DATA ITEM DESCRIPTION
Title: SOFTWARE BUILD PLAN

Number: DI-IPSC-82167
AMSC Number: N9873
DTIC Applicable: No
Preparing Activity: AS
Applicable Forms: N/A

Approval Date: 20171116
Limitation: N/A
GIDEP Applicable: No
Project Number: IPSC-2018-001

Use/relationship: The Software Build Plan consists of details regarding how the software functionality or capabilities will be built up over time to reach full capability in consonance with the system integration activities.

This Data Item Description contains the format, content, and intended use information for the data product resulting from the work task described by the contract.

Requirements:

1. Referenced documents. The applicable issue of the documents cited herein, including their approval dates and dates of any applicable amendments, notices, and revisions, shall be as specified in the contract.

2. Format. The Software Build Plan shall adhere to the format described under Content.

3. Content: The content shall contain the following:

3.1. Scope. This section shall be divided into the following paragraphs.
- 3.1.1. Identification. This paragraph shall contain a full identification of the system and the software to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).
- 3.1.2. Document overview. This paragraph shall summarize the purpose and contents of this document and shall describe any security or privacy considerations associated with its use.
- 3.1.3. Relationship to other plans. This paragraph shall describe the relationship, if any, to the Software Development Plan or to other project management plans.

3.2. Overview of required work. This section shall be divided into paragraphs as needed to establish the context for the planning described in later sections. It shall include, as applicable, an overview of:

- 3.2.1 Requirements and constraints of the system and software to be developed.
- 3.2.2 Requirements and constraints of the project documentation.
- 3.2.3 Position of the project in the lifecycle.
- 3.2.4 Requirements and constraints of the selected development strategy.

3.2.5 Requirements and constraints on project resources and schedules.
3.2.6 Other requirements and constraints, such as security, safety, methods, standards, interdependencies, etc.

3.3. Build Strategy**.** This section informs the reader what the high level plan, or methodology, is for the software build(s) and, most importantly, why the build is structured the way it is. The build is dependent on many factors which will change during execution and will require this plan to be actively updated and managed.

3.3.1. Factors: Each of the following subsections shall describe the factors required for this build, or builds, of software.
3.3.1.1. Software team construct by personnel to facilitate discovery, analysis and correction of problems.
3.3.1.2. Detail incremental capability buildup through the use of use cases and sequence diagrams used to describe functional mission and support threads (from basic infrastructure to full mission capability).
3.3.1.3. Interface evaluation as described by use cases and activity data flow.
3.3.1.4. Interaction and dependencies of the System/Software Integration Plan (SIP).
3.3.1.5. Interaction and dependencies with the ground/flight test plan of capability releases that can only be tested on a target platform.
3.3.1.6. Maturity definition(s) and criteria for a capability.
3.3.1.7. Defect burn-down strategy, if applicable.
3.3.1.8. Detail the development facility to include: compliers, debuggers, simulations, stimulations, hardware hosts, data collection, analysis tools and any specialized test equipment.
3.3.1.9. Test strategy to include any regression testing.

3.3.2. Build Processes. This section defines the processes and process execution used during release(s) of software. The follow subsections shall be described:
3.3.2.1. Build Planning
3.3.2.1.1. Linkage of the software builds in the Integrated Master Schedule (IMS) and other applicable program plans for all relevant activities (i.e. flight test, formal testing, etc.). Include expected software delivery dates for the integrators and test teams.
3.3.2.1.2. Build methodology.
3.3.2.1.3. Computer Software Component (CSC) testing with special focus on the completeness (via coverage analysis of requirements-based testing) and the testing of the software cybersecurity requirements:
3.3.2.1.3.1. Common Vulnerabilities and Exposure (CVE) List for the selection of security test tool(s).
3.3.2.1.3.2. Penetration testing.

3.3.2.1.3.3. Modified condition and decision coverage testing.
3.3.2.1.4. Identification of build artifacts.
3.3.2.1.5. Identification of required activities and/or resources.
3.3.2.1.6. Identification of project functions, components and subsystems to be built regardless of purchased or developed.
3.3.2.1.7. Identification of all external systems to be utilized or interfaced with.
3.3.2.1.8. Association of Build Release Indicators (i.e. the ability of the software to satisfy the functionality of each scheduled incremental build) to builds and also by capability.
3.3.2.1.9. Categorization of capabilities (i.e. functional, performance, problem correction, stress, failure recovery, baseline maturity, associated to system test and associated with flight test, as applicable).
3.3.2.1.10. Setup and establishment of configuration of the development and test environment(s).
3.3.2.1.11. Establishment of sequence and schedule for Build Release Indicators.
3.3.2.1.12. Meetings and agendas to facilitate execution.
3.3.2.1.13. Record keeping and reporting.

3.3.2.2. Build/Release Entrance and Exit Criteria

3.3.2.3. Build/Release Execution
3.3.2.3.1. Personnel management
3.3.2.3.2. Lab Utilization (Shift planning, prioritization and de-confliction)

3.3.2.4. Build/Release Metrics
3.3.2.4.1. Development Progress Profile (Percent of Work Complete).
3.3.2.4.2. Software Productivity.
3.3.2.4.3. Software Requirements Volatility.
3.3.2.4.4. Software Defects.
3.3.2.4.5. Software Size.
3.3.2.4.6. Build Release Indicators.
3.3.2.4.7. Test Progress Indicators (The progress made against initial plans for each formal build and tracked against test efforts leading toward the build for formal and/or qualification test).

3.3.2.5. Problem Report Management. Include Configuration Control Board (CCB) participants and participation.

3.3.2.6. Testing. This section should include:
3.3.2.6.1. Scope determination.

3.3.2.6.2. Testing strategy assure that no changes have affected previously successfully tested functionality.

3.3.2.6.3. Unit, component and CSCI level test procedures to be used.

3.4. Notes. This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

3.5. Appendices. Appendices may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendices may be bound as separate documents for ease in handling. Appendices shall be lettered alphabetically (A, B, etc.).

End of DI-IPSC-82167