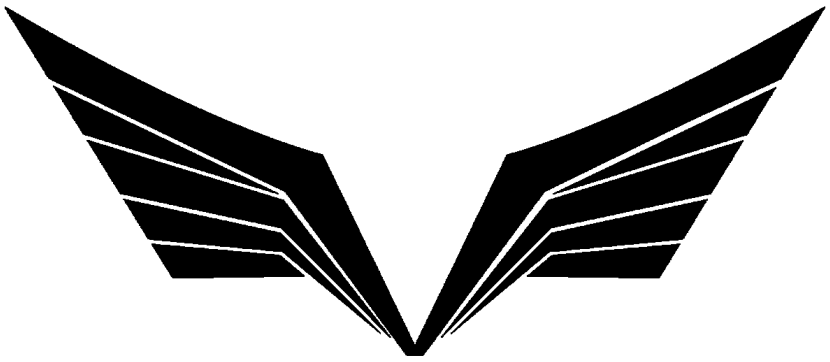

SUN DEVIL
ROCKETRY

Avionics Team

Software Quality Assurance Standard

Revision:	1.0
Date:	August 24, 2026
Drafted By:	Eli Sells
Avionics Lead	Eli Sells

TABLE OF CONTENTS

1. PURPOSE **3**
2. QUALIFICATION LEVEL DEFINITION **4**
3. QUALITY ASSURANCE REQUIREMENTS **5**
3.1. QA DOCUMENT 6
3.2. QA AUDIT 6
3.3. FUNCTIONAL HAZARD ANALYSIS 6
3.4. SECURITY ANALYSIS 7
3.5. RISK NULLIFICATION 7
3.6. CODE REVIEWS 7
3.7. SYSTEM REQUIREMENTS & TRACING 7
3.8. SYSTEM TESTING 7
3.9. INTEGRATION PERIODS 8
4. QUALITY ASSURANCE MILESTONE PROCEDURES **9**
4.1. DESIGN MILESTONE CHECKLIST 10
4.2. TEST READINESS MILESTONE CHECKLIST 11
4.2.1. TEST READINESS QA AUDIT CHECKLIST 12
4.3. FLIGHT QUALIFICATION CHECKLIST 13
5. VERSION HISTORY **14**

1. PURPOSE

This document defines the quality assurance standard for Sun Devil Rocketry's Avionics team, as well as quality-controlled software projects elsewhere in the club.

Many of SDR's software projects have a tangible impact on the safety of a flight or test. As such, procedures are needed to ensure that software satisfies its requirements and is robust enough to handle unexpected failure conditions. The procedures defined within this document help to mitigate the risk of a test/mission failure, as well as to mitigate any hazards that personnel or other equipment are exposed to during the system's operation.

2. QUALIFICATION LEVEL DEFINITION

Not every software component in use by avionics ends up on a rocket with an impact on flight dynamics or in a liquids test controlling the flow of propellants. Many are used outside of these contexts to analyze data after a test, to interface with other components, to act as a utility to make software development processes easier, or to test other components. Every software component in use at Sun Devil Rocketry is categorized as one of the following:

LEVEL	DESCRIPTION	EXAMPLES
Safety Critical	Failure of the system can affect safety margins.	A computer controlling parachute deployment, software to accept an abort command.
Mission Critical	Failure of the system can result in a failure of the test.	A computer broadcasting the location of a rocket for recovery, a data logger.
Not Critical	Failure of the system has no effect on safety margins or the test.	Internal tooling, post-flight data analysis

3. QUALITY ASSURANCE REQUIREMENTS

The following is an exhaustive list of the organization's **REQUIRED** QA practices for a given piece of software. Not all of these requirements are for every level of criticality – some apply only for the highest level, some only apply to lower level projects. Some projects may choose to adopt additional practices from this list or their own separate practices. That is allowed, so long as the below remain satisfied.

PRACTICE	DEVELOPMENT PHASE	LEVEL(s)
QA Document	All	Safety Critical
Independent QA Audit	Development	Safety Critical
Functional Hazard Analysis	Design, Development	Safety Critical
Security Analysis (can be included in FHA)	Design, Development	Safety Critical
Risk Nullification	Design	Mission Critical, Not Critical
Code Reviews	Development	Safety Critical, Mission Critical
System Requirements & Tracing	Design, Development, Testing	Safety Critical
System Testing	Testing	Safety Critical, Mission Critical
Integration Periods	Development, Testing	Safety Critical

3.1. QA DOCUMENT

As part of the quality assurance compliance process, projects at the safety critical level will be required to produce a quality control document that is checked into version control. Each quality control document shall have proof of compliance during each phase of development, including sections for:

1. [DESIGN PHASE] A brief overview of the project, plus the relevant standard it complies with. If any modifications are made to the quality assurance standard for that project, these are disclosed.
2. [DESIGN PHASE] Any supporting documentation for the project, such as a functional hazard analysis or user information.
3. [DESIGN PHASE] System requirements or a link to the system requirements document.
4. [DESIGN REVIEW] A page where the project engineer and a QA delegate clear the project for development.
5. [DEVELOPMENT PHASE] The components of the system.
6. [DEVELOPMENT PHASE] Test procedures to verify the system requirements.
7. [TEST READINESS REVIEW] A page where the project engineer and a QA delegate clear the project for testing.
8. [TESTING PHASE] Results of the test procedures.
9. [FLIGHT READINESS REVIEW] A page where the project engineer and a QA delegate clear the project for flight.

3.2. QA AUDIT

For safety critical projects, an independent quality assurance reviewer will analyze the project's compliance with this standard. Each project will have a QA audit after the development that takes a random sample of commits and analyzes whether it has met the required QA standard.

3.3. FUNCTIONAL HAZARD ANALYSIS

Safety critical projects require a functional hazard analysis before beginning development. This analysis should, at the design level, identify weaknesses in the system's design and mitigations that should be implemented to prevent harm to operators and bystanders. The mitigations defined in this document will be required to be tested alongside the system requirements.

3.4. SECURITY ANALYSIS

As part of the functional hazard analysis OR in a separate document, a security analysis should be performed. This should identify attack vectors on the system by malicious actors and attempt to mitigate them to prevent harm to operators and bystanders.

3.5. RISK NULLIFICATION

For projects not classified as safety critical, some documentation should be produced that explains why there is no safety impact from that project.

3.6. CODE REVIEWS

All functional software changes on a safety critical project MUST go through a code review process. It is highly recommended that all functional software changes on a mission critical project go through a code review process. Functional changes are defined as those which contain edits to source code that ends up in the final software application. Test code, continuous integration pipelines, submodule pullups, and documentation are exempt from this review unless the project specifies otherwise. As part of the QA audit, compliance with this rule will be examined.

3.7. SYSTEM REQUIREMENTS & TRACING

All safety critical projects MUST define system requirements that give a high level overview of the system's intended behavior. These requirements shall:

1. Be written in the imperative (ex. "X **shall** do Y")
2. Be verifiable via some test procedure (or analysis if the requirement is structural)
3. Not be contradictory with any other requirement.

The system requirements should be written at the design phase and reviewed. Any edits to the requirements after this point shall be disclosed in the QA document. Every system requirement should map to a test procedure and a positive test result by the flight readiness review.

3.8. SYSTEM TESTING

The project must undergo testing before flight qualification. These tests should verify all of the system requirements.

3.9. INTEGRATION PERIODS

After the development phase, additions to the system must be carefully controlled. Any changes shall be limited to non-functional changes, submodule pull-ups, or bug fixes for issues discovered in system testing. The integration period should be a set, agreed upon length, and the release can only take place after that minimum period has elapsed.

4. QUALITY ASSURANCE MILESTONE PROCEDURES

At each milestone of development (end of design, end of development, end of testing), an independent quality assurance reviewer should evaluate the project using the following checklists. The reviewer should have an understanding of this quality assurance standard, with a preference for technical expertise in the subject area. The reviewer should also be independent of the project's development during that milestone.

If Avionics or the relevant QA authority for the organization has defined a specific Quality Assurance and/or Software Quality Assurance role, the reviewer should hold that role.

All of the following checklists will be given in markdown format. The QA reviewer shall paste the completed checklist in a top level review comment.

4.1. DESIGN MILESTONE CHECKLIST

```
Design Compliance Checklist

Applicable Standard: Avionics Software Quality Assurance
      Standard

Reviewer: <QA Reviewer's name>
Date: <Date of review>

- [ ] System Requirements are defined
- [ ] System Requirements are complete and verifiable
- [ ] System Requirements have been reviewed by all relevant
      stakeholders, including:
      - [ ] All component leads/sub-project leads
      - [ ] At least one customer/stakeholder
- [ ] Up-to-date system requirements have been checked in to
      document control.
- [ ] If the article is not defined as safety critical, a
      justification is provided.
- [ ] Quality Assurance Document completed and checked in to
      document control.

Design Assurance Checklist (for safety critical articles only)

- [ ] A functional hazard analysis is provided and checked into
      document control
- [ ] The functional hazard analysis provides mitigations for
      every failure condition that harm safety margins.
- [ ] The functional hazard analysis has been reviewed by all
      relevant stakeholders, including:
      - [ ] All component leads/sub-project leads
      - [ ] At least one customer/stakeholder
      - [ ] The club's safety officer or an authorized delegate
```

- [ ] The functional hazard analysis includes a security analysis OR a separate security analysis is provided.

4.2. TEST READINESS MILESTONE CHECKLIST

```
Test Readiness Compliance Checklist

Applicable Standard: Avionics Software Quality Assurance
      Standard

Reviewer: <QA Reviewer's name>
Date: <Date of review>

- [ ] If updates to design artifacts were required, the updates
      are disclosed in the quality control document's test
      readiness section.
- [ ] Separately controlled components have a quality assurance
      document in document control.
- [ ] Components with combined control are included in the
      system requirements and hazard analysis.
- [ ] A table of test procedures is provided. Each test
      procedure:
      - [ ] Has verifiable success criteria
      - [ ] Maps to one or more system requirements AND/OR
      mitigations in the functional hazard analysis
- [ ] A traceability report shows that all system requirements
      have an associated test procedure.
- [ ] A quality assurance sample (attached) shows compliance
      with development practices. If any part of the sample is
      non-compliant, the responsible parties have been alerted
      and remediation has been performed.
```

4.2.1. TEST READINESS QA AUDIT CHECKLIST

```
Test Readiness QA Audit Checklist

Applicable Standard: Avionics Software Quality Assurance
      Standard

Reviewer: <QA Reviewer's name>
Date: <Date of review>
Sample Size: <Sample Size (at least 5, ideally ~5% of commits)>

Per the Avionics SQA Standard, the minimum unit of work in
accordance with the QA standard is a commit. The sample chosen
for this audit shall consist of at least 5 commits, with at
least one commit coming from every project. If this is not the
initial release of a software project, the commits will be
sampled after the prior release baseline.

Sample #<n>:
Engineer: <Name of engineer>
Project: <Name of project repository>
Commit SHA: <SHA-256 hash of commit>
Commit Message: <Commit message>

- [ ] If functional changes (code file rather than documentation
      or CI) are present in this commit, the commit is contained
      in a pull request.
- [ ] If a code review was performed, one reviewer did not make
      any of the changes OR at least two reviewers independently
      cross-checked the other's changes.
- [ ] The engineer properly disclosed their AI usage and
      followed the pull request form.
- [ ] If any of the above were not marked true, this issue was
      remediated within one week by the engineer or the project
      lead.
```

4.3. FLIGHT QUALIFICATION CHECKLIST

```
Flight Qualification Compliance Checklist

Applicable Standard: Avionics Software Quality Assurance
      Standard

Reviewer: <QA Reviewer's name>
Date: <Date of review>

- [ ] If updates to design artifacts were required, the updates
      are disclosed in the quality control document's flight
      qualification section.
- [ ] If updates to development artifacts were required, the
      updates are disclosed in the quality control document's
      flight qualification section.
- [ ] All entries in the requirement tracing table have a
      positive or negative test result and the test runner's name
      is provided.
- [ ] The integration period has been enforced in accordance
      with the SQA standard, and no major features have been
      added since the test readiness review.
```

5. VERSION HISTORY

Version/Date	Author	Reviewers	Change History
1.0 August 24, 2026	Eli Sells	Max Kublanov (Safety Officer), Nicholas Armistead (Subject Matter Expert)	Initial Version