Designation: F3201 – 16

Standard Practice for Ensuring Dependability of Software Used in Unmanned Aircraft Systems (UAS)[1]

This standard is issued under the fixed designation F3201; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ε) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 This standard practice intends to ensure the dependability of UAS software. Dependability includes both the safety and security aspects of the software.

1.2 This practice will focus on the following areas: (*a*) Organizational controls (for example, management, training) in place during software development. (*b*) Use of the software in the system, including its architecture and contribution to overall system safety and security. (*c*) Metrics and design analysis related to assessing the code. (*d*) Techniques and tools related to code review. (*e*) Quality assurance. (*f*) Testing of the software.

1.3 There is interest from industry and some parts of the CAAs to pursue an alternate means of compliance for software assurance for small UAS (sUAS).

1.4 This practice is intended to support sUAS operations. It is assumed that the risk of sUAS will vary based on concept of operations, environment, and other variables. The fact that there are no souls onboard the UAS may reduce or eliminate some hazards and risks. However, at the discretion of the CAA, this practice may be applied to other UAS operations.

1.5 *This standard does not purport to address all of the safety concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use.*

2. Referenced Documents

2.1 *FAA Standard:*[2]

FAA 23.1309–1E System Safety Analysis and Assessment for Part 23 Airplanes

2.2 *IEC Standard:*[3]

IEC 62304 Medical Device Software—Software Life Cycle Processes

2.3 *ISO Standards:*[4]

ISO 9001 Quality Management Systems—Requirements

2.4 *ICAO Standard:*[5]

ICAO 9859 Safety Management Manual

2.5 *NASA Standard:*[6]

NASA Technical Briefs Making Sense out of SOUP (Software of Unknown Pedigree)

2.6 *RTCA Standards:*[7]

RTCA DO-178C Software Considerations in Airborne Systems and Equipment Certification

RTCA DO–278A Software Integrity Assurance Considerations for Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems

RTCA DO-326 Airworthiness Security Process Specification

2.7 *Military Standards:*[8]

Department of Defense Joint Software System Safety Handbook

MIL-STD-882E Department of Defense Standard for System Safety

3. Terminology

3.1 *Definitions of Terms Specific to This Standard:*

3.1.1 *application programming interface (API)*—definition of the inputs and outputs for operations intended for use by other software modules.

3.1.2 *architecture*—architecture is made up of the definition of the sUAS Software components, the data that flows between

[1] This practice is under the jurisdiction of ASTM Committee F38 on Unmanned Aircraft Systems and is the direct responsibility of Subcommittee F38.01 on Airworthiness.

Current edition approved Sept. 1, 2016. Published September 2016. DOI: 10.1520/F3201-16.

[2] Available from Federal Aviation Administration (FAA), 800 Independence Ave., SW, Washington, DC 20591, http://www.faa.gov.

[3] Available from International Electrotechnical Commission (IEC), 3, rue de Varembé, P.O. Box 131, 1211 Geneva 20, Switzerland, http://www.iec.ch.

[4] Available from International Organization for Standardization (ISO), ISO Central Secretariat, BIBC II, Chemin de Blandonnet 8, CP 401, 1214 Vernier, Geneva, Switzerland, http://www.iso.org.

[5] Available from International Civil Aviation Organization (ICAO), 999 Robert-Bourassa Blvd., Montreal, Quebec H3C 5H7, Canada, http://www.icao.int.

[6] Available from U.S. National Air and Space Administration (NASA), 300 E. Street, SW, Suite 5R30, Washington, DC 20546, http://www.nasa.gov.

[7] Available from Radio Technical Commission for Aeronautics (RTCA), 1150 18th St., NW, Suite 910, Washington, DC 20036, http://www.rtca.org.

[8] Available from DLA Document Services, Building 4/D, 700 Robbins Ave., Philadelphia, PA 19111-5094, http://quicksearch.dla.mil.

the components (data flow), and the order of execution of the components (control flow).

3.1.3 *code churn*—the quantity and frequency of additions, deletions, and modifications to the source code for software.

3.1.4 *code coverage*—a measure used to describe the degree to which the source code of a program is tested by a particular test suite.

3.1.5 *customer*—includes stakeholders outside of the sUAS manufacturer who interface with the sUAS.

3.1.6 *dependability*—attribute of the software code that produces the consequences for which it was written, without adverse effects, in its intended environment.

3.1.7 *dynamic program analysis*—the practice of analyzing software while it is executing, for example monitoring memory access, allocation, and deallocation during program execution. For example, Valgrind is a popular open-source tool that performs this type of analysis.

3.1.8 *externally developed software (EDS)*—software developed outside of the sUAS manufacturer for which adequate records of the development process may not be available.

3.1.9 *EDS quality plan*—a plan to address the software quality in the event that EDS source code is not available. See Appendix X2 for more details.

3.1.10 *fuzz testing*—a testing technique wherein the input to a unit under test is unexpected in some way. Examples include testing with input that is invalid, unexpected, or random.

3.1.11 *internal user*—includes stakeholders within the sUAS manufacturer's organization who interface with the sUAS.

3.1.12 *internally developed software (IDS)*—software developed within the sUAS manufacturer's organization.

3.1.13 *penetration testing*—a testing method intended to identify and correct vulnerabilities and security defects by attempting to break, bypass, or tamper with software security controls.

3.1.14 *publish*—formalized release of a document to appropriate parties. A history should be maintained for published documents. The history may be part of revision control system, printed papers in a binder, or any other auditable system.

3.1.15 *quality assurance*—the practice of internally monitoring or auditing the development process.

3.1.16 *red team evaluation*—a process designed to detect network and system vulnerabilities and test security by taking an attacker-like approach to system, network, or data access, or combinations thereof.

3.1.17 *shall versus should versus may*—use of the word "shall" implies that a procedure or statement is mandatory and must be followed to comply with this practice, "should" implies recommended, and "may" implies optional at the discretion of the supplier, manufacturer, or operator. Since "shall" statements are requirements, they include sufficient detail needed to define compliance (for example, threshold values, test methods, oversight, and references to other standards). "Should" statements also represent parameters that could be used in safety evaluations, and could lead to development of future requirements. "May" statements are provided to clarify acceptability of a specific item or practice, and offer options for satisfying requirements.

3.1.18 *small unmanned aircraft system (sUAS)*—composed of small unmanned aircraft (sUA-see 4.2) and all required on-board subsystems, payload, control station, other required off-board subsystems, any required launch and recovery equipment, all required crew members, and command and control (C2) links between sUA and the control station.

3.1.19 *sUAS manufacturer*—the organization and personnel with design responsibility for the sUAS, including the dependability of the system software.

3.1.20 *sUAS Software*—includes both IDS and EDS.

3.1.21 *software baseline*—a known state of product software that has been formally reviewed and agreed on, that thereafter serves as the basis for further development, and can be changed only through formal change control procedures.

3.1.22 *software vulnerability*—a mistake in software (also known as a weakness) that can be directly exploited to get a cyber-enabled capability to function in an unintended manner. Typically this is the violation of a reasonable security policy for the cyber-enabled capability resulting in a negative technical impact. Although all vulnerabilities involve a weakness, not all weaknesses are vulnerabilities. For example, Common Vulnerabilities and Exposures is a dictionary of common names for publicly known software-related vulnerabilities.

3.1.23 *statement coverage*—a testing technique that involves the execution of all the statements at least once in the source code. As a metric, it is used to calculate and measure the number of statements in the source code which have been executed.

3.1.24 *threat modeling*—a structured approach that enables the sUAS manufacturer to identify, quantify, and address the security risks associated with an application. The process involves systematically identifying security threats and rating them according to severity and level of occurrence probability. The overall goal for threat modeling (also known as attack modeling) is the creation of customized knowledge about potential attacks relevant to the application or organization. This customized knowledge guides decisions about changes to the code and security controls to implement.

3.1.25 *tier 1 requirements*—required tasks and activities in this practice for a software malfunction or penetration that would result in a slight reduction in sUAS functional capabilities or safety margins (for reference see Minor failure conditions per AC 23.1309–1E).

3.1.26 *tier 2 requirements*—required tasks and activities in this practice for any software malfunction or penetration that would result in a significant reduction in sUAS functional capabilities or safety margins with potential for injury (for reference see Major failure conditions per AC 23.1309–1E).

3.1.27 *tier 3 requirements*—required tasks and activities in this standard for any software malfunction or penetration that would result in a large reduction in sUAS functional capabilities or safety margins and could be expected to result in serious

injury or fatality (for reference see Hazardous or more severe failure conditions per AC 23.1309–1E).

3.2 *Acronyms:*

3.2.1 *API*—Application Programming Interface

3.2.2 *CAA*—Civil Aviation Authority

3.2.3 *EDS*—Externally Developed Software

3.2.4 *FAA*—Federal Aviation Administration

3.2.5 *IDS*—Internally Developed Software

3.2.6 *sUA*—Small Unmanned Aircraft

3.2.7 *sUAS*—Small Unmanned Aircraft System

3.2.8 *UAS*—Unmanned Aircraft System

4. Applicability

4.1 The practice is written for all UAS intended for operation within airspace controlled by a CAA.

4.2 It is assumed that the maximum weight and airspeed of a sUAS will be specified by the nation's CAA. However, unless otherwise specified by a nation's CAA, this practice applies only to sUA that:

4.2.1 Have a maximum takeoff gross weight of 55 lb (25 kg) or less;

4.2.2 Have the capability to allow remote intervention by flight personnel in the management of the flight during normal operations.

4.3 This practice is intended for software that is part of a sUAS. It may be used by itself or in conjunction with other standards such as DO-178C, as deemed appropriate by the sUAS manufacturer in accordance with CAA guidance. This practice does not replace or supersede other standards, hence a sUAS manufacturer may choose to certify under alternatives such as DO-178C without reference to this practice, subject to CAA guidance. Appendix X1 contains guidance for producing artifacts corresponding to the requirements in Section 5.

4.4 The applicability of the practice extends to those software items in the sUAS that implement functions essential to safety. Software items that have no impact on safety are out of scope for this practice. For example, payload software on the sUAS that is not used to perform a safety-critical function is outside the scope of this practice.

5. Requirements

NOTE 1—The hazard analysis (see 5.2.1) will be used to determine the severity of a sUAS Software malfunction or failure and the corresponding tier. See Appendix X2 for examples of tier assignments to sUAS functions.

NOTE 2—The applicability of the each requirement is determined by the tier assignment and noted in parentheses next to the requirement. Unless otherwise indicated, sub-requirements inherit the tier assignments of the parent requirement (for example, if requirement 5.2.1 applies to Tiers 1, 2, and 3, then 5.2.1.1 also applies to the same tiers).

NOTE 3—Requirements may apply only to EDS, only to IDS, or to the sUAS Software (includes EDS and IDS). Unless otherwise indicated, sub-requirements apply to the kind of software (EDS, IDS, or sUAS Software) specified in the parent requirement. See Appendix X3 for scenarios for using this practice for EDS, IDS, and sUAS Software.

5.1 *Organizational Planning:*

5.1.1 *Tier 1, 2, 3*—The sUAS manufacturer shall publish an organizational software plan for sUAS Software.

5.1.1.1 This plan shall define the roles and responsibilities of each part of the manufacturer's organization involved in software acquisition, development, integration, and testing for all sUAS software projects in the organization.

5.1.1.2 The sUAS manufacturer should educate company executives and train employees on the risks and vulnerabilities of the EDS integration or software development approach, or both.

5.1.2 *Tier 2, 3*—The sUAS manufacturer shall record all uses and versions of EDS in the sUAS.

5.1.3 *Tier 3*—The sUAS manufacturer shall have an organizational response plan to address a flight critical software malfunction or penetration for sUAS Software.

5.1.3.1 The sUAS manufacturer should make information available to all users of its sUAS regarding the software issue and provide guidance within 24 hours of being made aware of the issue.

5.2 *sUAS Software Architecture and Use:*

5.2.1 *Tier 1, 2, 3*—The sUAS manufacturer shall conduct an analysis to determine the hazards and impacts associated with the potential malfunction, failure, or exploitation of the sUAS Software and identify potential risk mitigation.

5.2.1.1 The analysis shall define the sUAS Software's intended function(s) and document potential failure (gracefully or suddenly).

5.2.1.2 The analysis should be conducted using industry best practices (see references in Appendix X1) but should consider unique aspects of the sUAS size and operation.

5.2.1.3 Security vulnerabilities should be considered as possible causes of hazards in performing the hazard analysis.

5.2.2 *Tier 2, 3*—The sUAS manufacturer shall publish an EDS integration plan.

5.2.2.1 The plan shall document the tasks and milestones that need to be performed to acquire and integrate the EDS.

5.2.2.2 The plan should include release gates/checkpoints/milestones and associated criteria at one or more points during the acquisition and integration process, as well as configuration management for all EDS code and documents.

5.2.2.3 The EDS integration plan may be part of a larger software integration plan or other lifecycle documentation.

5.2.2.4 The EDS integration plan should address how configuration tables, data, and libraries that may be included in the EDS are integrated.

5.2.2.5 The sUAS manufacturer shall ensure that the EDS integration plan is folowed and track exceptions to the plan.

5.2.2.6 All exceptions to the EDS integration plan shall be incorporated into the plan and published.

5.2.3 *Tier 1, 2, 3*—The sUAS manufacturer shall publish a software development and integration plan for IDS.

5.2.3.1 The IDS development and integration plan shall establish the software baseline and document the tasks and milestones that need to be performed to develop the software for a specific project.

5.2.3.2 The IDS development and integration plan should include release gates/checkpoints/milestones and associated criteria at one or more points in the development process, and configuration management of code and documents.

5.2.3.3 The IDS development and integration plan should address the pedigree of software development and testing tools and the development and testing of configuration tables, data, and libraries.

5.2.3.4 *Tier 2, 3 only*—The sUAS manufacturer shall ensure that the IDS development integration plan is followed and track exceptions to the plan.

5.2.3.5 *Tier 2, 3 only*—All exceptions to the IDS development and integration plan shall be incorporated into the plan and published.

5.2.4 *Tier 2, 3*—The sUAS manufacturer shall publish an EDS maintenance plan that documents the criteria and method for how patches, bug fixes, upgrades, etc. provided by the EDS supplier will be applied.

5.2.4.1 The sUAS manufacturer shall perform maintenance on the EDS per the EDS maintenance plan.

5.2.4.2 The EDS maintenance plan shall contain processes that are performed at least annually. These processes may include reviewing EDS changes, release notes, bug fixes, etc.

5.2.4.3 The EDS maintenance plan shall include the process that ensures that the executable EDS object code and included configuration tables, data, and libraries are properly loaded in the appropriate computer(s) in the sUAS.

5.2.4.4 The EDS maintenance plan should include how configuration management will be leveraged to record changes to the EDS throughout its use.

5.2.4.5 The EDS maintenance plan may be part of a larger maintenance plan or other lifecycle documentation.

5.2.4.6 The sUAS manufacturer shall ensure that the EDS maintenance plan is followed and track exceptions to the plan.

5.2.4.7 All exceptions to the EDS maintenance plan shall be incorporated into the plan and published.

5.2.5 *Tier 1, 2, 3*—The sUAS manufacturer shall publish an IDS maintenance plan that documents how patches, bug fixes, upgrades, etc. will be applied.

5.2.5.1 The IDS maintenance plan should include how configuration management will be leveraged to record changes to the software throughout its use.

5.2.5.2 The IDS maintenance plan shall include the process that ensures that the executable object code and included configuration tables, data, and libraries are properly loaded in the appropriate computer(s) in the sUAS.

5.2.5.3 The IDS maintenance plan shall address changes to software baselines, archival and retrieval processes, load control onto the sUAS, and tools related to the maintenance of IDS.

5.2.5.4 The IDS maintenance plan may be part of a larger maintenance plan or other lifecycle documentation.

5.2.5.5 *Tier 2, 3 only*—The sUAS manufacturer shall ensure that the IDS maintenance plan is followed and track exceptions to the plan.

5.2.5.6 *Tier 2, 3 only*—All exceptions to the IDS maintenance plan shall be incorporated into the plan and published.

5.2.6 *Tier 2, 3*—The sUAS manufacturer should perform and market survey to determine what options, if any, are available to fill the functionality with EDS.

5.2.6.1 The EDS market survey should seek to determine the best of breed software options and their characteristics (for example, trade space between size, features, cost, speed, etc.) in order to understand the chosen features and complexity of the EDS compared with other available software.

5.2.7 *Tier 2, 3*—The sUAS manufacturer shall review and document sUAS Software architecture and the sUAS Software functional, interface, and performance requirements.

5.2.7.1 Performance requirements may be stated in terms of timing, precision, etc.

5.2.7.2 Functional and interface requirements may be assessed via the sUAS Software architecture.

5.2.7.3 The architecture should show the relationship of the sUAS Software function to total system function including all interfaces.

5.2.7.4 The architecture should show any mechanism for fail-safe or redundancy, or both, for sUAS Software.

5.2.7.5 The sUAS manufacturer may use traditional (for example, “shall” statements) or non-traditional (for example, in the form of user stories) requirements.

5.2.8 *Tier 2, 3*—The sUAS manufacturer shall neutralize unwaranted functionality of the sUAS Software by disabling, removing, or mitigating, or combinations thereof, risk associated with functions and features that are not needed for the intended function of the sUAS Software.

5.2.8.1 If code is disabled or removed, the sUAS manufacturer shall confirm through regression tests (see 5.6.4) the disabling or removal of those functions does not adversely affect the sUAS Software.

5.2.8.2 The sUAS manufacturer may write unit tests to ensure that unneeded code does not adversely affect the system.

5.2.9 *Tier 3*—The sUAS manufacturer shall establish and utilize a formal, documented process by which internal users and customers can report problems and have them resolved for sUAS Software.

5.2.9.1 The formal process should include resolution tracking and have visibility across developers and project managers.

5.2.9.2 If EDS is being integrated, the sUAS manufacturer should have a method to contact the EDS supplier with problem reports.

5.2.10 *Tier 3*—The sUAS manufacturer shall document and track the sUAS Software’s relevant service history where possible, including how many problems have been reported and fixed over time, and how many hours the software has been operated in similar context with user problem reporting in place.

5.2.11 *Tier 3*—The sUAS manufacturer shall implement a protection mechanism to mitigate the effects of EDS failure or exposure.

5.2.11.1 The sUAS manufacturer shall document and review the protection mechanism requirements.

5.2.11.2 The sUAS manufacturer shall verify the development and implementation of the protection mechanism and record the percentage of known vulnerabilities that the protection mechanism is effective against.

5.2.11.3 The sUAS manufacturer should justify the rationale for any unprotected vulnerability, error, or failure condition.

5.2.11.4 The sUAS manufacturer may implement execution wrapper software, employ middleware, monitor software, or use other means to protect the system.

5.2.12 *Tier 3*—The sUAS manufacturer shall use threat modeling to identify high risk areas in the sUAS Software.

5.2.12.1 The sUAS manufacturer should document and review data flow diagrams, call graphs, and other system visualizations.

5.2.12.2 The sUAS manufacturer should use visualizations at multiple levels of detail (for example, system, component, module), and these should be sufficient to understand relevant threats and vulnerabilities.

5.2.12.3 The sUAS manufacturer should determine countermeasures that can protect the system from exploits.

5.3 *Detailed Design Analysis:*

5.3.1 *Tier 2, 3*—The sUAS manufacturer shall record the amount of code churn for sUAS Software.

5.3.1.1 The sUAS manufacturer should ensure that the code churn trend is understood and investigate the root cause for any irregularities.

5.3.2 *Tier 3*—If EDS source code is available, the sUAS manufacturer shall produce a call graph by performing a call trace from all EDS API entry points that are used all the way through the EDS source and review and document risk areas, and determine the suitability of the EDS for the intended function and tier.

5.3.2.1 For EDS for which source code is not available, the sUAS manufacturer shall examine the usage history of the EDS in similar application. The sUAS manufacturer shall also review results of bench testing on the target hardware for all intended functions. Anomalies or unexpected results require root cause investigation and remediation.

5.3.2.2 Where practicable, the sUAS manufacturer should document system calls made by all EDS for all intended functionality.

5.3.3 *Tier 2, 3*—The sUAS manufacturer shall review the available documentation of the EDS, including description, code comments, and release notes where possible and determine the suitability of the EDS for the intended function and tier.

5.3.4 *Tier 2, 3*—The sUAS manufacturer shall ensure adequate documentation of the IDS, including description, code comments, and release notes for IDS.

5.3.5 *Tier 3*—The sUAS manufacturer shall review and record anomaly, bug, or problem reports, or combinations thereof, related to sUAS Software and their resolutions to better understand past performance and quality.

5.3.5.1 The sUAS manufacturer should track defects over time, and note defect root cause, prioritization, and repair.

5.4 *Code Review:*

5.4.1 *Tier 2, 3*—Source code may not be available for some EDS which would preclude meeting some or all requirements in this section. In the event that requirements cannot be met due to lack of EDS source code, the sUAS manufacturer shall produce a EDS Quality Plan that explicitly addresses how dependability is achieved without access to the source code. See Appendix X1 for required information for the EDS Quality Plan.

5.4.2 *Tier 1, 2, 3*—The sUAS manufacturer shall create a list of potential sUAS software vulnerabilities (for example, authentication routines, data validation code, dynamic memory allocation, etc.) that will be used to conduct code reviews.

5.4.3 *Tier 2, 3*—The sUAS manufacturer shall at least annually perform research (for example, internet search) to discover any known vulnerabilities and issues in the EDS and determine the suitability of the EDS for the intended function and tier.

5.4.4 *Tier 2, 3*—The sUAS manufacturer shall use one or more automated static code analysis tool(s) to review the IDS and adherence to coding standards, review detected issues, and address as necessary.

5.4.5 *Tier 2, 3*—The sUAS manufacturer shall use one or more automated static code analysis tool(s) to review the EDS, review detected issues, and address as necessary.

5.4.6 *Tier 2, 3*—The sUAS manufacturer shall document the coding standards used for development or evaluation of sUAS Software.

5.4.6.1 The sUAS manufacturer should identify language-specific issues that could affect the software performance.

5.4.6.2 Multiple coding standards may be used within the sUAS Software.

5.4.7 *Tier 2, 3*—The sUAS manufacturer shall perform a manual review of the sUAS Software source code based on the vulnerability list for the sUAS Software.

5.4.7.1 For sUAS Software, the sUAS manufacturer shall review the source code against the coding standards used.

5.4.8 *Tier 3*—The sUAS manufacturer shall document code coverage results from requirements-based or unit testing to assess if all parts of the sUAS Software are executed.

5.4.8.1 The sUAS manufacturer should ensure that the level of code coverage be at least statement code coverage. Code coverage less than 100 % should be justified.

5.4.9 *Tier 3*—The sUAS manufacturer shall perform an analysis of the code complexity of the sUAS Software to identify areas with a higher probability of defects.

5.4.9.1 The sUAS manufacturer should review the complexity metrics with software integrators, developers, and management to understand the complexity's contribution to probability of defect.

5.4.9.2 The sUAS manufacturer may calculate cyclomatic complexity for each function performed; values less than 20 typically possess a low probability of defect where values greater than 50 typically possess a high probability of defect.

5.5 *Quality Assurance:*

5.5.1 *Tier 2, 3*—Development history, software life cycle process, and requirement traceability may not be available for some EDS which would preclude meeting some or all requirements in this section. In the event that the requirements cannot be met due to lack of EDS information, the sUAS manufacturer shall produce a EDS Quality Plan that explicitly addresses how dependability is achieved without access to this information. See Appendix X1 for required information for the EDS Quality Plan.

5.5.2 *Tier 1, 2, 3*—The sUAS manufacturer shall establish a software quality assurance process for IDS.

5.5.2.1 The sUAS manufacturer shall review its IDS Quality Assurance process against an industry standard (such as International Organization for Standardization (ISO), International Civil Aviation Organization (ICAO) Safety Management System (SMS), Capability Maturity Model (CMM), etc.).

5.5.3 *Tier 1, 2, 3*—The sUAS manufacturer shall ensure that its personnel are competent, trained, and understand the importance of their activities to sUAS Software dependability.

5.5.3.1 The sUAS manufacturer shall determine the competency and training requirements for its personnel.

5.5.3.2 The sUAS manufacturer shall at least annually review the effectiveness of training and other activities to promote competency within the organization.

5.5.4 *Tier 2, 3*—The sUAS manufacturer shall review the EDS Supplier's Software Life Cycle and EDS development history.

5.5.4.1 The sUAS manufacturer should review artifacts to confirm that the EDS supplier follows their life cycle process (for example, source code change control, peer reviews, requirements reviews, quality processes).

5.5.5 *Tier 2, 3*—The sUAS manufacturer shall document the Software Life Cycle and development history for IDS.

5.5.5.1 The sUAS manufacturer should produce artifacts to show that they follow their life cycle process (for example, source code change control, peer reviews, requirements reviews, quality processes).

5.5.6 *Tier 3*—The sUAS manufacturer shall review traceability between sUAS Software requirements and associated tests.

5.5.6.1 The sUAS manufacturer shall identify and provide justification for any tests that do not trace to requirements and any requirements that do not trace to tests.

5.6 *Testing:*

5.6.1 *Tier 1, 2, 3*—The sUAS manufacturer shall define verification and validation (V&V) plans and procedures for the sUAS Software.

5.6.1.1 The sUAS manufacturer should ensure these plans and procedures account for demonstrating the software's intended function and account for the potential of hazardously misleading information.

5.6.1.2 The sUAS manufacturer may use automated tools to define and enforce the V&V procedures.

5.6.2 *Tier 1, 2, 3*—The sUAS manufacturer shall verify that the sUAS Software is completely and correctly integrated into the system through testing as defined in the V&V plan and documenting associated test results.

5.6.2.1 The sUAS manufacturer shall review all test results and address all failed tests, or justify why the failed tests do not present a hazard.

5.6.2.2 The sUAS manufacturer shall record the results of tests in a defect management system for resolution.

5.6.2.3 The V&V testing should ensure that the executable object code and included configuration tables, data, and libraries required for the intended function are correct and complete, and are properly executed in the appropriate computer(s) in the sUAS, using actual or emulated peripheral equipment to the maximum extent possible.

5.6.3 *Tier 1, 2, 3*—The sUAS manufacturer shall perform external penetration testing to check for both safety and security vulnerabilities for all interfaces of the sUAS that are an external attack surface.

5.6.3.1 The sUAS manufacturer shall review all test results and address all penetrations through software changes or justify why the failed tests do not present a hazard.

5.6.4 *Tier 1, 2, 3*—The sUAS manufacturer shall conduct regression testing after any sUAS Software change.

5.6.4.1 The sUAS manufacturer shall review all test results and address all failed tests or justify why the failed tests do not present a hazard.

5.6.4.2 Regression testing may include automatic testing when committing the code.

5.6.5 *Tier 2, 3*—The sUAS manufacturer shall perform robustness testing of the sUAS Software.

5.6.5.1 The sUAS manufacturer should utilize dynamic program analysis, fuzz testing, out-of-range inputs and timing, abnormal system initialization conditions, failure modes involving incoming data, fault injection, overflow protection, invalid state transitions, and other stressing cases to test the robustness of the software.

5.6.5.2 Robustness tests should be run on the code as it is being executed on the target hardware to find unexpected interactions between system components, safety and security issues, and unexpected faults.

5.6.5.3 The sUAS manufacturer shall review all test results and address all failed tests or justify why the failed tests do not present a hazard.

5.6.6 *Tier 3*—The sUAS manufacturer shall use internal subsystem-level penetration testing to find sUAS Software problems.

5.6.6.1 The sUAS manufacturer should provide penetration testers with all available information about their target.

5.6.6.2 The sUAS manufacturer shall review all test results and address all penetrations through software changes or justify why the failed tests do not present a hazard.

5.6.7 *Tier 3*—The sUAS manufacturer shall perform periodic red team evaluations of the sUAS Software, or justify (based on an unique risk assessment) why red team evaluations are not needed.

5.6.7.1 The unique risk assessment should consider the likelihood of attack, the environment in which the sUAS operates (for example, rural, urban), and public safety implications.

5.6.7.2 The sUAS manufacturer shall perform a root cause analysis and review results for any successful exploitation of the software and undertake remedial action.

6. Keywords

6.1 airworthiness; safety; security; small unmanned aircraft system; software

APPENDIXES

(Nonmandatory Information)

X1. GUIDANCE FOR SOFTWARE ARTIFACTS

X1.1 See Table X1.1

TABLE X1.1 Guidance for Software Artifacts

NOTE 1—The interim guidance column is intended to provide useful resources in assisting the sUAS manufacturer in producing artifacts. It should not be construed to mandate specific content, procedures, or format of the artifact.

Requirement	Artifact	Interim Guidance
5.1 *Organizational Planning*		
5.1.1	Organizational Software Plan	IEC 62304 Appendix B.4; BSIMM SM Level 1
5.1.2	Record of All Uses and Versions of EDS	—
5.1.3	Organizational Response Plan	BSIMM CMVM (All Levels)
5.2 *sUAS Software Architecture and Use*		
5.2.1	Hazard Analysis	SAE ARP 4761; MIL-STD-882E; IEC 62304 Section 7.1; ASTM WK49619 Standard Practice for Operational Risk Assessment of sUAS (pending ballot approval)
5.2.2	EDS Integration Plan	IEC 62304 Appendix B.5
5.2.3	Software Development / Integration Plan	IEC 62304 Appendix B.5
5.2.4	Software Maintenance Plan	IEC 62304; Appendix B.6 FAA AC 120-76B Section 13
5.2.5	Software Maintenance Plan	IEC 62304; Section 6 FAA AC 120-76B Paragraph 13 (i) and (j)
5.2.6	—	—
5.2.7	Software Requirements Document; Architecture Diagram	IEC 62304 Appendix B.5
5.2.8	Software Development / Integration Plan	IEC 62304 Appendix B.5
5.2.9	Problem Reporting Plan	RTCA DO-178C 11.17
5.2.10	Service History	—
5.2.11	Evidence of Review[A]	—
5.2.12	Threat Modeling Diagrams	NASA "Making Sense out of SOUP" Report; https://www.owasp.org/index.php/Application_Threat_Modeling
5.3 *Detailed Design Analysis*		
5.3.1	Record of Code Churn	—
5.3.2	Evidence of Review[A]	—
5.3.3	Evidence of Review[A]	—
5.3.4	Software Documentation	IEC 62304 Appendix B.5
5.3.5	Anomaly and Bug Reports	—
5.4 *Code Review*		
5.4.1	EDS Quality Plan	See description below
5.4.2	Vulnerability List	BSIMM CR.1.1; RTCA DO-326 Sec 2.3, 2.4.1
5.4.3	Evidence of Review[A]	—
5.4.4	Tool Documentation	BSIMM CR 1.4 and 2.6
5.4.5	Tool Documentation	BSIMM CR 1.4 and 2.6
5.4.6	Coding Standard	BSIMM SR 1.4; MISRA C and C++; JSF AV C++ Coding Standards
5.4.7	Evidence of Review[A]	—
5.4.8	Code Coverage Documentation	DO-178C Sec 6.4.4; NASA "Making Sense out of SOUP" Report
5.4.9	Complexity Analysis	DO-278A Sec 6.3.4(d)
5.5 *Quality Assurance*		
5.5.1	EDS Quality Plan	See description below
5.5.2	Software Quality Assurance Process	DO-278 Sec 8.0
5.5.3	Training Records	ISO 9001 Clause 5.2.2
5.5.4	Evidence of Review[A]	—
5.5.5	Software Life Cycle Document	IEC 62304 Appendix B.1; Joint Software System Safety Handbook Sec 2.5.3
5.5.6	Requirements Trace Matrix	DO-178C Sec 5.5
5.6 *Testing*		
5.6.1	V&V Testing Plan and Procedures	NASA-STC-8719.13C Appendix F; NASA NPR 7150.2A; RTCA DO-326 Sec 2.7.4.3
5.6.2	Evidence of Review[A]	—
5.6.3	Evidence of Review[A]	—
5.6.4	Evidence of Review[A]	—
5.6.5	Robustness Test Plan; Evidence of Review[A]	RTCA DO-326 Sec 2.7.4
5.6.6	Penetration Test Plan; Evidence of Review[A]	RTCA DO-326 Sec 2.7.4; BSIMM PT 2.3
5.6.7	Red Team Plan and Evidence of Review[A] ; or Unique Risk Assessment	BSIMM PT 3.1; See Requirement 5.2.1 for guidance on Risk Assessment

[A]Evidence of Review is an artifact that documents the requirement review and approval. It must include material reviewed, successful resolution, and closure of all action items required by the reviewers for approval, and record of reviewer approval including name and date.

X1.2 *EDS Quality Plan (may be required under 5.4 and 5.5)*—The EDS Quality Plan must explicitly address how software dependability is achieved without access to the source code. **Section 1** of this plan must show how the safety and

security of the sUAS will not be compromised due to an EDS malfunction or penetration through protection mechanisms per requirement 5.2.11 of this practice. **Section 2** of EDS Quality Plan must contain the description and results of robustness testing per requirement 5.6.5 of this practice. **Section 3** of the EDS Quality Plan must include any relevant information regarding usage history of the EDS in similar contexts (see 5.2.10 and 5.3.2.1 of this practice) and any relevant information regarding the development practices and software life cycle management practices of the EDS supplier.

X2. EXAMPLE OF ASSIGNMENT OF sUAS FUNCTIONS TO TIERS

X2.1 See Fig. X2.1 and Fig. X2.2.

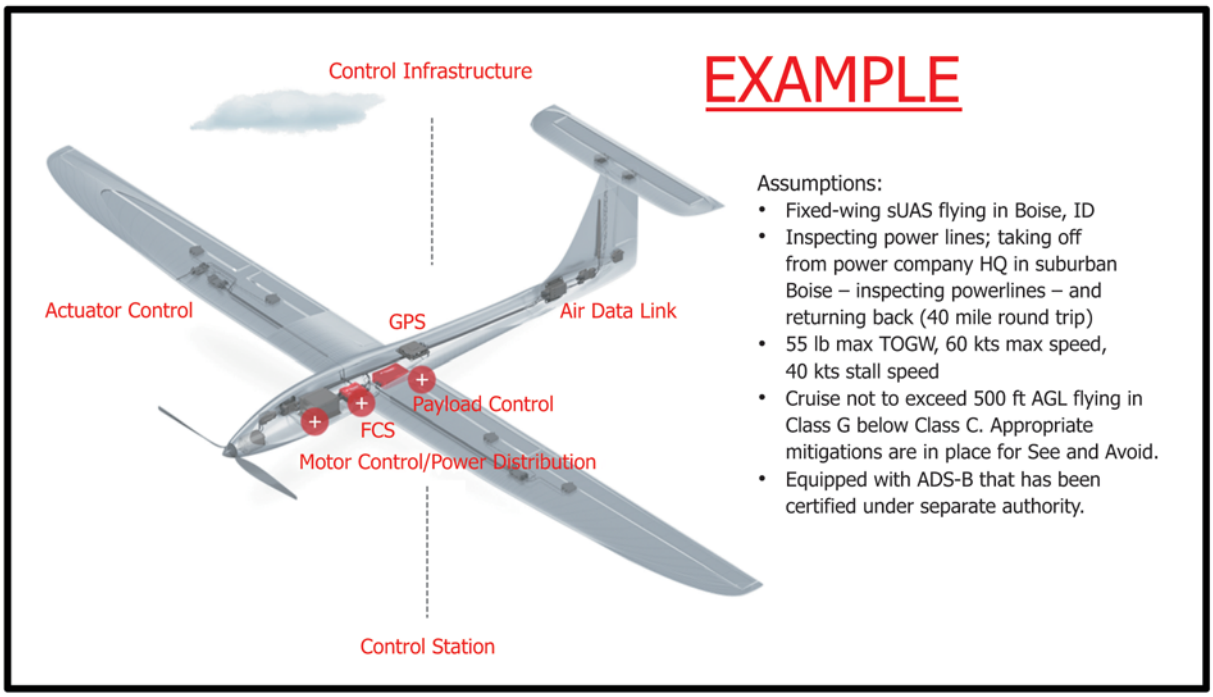
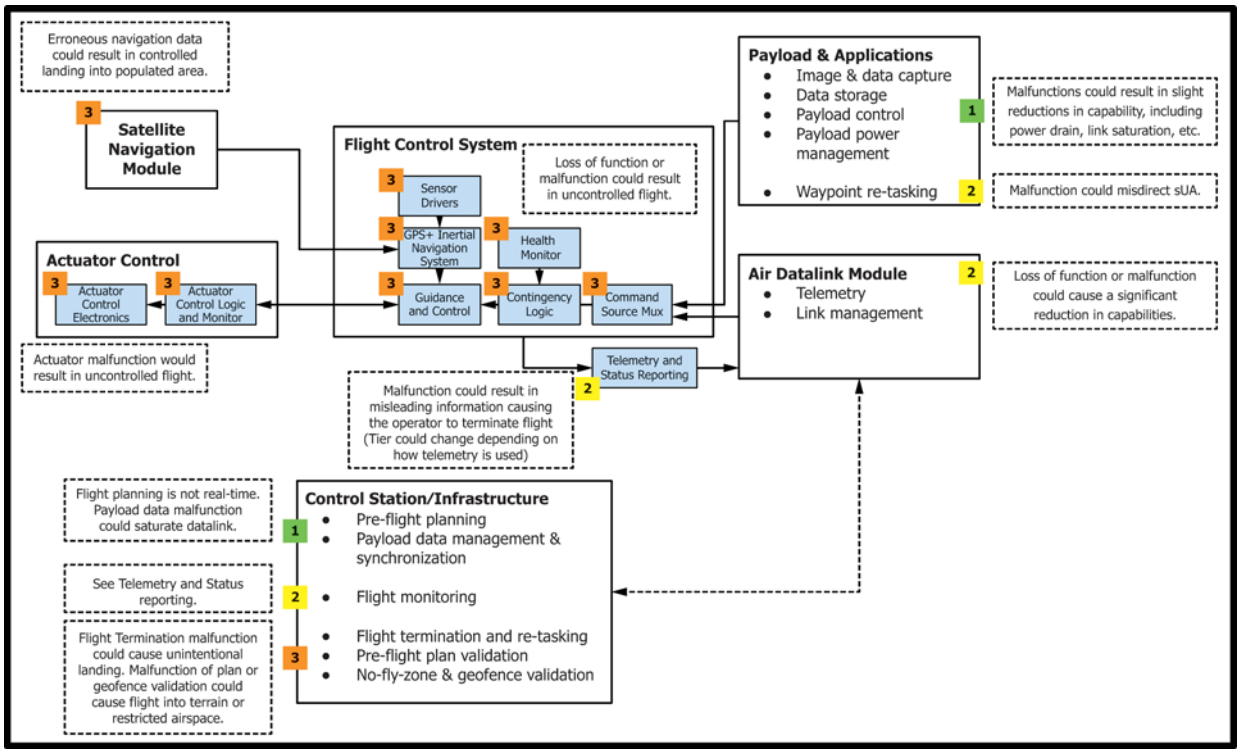


FIG. X2.1 Example sUAS Architecture and Operational Assumptions



Key:
3–Tier 3
2–Tier 2
1–Tier 1

FIG. X2.2 Example Tier Assignments to sUAS Functions

X3. USAGE SCENARIOS FOR THIS PRACTICE

X3.1 See Table X3.1.

TABLE X3.1 Usage Scenarios for this Practice

Scenario	Description	Externally Developed Software	Internally Developed Software	Comments
1	sUAS manufacturer intends to integrate open source map tool into the sUAS navigation system.	YES	NO	Assumes other components of sUAS have already obtained airworthiness approval
2	sUAS manufacturer had previously developed its own flight control and navigation software for its sUAS but adequate records of the development processes are not avaiable	NO	YES	Generating the necessary artifacts per this practice would be required.
3	sUAS manufacturer desires to update its flight control software with a 3rd party Kalman filter.	YES	YES	sUAS manufacturer would use EDS requirements for the Kalman filter and IDS requirements for its flight control software.
4	sUAS manufacturer desires to purchase firmware with no visibility into the software source code.	YES	NO	The sUAS manufacturer would develop a EDS Quality Plan per 5.4 and 5.5 of this practice.
5	sUAS manufacturer is using software developed in another division of its parent company.	YES	NO	Even though this software was developed “internally,” it was not developed for the intended function and should be treated as EDS.
6	sUAS manufacturer is developing its own code and wants to integrate RTCA DO-178C certified software into the sUAS.	YES	NO	The sUAS manufacturer could use the EDS requirements or decide to not use this practice and follow DO-178C for all of its software.
7	sUAS manufacturer wants to use a cloud-based system for sending route optimization/path commands to the sUAS.	NO	NO	The cloud software should not be considered part of the sUAS but the software requirements should include some reasonableness check on the inputs received from the cloud.

RELATED MATERIAL

Cook, S., Buttner, D., and Lester, E., "Dependability of Software of Unknown Pedigree," AIAA-2015-1867, AIAA Software Challenges in Aerospace, Orlando, FL, 2015.

Cook, S., et al, "Dependability of Software of Unknown Pedigree: Case Studies on Unmanned Aircraft Systems," Proceeding from the Digital Aviation Systems Conference, September 2015.