

JSC DATA REQUIREMENTS DESCRIPTION (DRD)
(Based on JSC – STD – 123)

1. DRD Title Software Assurance Plan	2. Version 09/25/2013	3. DRL No. 08	4. RFP/Contract No. NNJ13470997R/ NNJ14HA03B
5. Use (<i>Define need for, intended use of, and/or anticipated results of data</i>) Software assurance planning is used to document the software assurance activities to be performed during the life cycle phases.			6. DRD Category <input type="checkbox"/> Technical <input type="checkbox"/> Administrative <input checked="" type="checkbox"/> S&MA
6. References NPR 7150.2A, NASA Software Engineering Requirements JPR 7150.2, NASA Software Engineering Requirements NASA-STD-8739.9, Software Formal Inspections Standard NASA-STD-8719.13, Software Safety Standard NASA-STD-8739.8, Software Assurance Standard	8. Interrelationships (e.g. with other DRDs)		

A. SCOPE:

Software Assurance includes Software Quality Assurance, Software Quality Engineering, Software Safety, Software Reliability, Software Verification and Validation and Software Security Assurance. Software Assurance activities are conducted during the software development life cycle. The phases of the software development life cycle are:

- a) Concept and Initiation Phase
- b) Requirements Phase
- c) Design Phase
- d) Implementation Phase
- e) Integration and Test Phase
- f) Acceptance and Delivery Phase
- g) Operations/Maintenance Phase

Upon approval, the Software Assurance Plan will become a part of the contract as Attachment J-13.

B. CONTENT:

The contractor shall provide a Software Assurance Plan in accordance with JPR 7150.2, NASA-STD-8739.8 and NASA-STD-8719.13. The Software Assurance Plan shall identify the software assurance approval authority responsible for the establishment and composition of all software baselines and any changes to the baseline.

Software Assurance applies to all software developed for NASA, including:

- a) deliverable software,
- b) software included as part of deliverable hardware (including firmware),
- c) non-deliverable software (Commercially available or user-developed) used for development, testing, or acceptance of deliverable software or hardware (test and acceptance software; software design, test, and analysis tools; compilers, etc.)
- d) Commercially available (COTS), reused, or government-furnished software (GFS)
- e) Programmable Logic Devices (PLD) or Complex Electronics (CE). A programmable logic device or PLD is an electronic component used to implement user-defined functions into digital circuits. Unlike a

JSC DATA REQUIREMENTS DESCRIPTION (DRD)
(Based on JSC – STD – 123)

logic gate, which has a fixed function, a PLD has an undefined function at the time of manufacture. PLD functionality is typically described using a hardware description language (HDL), such as Verilog or VHDL, which is then converted, using PLD vendor-specific tools, into the hardware gate structure of the PLD to implement the function. PLDs can be implemented one-time or can be reconfigurable. While PLD functionality is generally described using an HDL, it can also be written in specialized variations of other programming languages. The functionality can range from a simple set of gates to a very complex set of gates (i.e. an embedded processor). The compilation of gates is hardware (regardless of the complexity) including the development of the embedded processor. For the purposes of this DRD, any software to be executed on a processor embedded within a PLD will be evaluated from a software perspective. The design and resulting hardware will be evaluated from a system perspective and is not the purview of this DRD.

C. FORMAT:

Contractor's format is acceptable but must include the content listed above. The reports shall be delivered in native format and be compatible with JSC standard software loads.

D. MAINTENANCE:

The plan shall be reviewed by the contractor annually and updated as needed or as directed by the contracting officer.

E. SUBMISSION AND DISTRIBUTION:

Submission and distribution shall be in accordance with Attachment J-01, Data Requirements List.