plan to perform maintenance.

1. \*Metrics.

NA

1. Off-the-shelf (OTS) software.

NA

1. Operations and retirement.

NA