

DATA ITEM DESCRIPTION

Title: Software Resources Data Reporting: Final Developer Report and Data Dictionary

Number: DI-MGMT-81740

Approval Date: 20070420

AMSC Number: D7725

Limitation:

DTIC Number:

GIDEP Applicable:

Office of Primary Responsibility: (D)OSD/PA&E/CAIG

Applicable Forms: Software Resources Data Reporting: Final Developer Report (Sample Format 3)

Use/Relationship: This Data Item Description (DID) contains information about the sample format, content, and intended use for the data deliverable resulting from the task in the statement of work. This data deliverable consists of two parts. The first part, the Final Developer Report, is used to obtain the actual (as-built) characteristics of a software product and its development process. The second part is the Software Resources Data Reporting (SRDR) Data Dictionary, which defines each of the data elements within the Software Resources Data (SRD) report and describes the methods and rules used to perform the data measurement. Every submission of this SRD report shall contain both the SRDR Final Developer Report and an associated SRDR Data Dictionary.

The SRD report is not a management or software metrics report. It is not intended for tracking progress of the development during contract execution, nor is it intended to collect financial information. It does, however, collect the person-hours expended during software development.

The intent of the SRDR process is to collect objective measurable data commonly used by industry and DoD cost analysts. These data are used to compile a repository of actual software product sizes, schedules, effort, and quality that Government analysts can draw upon to build credible size, cost, and schedule estimates of future software-intensive systems.

Information to be acquired through these data will include descriptive information about the product and developer and actual as-developed software product size, development schedule, peak staff, and direct labor hours incurred.

The contractor must provide a SRDR Data Dictionary that defines the data elements contained in the negotiated SRDR Final Developer Report. The definitions of the data items are negotiable but must include the following categories of data: Context, Project Description, Size, Effort, and Schedule. Optionally, data in the category of Quality may be provided.

The minimum level of detail to be reported in each SRDR submission shall be in accordance with the contract's Cost and Software Data Reporting (CSDR) Plan, DD Form 2794, as approved by the Office of the Secretary of Defense (OSD) Cost Analysis Improvement Group (CAIG) Chair. Discrete reporting is required for each Work Breakdown Structure (WBS) element identified in Box 13 of the CSDR Plan.

An SRDR submission shall be prepared in a Microsoft Excel-compatible electronic file format. For submissions that require discrete reporting of multiple WBS elements, the data shall be prepared and integrated into one electronic file.

The SRDR Data Dictionary shall be prepared in a readable electronic (digital) file format such as Microsoft Excel or Microsoft Word (e.g., pdf files are not acceptable).

Requirements:

1. Reference Documents. DoDI 5000.2, “Operation of the Defense Acquisition System,” establishes mandatory policies for requiring SRD reports. DoD 5000.04–M–1, “Cost and Software Data Reporting (CSDR) Manual,” prescribes procedures and instructions for stakeholders in the SRDR process. Detailed instructions for preparing the SRDR Final Developer Report and SRDR Data Dictionary are provided below.

2. Format. There is no prescribed data format for either the SRDR Final Developer Report or the SRDR Data Dictionary. The SRDR Final Developer Report shall be in a format agreed to by the contractor and the Government. Software Resources Data Report: Final Developer Report (Sample Format 3), shown in Figure 1, serves as a starting point for developing a tailored report.

3. Content. The SRDR Final Developer Report shall contain actual, as-built software measurement data as described in the contractor’s SRDR Data Dictionary. The data shall reflect scope relevant to the reporting event. SRDR submissions for contract complete event shall reflect the entire software development project. SRDR submissions for completion of a product build shall reflect size, effort, and schedule of that product build. The SRD report shall contain mandatory data elements as outlined below and reflected in Sample Format 3 (Figure 1). Data elements reported beyond those outlined in this DID shall be agreed upon by the Cost Working-group Integrated Product Team (CWIPT) and approved by the OSD CAIG Chair.

3.1. Report Context and Development Organization.

3.1.1. Security Classification. The top and bottom of every page shall be marked with the security classification of the report which typically will be “Unclassified”. However, if the appropriate security classification based on the classification level of the data reported is classified, contact the DCARC for special processing instructions.

3.1.2. System/Element Name. The name of the system or element identified for reporting on the contract’s CSDR Plan. The name shall include any applicable version, release, build, or other identifier. The System/Element name shall also reference the name of the WBS element and its associated WBS number from the contract’s CSDR Plan.

3.1.3. CSDR Plan Number. Provide the reference number of the OSD CAIG Chair-approved CSDR Plan associated with the SRDR submission.

3.1.4. Report As-of Date. Provide the date of the data, which is not necessarily the date the report was prepared.

3.1.5. Authorizing Vehicle. This is the prime contract number (if applicable) and amendment number (if applicable), or reference to a memorandum of

understanding or other documentation that authorizes the development of the subject software. A reporting subcontractor shall use the prime contractor's authorizing vehicle.

3.1.6. Reporting Event. Identify the event for which this SRDR submission is being prepared. This reporting event must correspond to a specific reporting event identified in Block 14 of the contract's CSDR Plan. Examples include contract end, increment 1 end, build 2 complete, spiral 3 complete, and so on.

3.1.7. Submission Number. SRD reports are submitted based on reporting events identified in Block 14 of the contractor's CSDR Plan. For each SRDR reporting event, enter "1" the first time an SRD report is submitted, enter "2" if the contractor submits a corrected/revised SRD report, and so on.

3.1.8. Development Organization. Enter the name of the company or organization responsible for development of the software product. The associated SRDR Data Dictionary must be used to explain the mapping of development organizations, software components, and SRD reports submitted.

3.1.9. Software Process Maturity. The SRD report shall report the characterization of the developer's software process maturity using a methodology such as the Software Engineering Institute (SEI) software Capability Maturity Model (CMM), the Capability Maturity Model Integration (CMMI)-SW, or an alternative equivalent rating. The reported software process maturity shall reflect the rating that the primary development organization has formally certified as of the date of the reporting event. Identify the name of the person that performed the assessment, the evaluator's affiliation, and the date of certification. If no formal certification has been conducted, leave these items blank. If a single submission is used to represent the work of multiple organizations, enter the level of the organization that will be expending the most effort on the development project (not necessarily the prime contractor) and note this in the associated SRDR Data Dictionary. If the Government has accepted an alternative assessment mechanism, such as the Air Force's Software Development Capability Evaluation (SDCE) or ISO-15504, enter those results and explain the meaning of the assessment in the SRDR Data Dictionary.

3.1.10. Precedents. List up to five analogous systems developed by the same software organization or development team.

3.1.11. SRDR Data Dictionary Filename. If the SRDR Data Dictionary is stored as a separate electronic file from the SRD report, provide the filename of the SRDR Data Dictionary file. Provide the date the associated SRDR Data Dictionary was last revised.

3.1.12. Comments. Provide any comments about report context and development organization. Include more detailed explanations in the associated SRDR Data Dictionary.

3.2. Product and Development Description.

3.2.1. Functional Description. For each element reported, provide a brief description of its function. What is it? What does it do?

3.2.2. Software Development Characterization. In general language, provide a brief description for each element reported that characterizes the software development work undertaken on that element. Examples might include completely new from-scratch development, rehosting of software to different processor/operating system, reengineering of legacy code into open architecture, translation of legacy code from Ada to C, and so on.

3.2.3. Application Type. Identify at least one application type (i.e., the end-user mission) developed using one or more domain names from those listed in Figure 2. A minimum of one primary application type shall be identified, but any number of application types may be listed. If none of the examples in Figure 2 are appropriate, enter a phrase to describe the application type and define it in the associated SRDR Data Dictionary. When internal development efforts within a program are large and independent, respondents may choose to report each using a separate SRD report instead of as various application types within a single report. For every application type reported provide:

3.2.3.1. Primary and Secondary Programming Language. Enter the primary and secondary computer language in which most of the development was conducted. This can be a compiled language, such as FORTRAN, Ada, or C, an interpreted language such as Basic, or a graphical or model-based language, such as Rhapsody/UML or Simulink. Use the amount of effort spent in development to determine the primary language rather than the amount of function delivered. Explain any interpretation of this item in the associated SRDR Data Dictionary.

3.2.3.2. Percentage of Overall Product Size. Enter the approximate percentage (up to 100%) of the product size that is of this application type. If relevant and appropriate, the contractor can include the integrated Commercial Off-the-Shelf (COTS)/Government Off-the-Shelf (GOTS) packages in this calculation. If so, an explanation must be placed in the SRDR Data Dictionary.

3.2.3.3. Actual Development Process. Enter the name of the development process followed for the development of the system. Do not indicate a software architecture method (such as object-oriented development) or a development tool (such as Rational Rose), as these do not specify a process. Typical types of development processes adopted include waterfall, spiral, or RAD. If the contractor uses an atypical internal process was used, provide a description of the development process in the SRDR Data Dictionary.

3.2.3.4. Upgrade or New Development? Indicate whether the primary development was new software or an upgrade. A software system is considered new either if no existing system currently performs its function or if the development completely replaces an existing system. A software system that replaces part of an existing system (such as the replacement of a database) should be considered an upgrade. An existing software system that was ported to a new platform or reengineered to execute as a Web or distributed application (for example) would be considered an upgrade unless it

was also completely redeveloped from scratch (new requirements, architecture, design, process, code, etc.).

3.2.3.5. Software Development Method(s). Identify the software development method or methods used to design and develop the software product (e.g., Structured Analysis, Object Oriented, Vienna Development Method, etc.).

3.2.4. Non-Developmental Software.

3.2.4.1. COTS/GOTS Applications Used. List the names of the applications or products that constitute part of the final delivered product, whether they are COTS, GOTS, or open-source products. If a proprietary application or product that is not generally commercially available will be included, identify it here and include any necessary explanation in the associated SRDR Data Dictionary.

3.2.4.2. Integration Effort (Optional). If requested by the CWIPT, the SRD report shall contain the actual effort required to integrate each COTS/GOTS application identified in Section 3.2.4.1. “Effort” may be expressed in terms of staff-hours, new/modified glue code, or a qualitative assessment of effort required (i.e., low, medium, high, etc.). The SRDR Data Dictionary shall contain appropriate definitions of the integration effort metric chosen by the contractor.

3.2.5. Staffing.

3.2.5.1. Peak Staff. For the element reported, enter the actual peak team size, measured in full-time equivalent (FTE) staff. Include only direct labor in this calculation unless otherwise explained in the associated SRDR Data Dictionary. The SRDR Data Dictionary shall include a definition of FTE that includes the hours per staff-month used to compute FTE.

3.2.5.2. Peak Staff Date. Enter the date when the actual peak staffing occurred.

3.2.5.3. Hours per Staff-Month. Enter the number of direct labor hours per staff-month. Indicate in the SRDR Data Dictionary whether the reported hours per staff-month reflect an accounting standard or a computation. If they reflect a computation, provide details on how the computation was performed.

3.2.6. Personnel Experience in Domain. Stratify the project staff domain experience by experience level and specify the percentage of project staff at each experience level identified. (Sample Format 1 identifies three levels: Highly Experienced, Nominally Experienced, and Inexperienced/Entry Level.) Provide a definition for each experience level (i.e., the number of years of experience) in the SRDR Data Dictionary. Also provide a definition of domain experience in the SRDR Data Dictionary (e.g., “Domain experience is defined as the number of years a project staff member has worked within a mission discipline such as real time fire control radar or missile guidance and tracking”). Additionally, the contractor is permitted to tailor the type of experience reported to track to

whatever type of experience is most applicable (e.g., Ada programming experience, total software development experience, etc.).

3.2.7. Comments. Provide any comments about the product and development description. Include more detailed explanations in the associated SRDR Data Dictionary.

3.3. Product Size Reporting.

3.3.1. Number of Software Requirements. Provide the actual number of software requirements. The method of counting actual number of requirements implemented by the development software must be the same as that used for counting estimated requirements (as reported in the SRDR Initial Development Report). Do not count requirements concerning external interfaces not under project control (see next item, "Total Requirements"). Alternative requirements counts based on Use Cases are also permitted. The SRDR Data Dictionary shall provide both a definition of what types of requirements are included in the count (i.e., functional, security, safety, other derived requirements, etc.) and the units (e.g., "shalls," "sections," paragraphs, etc.) and counting methods used.

3.3.1.1. Total Requirements. Enter the actual number of total requirements satisfied by the developed software product at the completion of the increment or project. This count must be consistent with the total size of the delivered software (i.e., it must not solely focus on new development, but must reflect the entire software product).

3.3.1.2. New Requirements. Of the total actual number of requirements reported, the SRD report shall identify how many are new requirements.

3.3.2. Number of External Interface Requirements. Provide the number of external interface requirements, as specified below, not under project control that the developed system satisfies. External interfaces include interfaces to computer systems, databases, files, or hardware devices with which the developed system must interact but which are defined externally to the subject system. If the developed system interfaces with an external system in multiple ways (such as for reading data and also for writing data), then each unique requirement for interaction should be counted as an interface requirement. Provide the actual number of interface requirements handled by the developed software using the same counting method as was used in the estimating (initial) reports. Explain any details about the counting methods for external interface requirements in the SRDR Data Dictionary.

3.3.2.1. Total External Interface Requirements. Enter the actual number of total external interface requirements satisfied by the developed software product at the completion of the increment or project. This count must be consistent with the total size of the delivered software (i.e., it must not solely focus on new development, but must reflect the entire software product).

3.3.2.2. New External Interface Requirements. Of the total number of external interface requirements reported, the SRD report shall identify how many are new external interface requirements.

3.3.3. Requirements Volatility. Indicate the amount of requirements volatility encountered during development using a qualitative scale (very low, low, nominal, high, very high) relative to similar systems of the same type. This should be a relative measure rather than an absolute one in order to understand the expectations of the impact of requirements volatility during the course of the software development. The contractor's specific definitions for each rank in the qualitative scale and overall definition of what constitutes requirements volatility shall be provided in the SRDR Data Dictionary.

3.3.4. Software Size.

3.3.4.1. Delivered Size. The SRD report shall capture the delivered size of the product developed, not including any code that was needed to assist development but was not delivered (such as temporary stubs, test scaffoldings, or debug statements). Additionally, the code shall be partitioned (exhaustive with no overlaps) into appropriate development categories. A common set of software development categories is new, reused with modification, reused without modification, and generated code. When code is included that was reused with modification or reused without modification, the contractor shall provide, in the SRDR Data Dictionary, an assessment of the amount of redesign, recode, and retest required to implement the modified or reused code. Code reused without modification may be further partitioned into reuse from a previous increment of this project (that was previously reported on an SRD report associated with that prior increment) or reused from a source external to the project. In all cases, the partitioning used for reporting shall be customized to conform to the contractor's standard internal reporting of software development categories as long as the partitioning does not double count or omit any delivered software.

3.3.4.1.1. Carryover Code. Do not count the same code as new in more than one SRDR incremental report. Except for the first increment SRD report or the overall project SRD report at contract completion, an SRD report shall distinguish between code developed in previous increments that is carried forward into the current increment and code added as part of the effort on the current increment. Examples of such carried-forward code include code developed in Spiral 1 that is included in Spiral 2 or code that is developed for Version 3.0 software that is included in Version 3.1 software. Table 1 provides one possible example of reporting code from previous builds for the Initial Developer Reports associated with builds.

Table 1: Example of Reporting Carryover Code from Previous Builds

		Build 1 Complete	Build 2 Complete	Build 3 Complete	Contract Complete
New Code	Human Generated	1,000	0	2,500	3,500
	Auto Generated	0	500	2,500	3,000
External Reused	With Modification	5,000	15,000	500	20,500
	Without Modification	3,000	0	2,000	5,000
Carryover Code from Previous Build	With Modification	0	0	12,250	N/A
	Without Modification	0	9,000	12,250	N/A
Total Delivered Code		9,000	24,500	32,000	32,000

3.3.4.1.2. Auto-generated Code. If the developed software contains auto-generated source code, the SRD report shall include an auto-generated code sizing partition as part of the set of development categories.

3.3.4.1.3. Subcontractor-Developed Code. The categories of delivered code in the SRD report shall be further partitioned by responsible developer, for example: Prime Contractor Only and All Other Subcontractors. If the subcontractor-developed code cannot be further partitioned, then report only total delivered code for the subcontractors. If the delivered size of one or more subcontractors is unknown, annotate in the SRD report form and provide additional explanation in the SRDR Data Dictionary.

3.3.4.2. Counting Convention. Identify the counting convention used to count software size. The specific definition must be provided in the SRDR Data Dictionary. While Source Lines of Code (SLOC) is a prominent unit of software size, the SRD report shall reflect units of measure in use internally to the contractor. Alternative units, such as function points, are permissible units of measure to report so long as the contractor consistently reports this on both the Initial Developer Report and the Final Developer Report. Units of measure that reflect weighted sum normalization of size into equivalent units, such as Equivalent New Lines of Code, shall not be used as a primary sizing unit of measure in the SRD report. This information may be reported in the SRD report's supplemental information.

3.3.4.3. Size Reporting by Programming Language (Optional). The SRD report shall partition software size by programming language if requested by the CWIPT.

3.3.4.4. Standardized Code Counting (Optional). If requested by the CWIPT, the contractor shall use a publicly available and documented code counting tool, such as the University of Southern California Code Count tool, to obtain a set of standardized code counts that reflect logical size. These results shall be used to report software sizing in the SRD report.

3.3.5. Comments. Provide any comments about product size reporting. Include more detailed explanations in the associated SRDR Data Dictionary.

3.4. Resource and Schedule Reporting. The Final Developer Report shall contain actual schedules and actual total effort for each software development activity.

3.4.1. Effort. The units of measure for software development effort shall be reported in staff-hours. Effort shall be partitioned into discrete software development activities as defined by the contractor's standard software development process. The following activities are taken from the activity definitions used in standard ISO 12207 and are intended as an example of partitioning software development effort. Additional software support activities are also shown.

- software requirements analysis,
- software architecture and detailed design,
- software coding and unit testing,
- software integration and system/software integration,
- software qualification testing,
- software developmental test and evaluation, and
- other software support activities:
 - software quality assurance,
 - software configuration management,
 - software program management,
 - data,
 - software process improvement,
 - IV and V, and
 - problem resolution.

3.4.2. WBS Mapping. For each software development activity reported in the Final Developer Report, identify, from the contractor's OSD CAIG Chair-approved CSDR plan, the contract WBS name(s) and WBS number(s) that capture that software development activity. Do not reference internal contractor cost account codes elements.

3.4.3. Subcontractor Development Effort. The effort data in the SRD report shall be separated into a minimum of two discrete categories and reported separately: Prime Contractor Only and All Other Subcontractors. The prime contractor shall report the subcontractor's actual effort, if available. If the subcontractor's actual effort data are not available, then the prime contractor shall estimate the subcontractor effort and clearly annotate that the sub-contractor effort reflects and estimate. If the reported subcontractor development effort cannot be partitioned by software development activity, then report only the total effort expended and provide a definition in the SRDR Data Dictionary that explains what software development activities are included in the subcontractor-reported development effort.

3.4.4. Schedule. For each software development activity reported, provide the actual start and end dates for that activity. Alternatively, month numbers, starting with month "1" at the time of Contract Award, can be used. If there were multiple start and stop dates for the same activity, as would be the case for iterative or spiral development, then report the earliest and latest end date for each activity, to the extent that is sensible for the approach used. If month numbers are used, provide the date that is equivalent to month "1" either as a comment in the SRDR Initial Developer Report or in the SRDR Data Dictionary.

3.4.5. Comments. Provide any comments about resource and schedule reporting. Include more detailed explanations in the associated SRDR Data Dictionary.

3.5. Product Quality Reporting (Optional). Quality should be quantified operationally (through failure rate and defect discovery rate). However, other methods may be used if appropriately explained in the associated SRDR Data Dictionary.

3.5.1. Required or Actual Mean Time to Serious or Critical Defect (MTTD) at Delivery. MTTD at time of delivery is one method by which a customer can specify nominal product quality. The definition of this measure must include whether minor or only major (mission compromising) defects are counted, and whether recurring known defects or only new ones are counted. Also, the operational time basis must be clarified, such as by indicating whether a system is operational only eight hours a day or continuously, or whether a system operates in a single instance or in multiple instances at different locations simultaneously. Use the associated SRDR Data Dictionary to clarify the counting method. Developers may customize these definitions to conform to their existing definitions. Developers should use existing procedures for distinguishing defects from routine development changes, such as problems found after an inspection, after a configuration control baseline, or after advancement to the next state of a development process.

3.5.2. Observed or Computed Reliability Compared With Nominal Reliability of Analogous Systems. An alternative method to reporting nominal quality is to compare the required or actual reliability of the system being reported on with the typical reliability for systems of this type. For example, if the system is an operational flight program (OFP), higher than nominal reliability might be

expected for the OFP of a fly-by-wire aircraft. On the other hand, if the OFP were to control a pilotless vehicle, such as a surveillance or drone aircraft, the required reliability might be lower than average for OFP systems. A customization of this item could allow the response to be in terms relative to other similar systems; for example, a scale such as “much higher,” “somewhat higher,” “nominal,” “lower,” or “much lower” might be appropriate. As with any customization, the explanation of the data must be included in the SRDR Data Dictionary.

3.5.3. Comments. Provide any comments about the product quality. Include more detailed explanations in the associated SRDR Data Dictionary.

3.6. Point of Contact (POC) Information. Enter the following information for the person to be contacted for answers to any questions about this report, the data reported, or the associated SRDR Data Dictionary:

- name: last name, first name, and middle initial;
- department name;
- telephone number, including area code;
- e-mail address;
- fax number, including area code;
- signature(an electronic signature is acceptable); and
- date signed (usually later than the “as of” date).

3.7. SRDR Data Dictionary. The SRDR Data Dictionary shall contain, at a minimum, the following information in addition to the specific requirements identified in Sections 3.1 through 3.6:

3.7.1. Experience Levels. Provide the contractor’s specific definition (i.e., the number of years of experience) for personnel experience levels reported in the SRD report.

3.7.2. Software Size Definitions. Provide the contractor’s specific internal rules used to count software code size. The Software Engineering Institute (SEI) technical report, “Software Size Measurement – A Framework for Counting Source Statements,” has sample checklists for physical and logical code counting; however, the contractor may use any checklist. The rules must address (a) what is counted (i.e., terminal semi-colons, non-comment non-blank physical lines, etc.), and (b) what is included in the logical size count (i.e., job control language, included files, comments, etc.).

3.7.3. Software Size Categories. For each software size category identified (i.e., New, Modified, Unmodified, etc.), provide the contractor’s specific rules and/or tools used for classifying code into each category.

3.7.4. Peak Staffing. Provide a definition that describes what activities were included in peak staffing.

3.7.5. Requirements Count (Internal). Provide the contractor's specific rules and/or tools used to count requirements. The definition must also identify the source document used for tallying requirements (i.e., system/subsystem design description, software specification document, etc.).

3.7.6. Requirements Count (External). Provide the contractor's specific rules and/or tools used to count external interface requirements. The definition must also identify the source document used for tallying requirements (i.e., SV-6, ICD, etc.).

3.7.7. Requirements Volatility. Provide the contractor's internal definitions used for classifying requirements volatility.

3.7.8. Software Development Activities. Provide the contractor's internal definitions of labor categories and activities included in the SRD report's software activity. This definition should not focus on a textbook software engineering definition, but should rather focus on the natural manner in which the contractor describes the kinds of efforts that are included in the software development activity.

3.7.9. Comments. Provide any additional information that would permit a DoD cost analyst to correctly interpret the contractor's data.

Figure 1. Software Resources Data Report: Final Developer Report
(Sample Format 3), Page 1

Section 3.1.1 SECURITY CLASSIFICATION						
SOFTWARE RESOURCES DATA REPORTING: FINAL DEVELOPER REPORT (SAMPLE FORMAT 3) <i>Due 60 days after final software delivery and 60 days after delivery of any release or build.</i>						
Section 3.1: REPORT CONTEXT AND DEVELOPMENT ORGANIZATION						
SYSTEM/ELEMENT NAME Section 3.1.2				REPORT AS OF Section 3.1.4		
REPORTING EVENT Section 3.1.6				CSDR PLAN # Section 3.1.3		
AUTHORIZING CONTRACT VEHICLE Section 3.1.5			SUBMISSION # Section 3.1.7	SUPERSEDES # (if applicable) Section 3.1.7		
DEVELOPMENT ORGANIZATION Section 3.1.8		SOFTWARE PROCESS MATURITY Section 3.1.9	LEAD EVALUATOR Section 3.1.9			
		CERTIFICATION DATE Section 3.1.9	EVALUATOR AFFILIATION Section 3.1.9			
PRECEDENTS (List up to five similar systems by the same organization or team.) Section 3.1.10						
SRDR DATA DICTIONARY FILENAME Section 3.1.11				LAST REVISION DATE Section 3.1.11		
COMMENTS Section 3.1.12						
Section 3.2 PRODUCT AND DEVELOPMENT DESCRIPTION						
FUNCTIONAL DESCRIPTION Section 3.2.1						
SOFTWARE DEVELOPMENT CHARACTERIZATION Section 3.2.2						
APPLICATION TYPE	PRIMARY PROGRAMMING LANGUAGE	SECONDARY PROGRAMMING LANGUAGE	PERCENT OF PRODUCT SIZE	ACTUAL DEVELOPMENT PROCESS	SW DEVELOPMENT METHOD(S)	UPGRADE OR NEW?
Section 3.2.3	Section 3.2.3.1	Section 3.2.3.1	Section 3.2.3.2 %	Section 3.2.3.3	Section 3.2.3.5	Section 3.2.3.4
			%			
SECTION 3.2.4 COTS/GOTS APPLICATIONS USED:						
NAME		INTEGRATION EFFORT (OPTIONAL)	NAME		INTEGRATION EFFORT (OPTIONAL)	
Section 3.2.4.1		Section 3.2.4.2	Section 3.2.4.1		Section 3.2.4.2	
Section 3.2.4.1		Section 3.2.4.2	Section 3.2.4.1		Section 3.2.4.2	
Section 3.2.5 STAFFING						
PEAK STAFF (Maximum Team Size in FTE) Section 3.2.5.1			PEAK STAFF DATE Section 3.2.5.2	HOURS/STAFF-MONTH Section 3.2.5.3		
PERSONNEL EXPERIENCE IN DOMAIN						
HIGHLY EXPERIENCED: Section 3.2.6 %	NOMINALLY EXPERIENCED: Section 3.2.6 %		INEXPERIENCED/ENTRY LEVEL: Section 3.2.6 %			
COMMENTS Section 3.2.7						
Section 3.3 ACTUAL PRODUCT SIZE REPORTING						
NUMBER OF SOFTWARE REQUIREMENTS	Total Section 3.3.1.1	NUMBER OF EXTERNAL INTERFACE REQUIREMENTS		Total Section 3.3.2.1	REQUIREMENTS VOLATILITY Section 3.3.3	
	New Section 3.3.1.2			New Section 3.3.2.2		
Section 3.3.4 FINAL TOTAL DELIVERED CODE			COUNTING CONVENTION	PRIME CONTRACTOR ONLY	ALL OTHER SUBCONTRACTORS	
Section 3.3.4.1 AMOUNT OF DELIVERED CODE DEVELOPED NEW			Section 3.3.4.2 HUMAN GENERATED	Section 3.3.4.1	Section 3.3.4.1.3	
			Section 3.3.4.2 AUTO GENERATED	Section 3.3.4.1.2	Section 3.3.4.1.3	
Section 3.3.4.1 AMOUNT OF DELIVERED CODE REUSED FROM EXTERNAL SOURCE (i.e. NOT INHERITED FROM PREVIOUS INCREMENT/BUILD OR PREDECESSOR)			Section 3.3.4.2 WITH MODIFICATIONS	Section 3.3.4.1	Section 3.3.4.1.3	
			Section 3.3.4.2 WITHOUT MODIFICATIONS	Section 3.3.4.1	Section 3.3.4.1.3	
Section 3.3.4.1 AMOUNT OF DELIVERED CODE INHERITED (i.e. REUSED FROM PREVIOUS INCREMENT/BUILD OR PREDECESSOR)			Section 3.3.4.2 WITH MODIFICATIONS	3.3.4.1.1	Section 3.3.4.1.3	
			Section 3.3.4.2 WITHOUT MODIFICATIONS	3.3.4.1.1	Section 3.3.4.1.3	
COMMENTS Section 3.3.5						

Figure 1. Software Resources Data Report: Final Developer Report
(Sample Format 3), Page 2

Section 3.1.1

SECURITY CLASSIFICATION

SOFTWARE RESOURCES DATA REPORTING: FINAL DEVELOPER REPORT (SAMPLE FORMAT 3)					
SECTION 3.4 ACTUAL RESOURCE AND SCHEDULE REPORTING					
SOFTWARE ACTIVITY NAME	MAPS TO CSDR WBS NUMBER(S)	START MONTH	END MONTH	TOTAL HOURS PRIME CONTRACTOR ONLY	TOTAL HOURS ALL OTHER SUBCONTRACTORS
Section 3.4.1 (Example: SOFTWARE REQUIREMENTS ANALYSIS)	Section 3.4.2	Section 3.4.4	Section 3.4.4	Section 3.4.1	Section 3.4.3
Section 3.4.1 (Example: SOFTWARE ARCHITECTURE AND DETAILED DESIGN)	Section 3.4.2	Section 3.4.4	Section 3.4.4	Section 3.4.1	Section 3.4.3
Section 3.4.1 (Example: SOFTWARE CODING AND UNIT TESTING)	Section 3.4.2	Section 3.4.4	Section 3.4.4	Section 3.4.1	Section 3.4.3
Section 3.4.1 (Example: SOFTWARE INTEGRATION AND SYSTEM/SOFTWARE INTEGRATION)	Section 3.4.2	Section 3.4.4	Section 3.4.4	Section 3.4.1	Section 3.4.3
Section 3.4.1 (Example: SOFTWARE QUALIFICATION TESTING)	Section 3.4.2	Section 3.4.4	Section 3.4.4	Section 3.4.1	Section 3.4.3
Section 3.4.1 (Example: SOFTWARE DEVELOPMENTAL TEST AND EVALUATION)	Section 3.4.2	Section 3.4.4	Section 3.4.4	Section 3.4.1	Section 3.4.3
ALL OTHER DIRECT SOFTWARE ENGINEERING DEVELOPMENT EFFORT Section 3.4.2 (Example: SOFTWARE PROGRAM MGT, SOFTWARE QUALITY ASSURANCE, SW CM)	Section 3.4.2			Section 3.4.1	Section 3.4.3
TOTAL SOFTWARE DEVELOPMENT EFFORT				Section 3.4.1	Section 3.4.3
COMMENTS					
Section 3.4.5					
Section 3.5 PRODUCT QUALITY REPORTING (OPTIONAL)					
REQUIRED OR ACTUAL MEAN TIME TO SERIOUS OR CRITICAL DEFECT (MTTD) AT DELIVERY IN HOURS (Provide the specific definition of this measure in the associated Data Dictionary.)					
Section 3.5.1					
OBSERVED OR COMPUTED RELIABILITY COMPARED WITH NOMINAL RELIABILITY OF ANALOGOUS SYSTEMS (Provide details about the analogous systems and define nominal reliability in the associated Data Dictionary.)					
Section 3.5.2					
COMMENTS					
Section 3.5.3					
Section 3.6 POINT OF CONTACT (POC) INFORMATION					
NAME (Last, First, Middle Initial)		DEPARTMENT		TELEPHONE NO. (Include Area Code)	
Section 3.6		Section 3.6		Section 3.6	
E-MAIL ADDRESS	FAX NO. (Include Area Code)	SIGNATURE		DATE SIGNED (YYYY-MM-DD)	
Section 3.6	Section 3.6	Section 3.6		Section 3.6	

Section 3.1.1
SECURITY CLASSIFICATION

Figure 2. Application Types

<p><u>Warfare Mission Areas</u> Antiair Warfare Antisubmarine Warfare Naval Antisurface Ship Warfare Amphibious Warfare Chemical Warfare Biological and Radiological Defense Land Warfare Special Warfare Strategic Warfare Tactical Air Warfare Electronic Warfare Strategic Defense Initiative</p> <p><u>Mobility Mission Areas</u> Air Mobility Land Mobility Sea-Surface Mobility Undersea Mobility Space Mobility</p> <p><u>Communications, Command and Control/Intelligence Mission Areas</u> Communications, Command and Control Intelligence, Including Reconnaissance</p> <p><u>Mine and Obstacle Mission Areas</u> Land Mine/Obstacle/Countermeasures Sea Mine/Countermine</p> <p><u>Mission and System Support Mission Areas</u> Logistics Manpower, Personnel and Training Mission/System Support</p> <p><u>Weapon Systems Functions</u> Target Acquisition/Search/Detect Threat Evaluation Target Tracking Weapon Assignment Fire Control Acquisition and Designation Launch Propulsion Control Flight Controls Conventional Munitions/Weapons Directed Energy Weapons Hard Target Kill/Anti-Armor Fuzing Chemical Warfare (Offense)</p>	<p><u>Defensive Systems Functions</u> Hit Avoidance Signature Control/Suppression Reduction Armor, Infantry and Crew Protection EMP Hardening/Survivability from Nuclear Weapons Damage Control Chemical/Biological Defense Deterrence</p> <p><u>Mine Functions</u> Mine Mooring Mine Neutralization/Destruction</p> <p><u>C3I Functions</u> Information Management Communication Guidance/Navigation/Position Location Avionics/Vetronics/Display Systems</p> <p><u>Electronic Warfare Functions</u> Electronic Countermeasures Jamming Deception Cryptography Electronic Counter Countermeasures Low Probability Electromagnetic Signal Measurement/Intelligence Jam Resistance</p> <p><u>Assessment/Analysis Functions</u> Simulation Weapons and Munitions Effects/Target Kill Assessment Vulnerability Analysis</p> <p><u>RDT&E Functions</u> Energetic Materials Manufacturing Technology Electronics Other Than Electronics Materials Development Metals, Ceramics, Organics and Composites Electronics Test Equipment/Technology Structural Electronics Reliability Maintainability Structures, Including Design and Manufacture Missile Aircraft Hull Body/Chassis</p>	<p><u>Miscellaneous Functions</u> Multi-Function Applications Robotics Human Factors/Human Engineering Artificial Intelligence/Adaptive Systems Basic Scientific Research/University Interactions</p> <p><u>Supply/Support/Construction Functions</u> Material Distribution and Payload Handling/Supply Systems Training Field Services (Water, Food, Tents, etc.) Bridging/Obstacles Support and Auxiliary Equipment Habitability Environmental Effects Facility Construction</p> <p><u>Management/Personnel Functions</u> RDT&E Management Acquisition Management Financial Management Medical/Casualty Care Performance Appraisal</p> <p><u>Other Embedded Functional Areas</u> Avionics Audio Signal Processing and Enhancement Command and Control Command, Control and Information Command, Control, Communications and Information Command, Control, Communications, Computers and Information Digital Signal Processing Guidance and Control Image Processing and Enhancement Operational Flight Program Simulation Telemetry Target Seeking Embedded Trainer Software Embedded Weapon</p> <p><u>Other Software System Functions</u> Decision Support Financial, Accounting, Bookkeeping, Payroll, etc. Information System Management Information System Personnel, Human Resources, etc. Operating System Online Training or Education Software</p>
---	---	--