

3.2.1. Section 3.2.1 of the Management Plan shall include new development, modification, reuse, re-engineering, and all other activities resulting in software products. This section shall document the following items as implemented on this program:

- a. Program organizational structure, and authority and responsibility of each organizational unit responsible for software tasks, organizational units supporting software development, including subcontractor and other external organizations. Identification of software lead personnel (e.g., group, team, and product leads) and the lead personnel of organizational units directly supporting software development (e.g. quality, and configuration management). Skill and experience levels related to these lead positions.
- b. Constraints, restrictions, assumptions, inter-dependencies, and overarching requirements on the software development effort.
- c. Integrated Product Teams related to software, their charters, memberships, meeting frequency, rolls, and responsibilities.
- d. The software engineering (development) environment (for development, operation, or maintenance, as applicable), including development environments, test environments, libraries, equipment, facilities, standards, procedures, and tools. Programming languages, operating systems, integrated development environments, and use of non-deliverable software.
- e. Coding standards to be applied to the deliverable software products. If a uniform coding standards does not apply across all software products, explain the rationale, and describe the each coding standard being applied for each software product, including:
 1. Format (e.g. spacing, indentation, capitalization, order of information)
 2. Header information
 3. Embedded comments
 4. Naming conventions (e.g. variables, functions, parameters, procedures, files, etc.)
- f. Identify the deliverable software products, components, and databases as new, modified, reused, or reengineered. Describe the process and criteria used to determine reuse or re-engineering of software. Describe the criteria used to determine whether software is new, modified, reused, or reengineered. Describe the existing lifecycle support documentation associated with modified, reused, or reengineered software, and identify differences between that documentation and the documentation requirements for new software development. Identify the restrictions imposed on reused, modified, and reengineered software.
- g. Repositories, libraries, tools, and the processes used to gather, record, distribute, and manage the information and data used by the software developers to create the deliverable software products (e.g. software development folders/library). Structure, content, and control mechanisms of this information and data.
- h. Contractually deliverable software documentation (CDRLS), the entity (e.g. person, group, organizational unit) responsible for its preparation, the reviewer(s), and scheduled delivery dates. Identification and description of contractor internally generated

documents related to, or used to perform software development (e.g. unit test cases, procedures, IPT minutes, reports, trade studies, white papers, etc.)

- i. The software life cycle process or processes to be implemented to produce the deliverable software products. Describe the life cycle process activities, and artifacts; their phasing and timing, together with staffing, physical resources, software task complexity, and schedules. Describe the process, tasks, artifacts, and successful completion criteria for performing:
 1. Software Requirements Analysis
 2. Software Architectural Design
 3. Software Detailed Design
 4. Software Coding
 5. Software Unit Testing
 6. Software Integration Testing
 7. Hardware and Software Integration
- j. Management of software safety, software security, and other software critical requirements of the software products.
- k. Management of software product subcontractors, including involvement and interactions between the subcontractor and the prime contractor on this program.
- l. Configuration management of software, describing the methods used, activities performed, and artifacts generated to perform:
 1. Configuration identification
 2. Configuration control
 3. Configuration status accounting
 4. Configuration audits
- m. Release management (e.g. packaging, storage, handling, and delivery, etc.).
- n. Identification of those configuration items that will require formal configuration control, and when those items will be placed under formal configuration control. Process for the logging, analysis, and implementation of software change requests, software change control boards, the tracking of software changes, and the notification of concerned parties. Describe automated configuration management tools and their use on this program.
- o. Management of software quality, describing the standards, methodologies, procedures, physical resources, activities, artifacts generated, and tools involved in performing software quality assurance.
 1. Describe the types of software quality inspections performed, and the products to be inspected.
 2. Describe the content and use of quality records created from performing quality inspections.

3. Describe the timing and frequency of software quality inspections.
 4. Describe staffing, physical resources, inspection schedules, and responsibilities for conducting the quality assurance activities.
 5. Describe the process for the identification, collection, recording, maintenance, and disposition (resolution) of software quality records (e.g. inspection artifacts, and reports).
 6. Describe the rolls and activities performed in supporting software development, integration, verification, and validation.
 7. Describe the process for identifying, analyzing, managing, and resolving software problems.
 8. Describe the corrective action process used when quality inspections reveal issues.
 9. Describe the specific quality characteristics defined for this program.
 10. Describe the organizational reporting chain for quality control and the level of independence of the quality organization from product development.
- p. Software measures and metrics use shall be described, including collection and reporting frequency, methods for analyzing, and reporting. Describe the bounding criteria applied to metrics and the corrective action processes to be applied when the bounding criteria is violated.
 - q. Verification process used to confirm that each software product reflects the associated requirements (requirements baseline).
 - r. Validation process used to confirm that the requirements associated with the intended use of the software product are fulfilled.
 - s. Describe the process for performing software regression testing.
 - t. Describe the software build and release process and schedule. If multiple or iterative software builds will occur, describe the content or functionality contained in each build and the scheduling for each build.
 - u. Contractor and subcontractor involvement in formal programmatic and technical reviews related to software.
 - v. Software risk management process if different from the overall program risk management process.
 - w. Software security policy applied to this program, including rules for need-to-know and access-to-information.
 - x. Identification of data rights, warranty, ownership, and licensing associated with deliverable software products.
 - y. Means for scheduling, tracking, and reporting software tasks with status. Identify the entity (e.g. group, organization unit, person) responsible for the software tasks.
 - z. Specialized, additional, and ongoing training required for software development personnel.

- aa. Separate plans and other contractor documentation that provides the required content may be referenced within the management plan and attached as an appendix. Appendix references used in the management plan shall point to the specific section, subsection, page or paragraph providing the information.