**Project Title**

**Software Description & NASA Software Engineering Classification**

**Introduction**

What motivated the creation of the software and how does this software address the problem.

**Applications and Scope**

Where will the software be used, and to what extent?

**Capabilities**

What can this software do, how has it improved the way work is performed and decisions are made.

**Interfaces**

What languages does it use, how do the users actually interface with the end product.

**Assumptions, Limitations, & Errors**

This is where limitations of the theory, model, science, etc should be briefly documented. If the tools only work for a specific scenario, say so.

**Additional Information**

**Software Classification & Justification**

This software is considered to be Class E software per NPR 7150.2.

This software is used to:

* + Perform minor desktop analysis of science or experimental data.

The software is not used to:

* + make decisions for an operational Class A, B, or C system or to-be built Class A, B, or C system
	+ support engineering development
	+ test other Class D software systems
	+ support mission planning or formulation
	+ operate a research, development, test, or evaluation laboratory
	+ provide decision support for non-mission critical situations
	+ in a Major Engineering/Research Facility
	+ perform research associated with airborne vehicles or systems

 The software will not:

* + operate, directly support, or be flight qualified to support an operational system
	+ be used in technical decision concerning an operational system
	+ directly affect primary or secondary mission objectives
	+ adversely affect the integrity of engineering/scientific artifacts
	+ have an impact on operational vehicles

 Additionally, if the software had anomalous behavior, that behavior would not cause or contribute to a failure of a system function:

* + resulting in a minor failure condition for the airborne vehicle
	+ with an effect on airborne vehicle operational capability or pilot workload

 When these criteria are no longer valid, categorization/classification will be reevaluated and the project will start following the procedures for the higher class.

**Not Safety Critical**: The software does not:

* Reside in a safety-critical system with at least one of the following being applicable to the software:
	+ Causes or contributes to a hazard
	+ Provide control or mitigation for hazards
	+ Controls safety-critical functions
	+ Processes safety-critical commands or data
	+ Detects and reports, or takes corrective action, if the system reaches a specific hazardous state
	+ Mitigates damage if a hazard occurs
	+ Resides on the same system (processor) as safety-critical software
	+ Process data or analyze trends that lead directly to safety decisions (e.g. determining when to turn power off to a wind tunnel to prevent system destruction)
	+ Provide full or partial verification or validation of safety-critical systems, including hardware or software systems