DEPARTMENT OF DEFENSE
SOFTWARE RESOURCE DATA REPORT (SRDR)
VERIFICATION AND VALIDATION (V&V) GUIDE
VERSION 3.0

29 February 2016

Primary Authors:

Nicholas Lanham, NCCA

Mike Popp, OMNITEC/NAVAIR 4.2

This document was generated as a result of the AFCAA-led, Software Resource Data Report Working Group (SRDRWG). This working group represented a joint effort amongst all DoD service cost agencies. The following guidance describes SRDR data verification and validation best practices as documented by NAVAIR 4.2, NCCA, AFCAA, ODASA-CE, and MDA.

Software Resource Data Report (SRDR) Unified Review Function (SURF) Team Charter

The purpose of the of SRDR Unified Review Function (SURF) team is to maintain the Government V&V efforts implemented by the Department of Defense (DoD) cost community over the past decade. The SURF team will supplement the Defense Cost and Resource Center (DCARC) review at the time of the SRDR submission, but will not replace the standard DCARC review. Instead the SURF team and DCARC analyst will assess the SRDR submission jointly; the SURF team will supplement the identified DCARC analyst's review at the time of the SRDR submission in place of the existing process, in which the primary Government review occurs prior to final DCARC acceptance. This adjustment to the existing process significantly enhances the Government's review effort and considerably reduces the amount of Government and contractor processing time due to incorporation of the SURF team's comments prior to final DCARC acceptance.

In the previous process, the typical SRDR report would be reviewed by DCARC who would identify issues such as errors, allocation schemes, and inconsistent rollup mappings, etc., which would then require discussion with the contractor and re-submittal of the report in order to address the anomalies. This process would then be repeated until all of the anomalies had been corrected and often required numerous re-submissions. Once the report was accepted and the final DCARC acceptance letter was generated the Government review team tasked with populating the master SRDR dataset would have access to the SRDR and would then raise additional concerns to the Points of Contact (POC) listed on the DCARC acceptance letter. After the issue has been discussed with DCARC, CAPE, and the supporting service cost agency, the Government review team would typically re-engage with the submitting contractor to request a follow-up adjustment to address the issue. In most cases, this process required several follow-up discussions between the submitting contractor and Government review teams in order to remedy the identified issue, and the contracting organization would often contest the need to re-submit the report.

Establishing a joint SURF team that includes members from each DoD service cost organization and altering the process to provide for Government review prior to final DCARC acceptance will provide significant benefit to Government review personnel and supporting contracting organizations by reducing the time required to verify and validate SRDR submissions. In addition, the SURF team will conduct the Government review based on a consistent set of V&V guidelines (e.g., SRDR V&V Guide) that will be referenced by each DoD organization that will review SRDR data submissions. The SURF team will manage and update the existing SRDR dataset hosted to the DACIMs Web portal – a dataset that is already maintained by Government personnel, is frequently referenced by the DoD cost community, and contains historical software development data specific to DoD weapon systems.

Verification and Validation (V&V) Guide Purpose

The purpose of the following SRDR data review guide is to provide a structured list of questions, focus areas, and possible solutions to cost community members tasked with inspecting SRDR data submissions for completeness, consistency, quality, and usability. Contractors who have been awarded contracts that meet or exceed the SRDR reporting requirements included within DODM 5000.04-M-1 are also required to submit SRDR data in accordance with the Defense Cost and Resource Center (DCARC) submission process. These submissions undergo a rigorous DCARC review designed to focus primarily on data formatting consistency, ensuring that all required fields have been populated and that the submissions comply with the reporting timeline(s) and other requirements included within the corresponding Contract Data Requirements List (CDRL).

Historically, some data submissions have included erroneous data that could not have been identified as such without several comparisons to prior builds and/or additional reviews of the supporting SRDR data dictionary. Even in the event that a submission has been accepted and does not include erroneous information, there may be several critical SRDR reporting elements that have not been populated or whose final values have been reported using a vastly different methodology than prior submissions (whether they are Source Lines of Code (SLOC), Effort, Requirements, etc.). As such, the quality of these data submissions has frequently been inconsistent and, at times, has even required additional revision after formal DCARC acceptance.

As a result, the following set of SRDR V&V questions and guidelines has been developed to highlight data inconsistencies, erroneous values, and/or substantial deviations from historical reports (i.e., prior software builds and/or increments). Over the past seven years, these guidelines have been used to determine the quality of each SRDR data submission, down to the lowest level Computer Software Configuration Item (CSCI). Without the following V&V guidance, the Department of Defense (DoD) cost community would not have the ability to consistently (organization-to-organization) isolate software cost relationships and performance trends based on a grouping of quality SRDR data submissions.

Considering the continuing reliance on software-intensive development efforts required to support DoD programs, this document represents an effort to proactively establish a consistent guide for any organization assessing the realism, quality, and usability of SRDR data submissions. In addition, this document has assisted the DoD cost community in establishing numerous software cost estimating relationships, predictive software performance algorithms, and software cost estimating models currently used across the department.

While it is impossible for any guide to cover every situation an analyst may encounter when reviewing SRDR data submissions, this guide provides the cost analysis community with a list of questions structured by SRDR reporting area, as well as a list of possible resolutions that describe how issues were historically handled by the reviewing Government analysis team. After reviewing all of the SRDR areas, a qualitative determination can be made on the usability of a given data point for future estimating and analysis purposes. The guide is organized into the following areas:

0 Review of an SRDR submitted to DCARC

2.0 Quality Tagging

3.0 Solutions for Common Findings

4.0 Pairing Data

0

Possible Automation

1.0 Review of an SRDR submitted to DCARC

In order to accept or evaluate the Quality of an SRDR submission, the report must first be reviewed for completeness and consistency. The following sections highlight the standardized list of questions used to assess the Quality of each record included within the existing SRDR database and highlights the major areas of review that are recommend to be referenced when assessing an SRDR submission. It is recommended that items included within this guide be utilized at the lowest level Computer Software Configuration Item (CSCI) or Contractor Work Breakdown Structure (CWBS) element provided within the submission.

1.1. Reporting Event

When evaluating the Reporting Event portion of the SRDR, the analyst is examining the following:

- Is the submission compliant with the CSDR Plan, i.e., a comparison of the submission to the plan requirement?
- Does the report reference the CSDR Plan?
- Has the play type been identified (for example: prime contract, subcontract, or not applicable)?
- What is the difference between the report As Of Date against the Actual Date (e.g. Date Prepared) of submission/acceptance (this is important when making a decision about the usability of an initial report – see 4.0 Pairing Data)?
- Is there consistency of nomenclature and WBS numbering from submission to submission?
- Is there an easily identifiable event associated with the submission (for example: Contract Award, Build 2 Release, Build 1 Complete, Contract Complete, etc.)?
- If there are prior submissions, is this submission an update to a prior submission or a new event? If the submission is an update to an existing submission, does the latest submission clearly describe what report the prior submission is linked to?
- If a prior submissions exists, is the information that has changed readily identifiable and a reason for the change provided (either in the data dictionary or comments section)?
- Is it clear if the information represents a Technology Demonstration (TD) or Engineering and Manufacturing Development (EMD) phase if the program is in that stage of development?
- If this is a Final Report, was there an Initial Report it can be traced to?

1.2. Demographic and Common Heading Information

- Has the program name been identified?
- Has the Major Defense Acquisition Program (MDAP) or Major Automated Information System (MAIS) designation been listed?
- Is the Prime Mission Product (PMP) name been clearly identified (for example: most current official military designation?
- Has the Defense material item category been provided in accordance with MIL-STD-881C guidance (for example: Aircraft, radar, ship, Unmanned Ariel Vehicle (UAV) system)?
- Is the system description been included within the submission?

- Have the program phase and/or milestone been included within the report (for example: Pre-A, A, B, C-LRIP, C-FRP, O&S, etc.)?
- Has the contractor or organization that performed the work been identified?
- Has the reporting contractor or organization address and zip code been included?
- Has the specific site or subdivision for the contractor been identified?
- Has the contractor or submitting organization illustrated whether they were the primary or secondary developer?
- If effort was outsourced, has the outsourced organization been provided?
- Is the contract number reported?
- Are precedents reported and consistent from submission to submission?
- Is the software process maturity and quality reporting definition provided (For example: Capability Maturity Model (CMM), Capability Maturity Model Integration (CMMI), or other alternative rating)?
- Is the Process Maturity rating reported with an associated date, and has it changed from a prior submission?
- Is the specific U.S Military service branch or customer identified (For example: Navy, Air Force, Army, prime contractor, etc.)?
- Has the specific contract type been identified? For contracts, task orders, or delivery orders with multiple CLINs of varying contract types, the Contract Type reporting should be the one associated with the plurality of cost.
- Has the total contract price been identified?
- Has the contract Period of Performance (PoP) been identified?
- Has the report type been identified (for example: Initial, Interim, or Final)?
- Is there a single submission Point of Contact (POC) and supporting contact information included within the report?
- Has the funding appropriation been identified (for example: Research, Development, Test and Evaluation (RDT&E), Procurement, Operation and Maintenance (O&M), Foreign Military Sales (FMS), etc.)?
- Does the submission include adequate detail within the comments section to support analysts who may reference the submission sometime in the future (For example: Provide context for analyzing the provided data, such as any unusual circumstances that may have caused the data to diverge from historical norms)?

1.3. Software Characterization and Development Process

When contractors include Operating Environment (OE) and Application Domain (AD) designations in their SRDR submissions, the Government review team must then determine if those designations are consistent with the data validation guidance referenced below. Application Domain designations provide the cost community with an additional set of data groupings which support software Cost Estimating Relationship (CER) development. These designations add onto the existing set of software estimating relationships specific to language type, analogous program(s), and/or developing organization(s). In addition, application domains also provide the ability to more efficiently map existing SRDR data submissions to commercial software cost estimating models. In an attempt to ensure consistent application of OE and AD data identification from submission to submission, the following guidance, category descriptions, and review process have been provided below.

The existing AD reference list has been reduced to 17 categories, which may be grouped into four higher-level Super Domains (SD). In addition, OE designations may be mapped into eight summary-level environments split into Manned or Unmanned categories, if applicable.[1]

1.3.1. Super Domain and Application Domain Designation:

When assessing the Application Domain (AD) designation chosen by the submitting organization, the reviewing Government analyst must begin by tracing the given Computer Software Configuration Item (CSCI) into an SD category using the mapping tree illustrated within Figure 1. During this process, the reviewing analyst must be sure to review the SRDR data dictionary and supporting functional description for each CSCI in the overall SRDR submission. When reviewing the SRDR submission details, supporting comments, or data dictionary for the given CSCI, the analyst must then refer to the SD/AD definitions included within Appendix A.

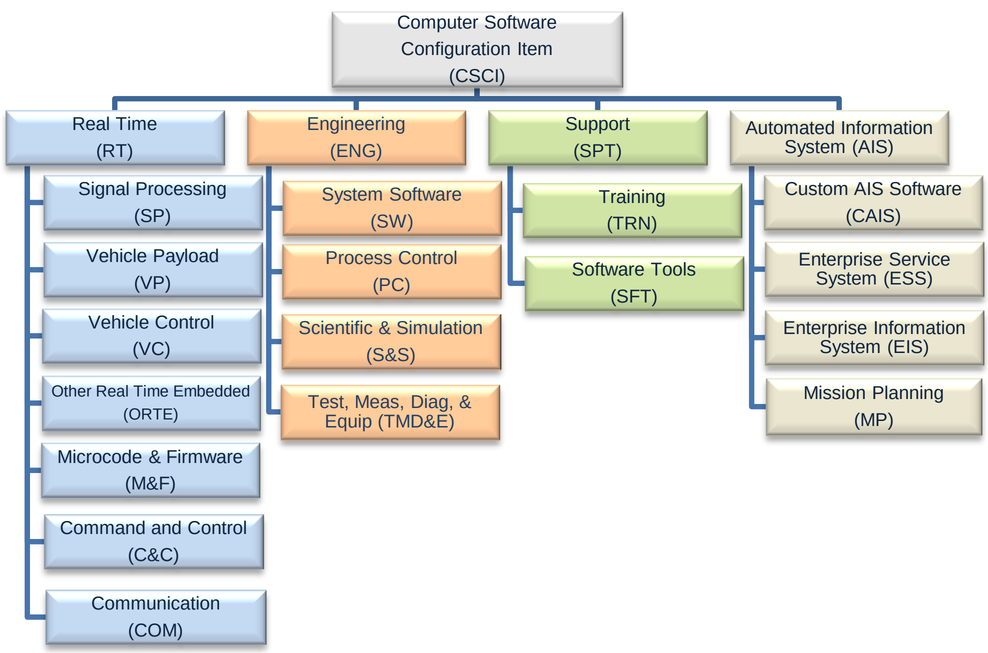

Figure 1: Classification Map (Super Domains and Application Domains)

After the analyst has reviewed the supporting functional description and SRDR data dictionary documentation, the CSCI can be mapped into one of the SD categories shown within Figure 1. After determining the summary-level SD designation, the government analyst can then isolate lower-level AD designations by referencing the definitions in Appendix A.

Another method of mapping a CSCI to SD and AD category is to analyze the WBS title to isolate the proposed software function specific to that CSCI. This function can also aid in the comparison of planned CSCI function to the SD and AD category description in Appendix A. The following

[1] "SRDR Working Group Data Collection on CSCIs by Application Domains" version 12 (Jones, et al.) 2015.

list of questions should assist the reviewing Government analyst to ensure the necessary information has been provided for the correct SD and AD designation.

- Does the SRDR submission, comments section, or data dictionary include a clear system level functional description and software operational overview?
- Does the SRDR submission include a detailed functional description for each CSCI? (Note: it is not uncommon for submissions to include the same, abbreviated functional description for every CSCI rather than a detailed functional description specific to each CSCI included within the SRDR submission. If this scenario occurs, we recommend contacting the submitting organization for additional detail).
- If a detailed CSCI-level functional description is not included within the SRDR submission, is it included within the supporting data dictionary or comment section?
- Does each CSCI or WBS element include a naming convention specific to the intended software function?

1.3.2. Operating Environment (OE) Designation:

The latest version of the SRDR Data Item Description (DID) requires the submitting organization to select the appropriate operating environment specific to the system specific to the SRDR. In order to confirm the most applicable OE has been selected, the reviewing Government analyst must compare the system description included in the MS Excel submission, SRDR data dictionary, or statement of work, to determine the appropriate OE designation. If the SRDR submission includes a sufficient system level description, the analyst can use the OE definitions included in Table 1 to select the appropriate OE type if one has not already been included by the submitting organization.

Table 1: Operating Environment (OE) List

Operating Environment (OE)		Examples
Surface Fixed (SF)	Fixed (SF)	software is at a fixed site
Surface Mobile (SM)	Manned (SMM)	software is moved somewhere and set up
	Unmanned (SMU)	
Surface Portable (SP)	Manned (SPM)	software is in a handheld device
Surface Vehicle (SV)	Unmanned (SVM)	software is embedded in as part of a moving ground vehicle
	Unmanned (SVU)	
Air Vehicle (AV)	Manned (AVM)	software is embedded as part of an aircraft
	Unmanned (AVU)	
Sea System (SS)	Manned (SSM)	software is embedded as part of a surface or underwater boat/ship or boat
	Unmanned (SSU)	
Missile System (MS)	Unmanned (MSU)	Software is embedded as part of a missile system
Ordnance System (OS)	Unmanned (OSU)	software is embedded as part of an ordnance system
Space System (SPS)	Manned (SPSM)	software is embedded as part of a spacecraft
	Unmanned (SPSU)	

In order to map a CSCI submission to an OE, the reviewing analyst must compare the system-level functional description (included as part of the SRDR data dictionary) against the list of OE descriptions in Table 1. Each of the provided OE designations has been derived from the list of "major systems" or "end products" listed within MIL-STD-881C[2], section 1.5.4, and is specific to DoD operational system environments.

In most cases, the OE designation can be determined using the system's functional description to isolate the system's operational platform type (i.e. aircraft, ground vehicle, etc.) and to determine the Manned or Unmanned operational configuration for a given system. When system-level characteristics (i.e. manned vs. unmanned aerial or ground vehicle) cannot be clearly identified and mapped into the OE categories described within Table 1, the reviewing Government analyst must request additional system level information be provided. The analyst must also confirm that different OE designations are not required for separate, lower-level WBS or CSCI elements. In addition, the following list of questions should assist the reviewing Government analyst to ensure the necessary information has been provided to apply the correct OE designation.

- Does the SRDR data dictionary include a clear system-level functional description and software operational overview?
- If a system-level functional description has been included, does it include details regarding manned or unmanned system configurations?
- Has the state of development been identified (For example: Prototype, Production Ready, or a mix of the two)?

1.3.3. Development Process:

- Has the contractor listed a standard process, or is there a unique identifier in the SRDR data dictionary describing what the process is?
- Has the contractor indicated whether the software is an upgrade or new development? If not, why not?
- If an upgrade, does the SW sizing reflect significant reuse or modification SLOC totals when compared to New code?
- What precedents or prior builds are identified to give credibility to the upgrade designation?
- Has the development method also been identified (for example: Structured Analysis, Object Oriented, Vienna Development, etc.)?

1.4. Personnel

- If skill mix percentages are included within the SRDR submission, do the percentages sum to 100%?
- If there was a prior submission, has the skill mix changed dramatically and, if so, is there an explanation why? Conversely, did it remain unchanged? If so, why?
- Does the data dictionary define what the skill level requirements are, and is the contractor adhering to that definition?
- Does the skill mix make sense relative to the complexity of the code (unusual amount of very low or very high skill mix, for example)?

[2] "Work Breakdown Structures for Defense Materiel Items," MIL-STD-881C, 3 October 2011.

- Does the peak headcount make sense against the reported schedule and hours? A simple test is to divide the total reported hours by the schedule months and then convert the resulting average monthly hours into an average Full Time Equivalent (FTE) count using the reported hours in a man-month. The peak headcount must be higher than this FTE monthly average. At the same time the peak headcount should not be wildly disproportional to that average either.
- Has the contractor identified the standard hours in an accounting month when determining the peak FTE?

1.5. Sizing and Language

This is a very important section of the review, and one where interpretation and very close review of prior submissions (if available) will help when assessing the quality of the data point.

1.5.1. Requirements

Requirements count is one of the more inconsistent areas within SRDR submissions (e.g. total and/or new requirements, internal and/or external, as well as the method of counting).

- Does the submission clearly illustrate the number of Inherited, Added, Modified, Deleted, and Deferred requirements for both internal and external categories?
- Has the submitting organization separated the provided requirements by Security, Safety, and Privacy or Cybersecurity?
- Do the number of requirements trace from the parent to the children in the WBS? If not, this could imply that some portion of the software effort is only captured at higher-level WBS/CSCI elements and should be cross checked.
- Does the total requirements value also include a clear breakout of new requirements?
- Does the data dictionary provide a description of how a requirement is counted (e.g. discrete shall statements, functions derived from shall statements, etc.) and what constitutes a new requirement versus existing?
- If external interface requirements are identified, does the dictionary describe what these are and how they were determined?
- Has the submitting organization included Requirements Volatility rating?

1.5.2. Source Lines of Code (SLOC)

- Was the primary programming language reported?
- Were SLOC counts reported as estimated or actual results?
- How was the SLOC counted (e.g., logical, non-comment source statements, physical, total, etc.)?
- Did the submitter us the Aerospace-approved version of the University of Southern California (USC) Center for Systems and Software Engineering (CSSE) Unified Code Count (UCC) tool to count the provided SLOC totals? If not, was the name of the code counting tool used by the submitting organization included within the supporting comments section and/or data dictionary?
- Has SLOC been reported for each separate CSCI or WBS element included within the submission?

- Are the SLOC counts for different types of code (e.g., new, modified, reused, auto-generated, Government-furnished, and deleted) separated or are they mixed together?
- Did the submitter describe each code type?
- Do multiple records have the same SLOC sizing data (i.e. size data is repeated for each code type or total size is repeated)? Should they be repeated because they are roll ups of WBS/CSCI elements or has a proration scheme been used to estimate sizing values?
- For a Final report does the size look realistic? For example: is all of the code rounded to the nearest 1000 lines, or does the dictionary indicate that they had difficulty counting code that may have come from a subcontractor?
- Were SLOC counts reported in another data submission and are they traceable from submission to submission or build to build, if applicable?
- Were SLOC counts reported in another data submission and are they traceable from submission to submission or build to build?
- For code that is part of a follow-on build, is the code stand-alone, or do the counts look like they include the prior build as part of the total (the exception being prior build code rolled in as reuse)? Note: If the code does not appear to be stand-alone, it may not be usable for analysis without additional processing.
- Has the contractor or submitting organization provided the name of the software products that have been referenced to generate the provided reuse SLOC totals?
- When subcontractor code is present, is it segregated from the prime contractor effort, and does it meet the same criteria for quality as the prime’s code count?
- Has the submitting organization clearly listed the number and names of the Commercial Off the Shelf (COTS) or Government Off the Shelf (GOTS) used to develop each CSCI or WBS element?
- If COTS or GOTS items have been included within the submission, has the submitting organization provided the SLOC total required to integrate the identified COTS/GOTS product (i.e. Glue code)?
- If COTS or GOTS integration or glue code has been included within the submission, does the total seem realistic when compared to the total SLOC included in the CSCI or WBS element (For example: COTS integration code equals 500 KSLOC and the total SLOC for the specific CSCI or WBS element equals 150 KSLOC)? Note: this scenario sometime occurs when the submitting organization counts the total SLOC of the specified COTS or GOTS product vice the integration or glue code required to integrate the product.

1.5.3. Non-SLOC Based Software Sizing

- Were SLOC counts reported, or were other counting or sizing metrics used (e.g. function points, use cases, rung logic ladders, etc.)? If so, has the submitting organization obtained the appropriate authorization to report Non-SLOC based sizing within the corresponding CSDR plan?
- If function points have been provided has the submitting organization clearly illustrated the function point count type (For example: Enhancement Project, Application, or Development Project)?

- Has the submitting organization provided the number of Data Functions and Transactional Functions (For example: Internal Logic Files, External Interface File, External Inquiries, External Inputs, and External Outputs)?
- Has the submitting organization included the Value Adjustment Factor?
- If the submitting organization has provided sizing metrics using the Reports, Interfaces, Conversions, Extensions, Forms, and Workflows (RICE-FW) convention, has the complexity of each RICE-FW category been provided?

1.5.4. Product Quality Reporting

- Has the submitting organization provided a breakout of the number of software defects Discovered, Removed, and Deferred?
- Has the priority level for each category of software defects been provided?
- If the report is an interim or final submission, has the number of Discovered, Removed, and Deferred defects changed from the previous submission? If significant changes have occurred, does the supporting comments section and/or data dictionary provide details regarding what drove the significant change in product quality metrics?

1.6. Effort

Effort is an interesting area within historical data submissions due to many variations in how the contractors choose to “bucket” hours, along with the completeness of those hours. When assessing the quality of a given data point, the most important variable to assess is completeness vice the allocation of those hours into the seven primary collection areas (e.g. Requirements Analysis; SW Architecture and Design; Code and Test; SW and Systems Integration; SW Qualification Testing; Developmental Testing; and Other). Determining completeness is not always easy, especially when the contractor collects or reports their historic actuals using alternative categories or just reports all effort within the Other category. This does not mean the data point is not useful. In fact, in many cases it is as useful as a data point that did include a partition of hours; provided the submitting organization is clear on what the reported hours include.

Effort has the additional challenge of factoring in phase (e.g. Technology Development, Engineering Manufacturing and Design, or build sequence), and in an increasing number of submissions, the collection of activities like system test or software program management are included as a completely separate WBS or CSCI reporting element that is not allocated back to the corresponding group of CSCI or WBS element(s). Depending on what is included in the submission, it may be necessary to allocate the provided hours in order to make it useful for analysis at the CSCI or WBS level. In other cases, the determination may be made that only roll-up data can be useful. Whatever the case, the goal is to conduct minimal post-processing of the contractors’ submitted information (Note: this represents a last-ditch effort to preserve a valid data point with non-allocated contractor reported labor effort). It is better to mark a data point as questionable or not usable if it is not possible to reasonably allocate or combine the data, or if the determination of completeness cannot be made. At no point should normalization be conducted in an attempt to account for missing content or to make the data point “complete.” Common questions to ask when looking at the effort are:

- Was effort data reported for each CSCI or WBS?
- Was effort data reported as estimated or actual results? If the submission includes estimated values and actual results, does the report include a clear and documented split between actual results and estimated values?

- Is the effort data reported in hours?
- Is effort data broken out by activity?
- Was the specific ISO 12207:2008 activities that are covered in the effort data (For example: Requirements analysis, architectural design, detailed design, construction, integration, qualification testing, and support processes) clearly discernable?
- Were common WBS elements/labor categories such as System Engineering (SE), Program Management (PM), Configuration Management (CM), or Quality Management (QM) been broken out separately?
- Is there an explanation of missing activities included within the supporting SRDR data dictionary?
- Was the effort data for each activity based on a proration scheme, i.e. percentage based? The analyst will typically have to calculate and confirm if the same percentages show up across multiple CSCIs or WBS elements.
- Do the children or lower-level WBS/CSCI elements add up to the parent? If not, is there effort that is only captured at a higher-level WBS/CSCI level that should be allocated to the lower-level WBS/CSCI elements?
- Did the submitter describe what activities are included in the Other category?
- Does the submission include Effort and SLOC data specific to each CSCI or WBS element included within the report? For example, does the WBS/CSCI include effort data without corresponding SLOC sizing data? (Note: this scenario is most common for software categories such as Program Management, Quality Assurance, Configuration Management, etc. See 3.1 Allocation)
- Does the submission include unique values for each of the lower-level CSCI or WBS elements? For example, do multiple related records have the same effort data (i.e. activity effort is repeated or total effort is repeated)?
- Has the subcontractor's effort been reported separately? For example, has the subcontractor data been mixed within the prime contractor's values, is the data missing, or has the data been reported separately?
- Are the provided effort values specific to the specific development effort referenced within the submission? For example, is the effort data cumulative from the prior build? Is the corresponding sizing information also cumulative or stand alone? Depending on what is found, it may be necessary to combine or conduct post-processing to make it a valid, standalone data point.
- Do all CSCI or WBS elements include effort values that are inclusive of common "overhead" or "indirect" labor categories within the provided effort total? For example, are there separate CSCI or WBS elements that reflect "effort-only" data within a separately reported CSCI or WBS element? (i.e. has quality assurance or configuration management effort been reported as separate WBS/CSCI elements)? If so, can that effort be reasonably allocated back to the primary WBS/CSCI?
- Have the provided Effort totals been broken out by month?
- If subcontractor hours have not been provided, did the reporting organization provide subcontractor dollars?

1.7. Schedule

At the surface level, a schedule assessment should be straightforward – i.e. is there a start and stop date (or months elapsed) for each activity or for the WBS/CSCI as a whole? But just like Effort, there are nuances and questions to ask when assessing schedule data. One very common occurrence is the reporting of identical, or near identical, schedules for every CSCI or WBS element regardless of size or requirements. In this case the schedule may be accurate, but the level where the schedule analysis occurs would be the roll-up level and not the individual CSCI. Similarly, on follow-on builds the analyst must look closely to see if the schedule is independent for the build or reflects a continuation of the prior activity. It is very possible that the CSCI or WBS element could be good for productivity analysis, but not good for schedule analysis. At the same time, roll-ups that are not used for productivity analysis may be used for schedule analysis. Common questions for schedule are:

- Has schedule data been included in the submission?
- Has the submitting organization clearly stated if the provided schedule data was reported as estimated, allocated, or actual results?
- Has schedule data been reported in number of months from contract start or as calendar dates?
- Is schedule data broken out by SRDR activity?
- Does the report include unique schedule start and end date values? For example, do multiple records have the same schedule data, e.g., same calendar dates for multiple WBS/CSCIs or builds?
- Does the provided schedule data represent the unique duration specific to the corresponding product sizing and/or effort data? For example, does the reported schedule data represent concurrent builds vice separate labor categories?
- If so, do the dates in the current submission align with the prior, and if not is there an explanation for significant changes in the schedule?

1.8. Estimate at Completion (EAC) Values

As a result of the latest SRDR Data Item Description (DID) update, DoD service cost agencies have requested the inclusion of contractor Estimate at Completion (EAC) values for labor hours (or subcontractor dollars if subcontract has not been provided). These fields will provide Government analysis teams with the ability to trace estimated hours up to the parent WBS element, as well as to the corresponding EAC Hours (EAC_H) are summarized by Monthly hours, Actuals To Date (ATD), and Total Hours. Common questions that should be asked when verifying contractor-supplied EAC values are:

- EAC_H: Has a description been provided that describes which ISO 12207:2008 elements have been included within the provided total?
- EAC_H: Do sub-element EAC values sum to the parent EAC total value?
- If the report is a final report, does the provided ATD total match the provided EAC total?

2.0 Quality Tagging

In the end, the purpose of conducting an SRDR V&V review is to determine if the data point(s) can be used when building future estimates and cost models. Historically, only a relatively small percentage of the submissions will actually fall into the category of “Good” for either productivity or schedule analysis. This does not mean that submissions are bad in the aggregate. In fact, most data points generally pass the screening process but turn out to be an initial, or a roll-up, or an interim or some other case where using the data point may present an issue. In addition to reviewing the data itself, there is also an evaluation of the resulting productivity to see if it falls within acceptable benchmarks. Falling outside a specified benchmark range is not sufficient reason to reject a data point, however, the reason why a given value falls outside this range must be determined. Appendix B provides Government analysts with a list of ways that data can be tagged for productivity (size and effort information) and Appendix C for schedule usage. Appendix D contains an SRDR scorecard method of evaluating data that uses four unique "tag" and "validation" criteria to assist with data parsing and making determinations about data quality.

3.0 Solutions for Common Findings

Past review has found that there are times where the government has to conduct post-processing of the provided data in order to make it suitable for estimating or analysis purposes. Fortunately, this is not a normal occurrence, but a necessary one. The three most common issues encountered are: (1) collection of hours against a WBS that should be allocated to the CSCI or other WBS element to make them complete; (2) software build information that is not stand-alone and the builds need to be combined in some fashion; and (3) combining of a TD portion with the subsequent EMD portion to make a complete software development data point. These three issues and the corresponding recommended solutions will be addressed in the subsections that follow. (Note: Some of these methods rely on the calculation of Equivalent SLOC, or ESLOC. The precise ESLOC formula is not material, as long as it's applied consistently.)

3.1. Allocation

In the past, "common" collected hours have been allocated back to the CSCI's based on ESLOC. This adds no hours at the parent level; it is merely a distribution scheme to make the data comparable to other submissions that do not have the "common" WBS collection as a separately reported item. If the decision is made to not allocate then the parent becomes the "Good" data point and lower-level CSCI's would be flagged as "question with missing hours." Table 2 illustrates an example of how allocation has been addressed in a collection of CSCI's for a given program.

Table 2: Allocation

CSCI	ESLOC	Reported Hours	Allocated Hours	Total Hours
1	15,914	6,187	700	6,887
2	21,907	8,414	964	9,378
3	1,826	1,406	80	1,486
4	8,014	5,038	353	5,391
5	23,900	14,217	1,051	15,268
6	146,524	46,192	6,445	52,637
Mgmt. and Support	0	9,593	0	0
Total	218,085	91,047	9,593	91,047

As the table shows, the allocation of the Management and Support hours across the 6 CSCIs does not change the total Effort but rather provides a more accurate method of allocating management and support hours that should have been reported by individual CSCI. Using CSCI #5 as an example, it represents (23,900 / 218,103) = 10.96% of the ESLOC, and therefore it is allocated 10.96% of the 9,593 Management and Support hours, or 1,051. The reported 14,217 hours plus the 1,051 allocated hours results in a total of 15,268 hours.

3.2. Combining

Another issue that can arise with multiple builds is when either Effort or Size is reported as cumulative (i.e. including previous builds) while the other is reported as discrete (i.e. the total for that build only). In this case, we can combine the build information to create a high quality data point. This is easier to explain with an actual example. Table 3 illustrates this combining issue by showing the size and hours for Build 1 and 2 as well as the new nominal data point developed by the Government review team.

Table 3: Combining

Build No.	New	Mod	Reuse	Auto	ESLOC	Hours	Hours/ESLOC
Build 1	62,206	16,215	427,770	0	93,469	209,359	2.23
Build 2	98,032	25,107	456,048	0	132,566	111,191	0.83
Combined	98,032	25,107	456,048	0	132,566	320,550	2.42

By reading the supporting SRDR data dictionary and looking at the aggregate of submissions for this project, it became clear that the size for Build 2 was building upon the work done in Build 1, but that the reported hours were discrete for each build. By combining the hours for Builds 1 and 2, along with the cumulative code reported in Build 2, a usable data point can be created. If this combination process were not completed, then neither one of the data points should be flagged as good, and there would be questionable productivity data from these submissions. (Build 1 would be considered an Interim, and Build 2 would be a question on size.) In this instance, the apparent spike in productivity (i.e., low Hours/ESLOC) for Build 2 is misleading, because the calculation includes only the Build 2 Effort but takes "credit" for all the code developed to date, including during Build 1.

3.3. Early Acquisition Phase Combining

This last issue stems from recent acquisition guidance requiring separate reporting for TD and EMD phases, vice the traditional System Design and Development (SDD) strategy, which is driving a break-point within contractor submitted effort and schedule data. In order to make this information comparable to other data, it will be necessary to combine TD and EMD data to make a "complete" development data point. At this point in time, no EMD report has followed a corresponding TD report, so the full difficulty of conducting this analysis remains unknown.

4.0 Pairing Data

The process of pairing an Initial and Final report to create a "paired data" point goes beyond just matching two reports, and aims to verify that the pair is suitable for growth analysis. In order to qualify, both the Initial and Final submissions must be analyzed and tagged as "good." The Initial must be the first Initial, keeping in mind that there can be several Initial submissions for the same event; and the Initial submission must not have a prolonged time lapse between when it was *planned* to be submitted and when it finally was accepted (this is an "unfair" opportunity for the contractor to update their initial "estimates" to something closer to what they are/were experiencing). In the event that either of the Allocation or Combining processes was required to create a “good” data point (see Sections **Error! Reference source not found.** and **Error! Reference source not found.**, respectively), the same process must have occurred on the Initial as well the Final.

5.0 Possible Automation

The establishment of a standardized XML data schema can potentially automate portions of the submission review process described above. Performing automated reviews prior to acceptance of the SRDR by DCARC has the potential to significantly increase the quality of the submission by finding issues when the government still has leverage to obtain contractor corrections. It also eliminates errors associated with translation of the data into alternative centralized storage formats like MS Excel data consolidation or the Cost Assessment Data Enterprise (CADE). The level of automation undertaken will evolve both with an understanding of what can be automated and with the funding to implement it. Initially, the review may start with the submission as a standalone entity, but with time there can be comparisons done with prior submissions looking for inconsistencies. While automation will not replace a review team making a final determination of quality, it can eliminate much of the drudgery associated with time-consuming mechanical checks and data transfer. Automation can:

- Flag empty data fields
- Validate that a supporting CSDR Plan exists
- Verify that the Actual Date is on or after the Report As Of Date, and that the submission occurs relatively close to the Planned Date.
- Verify that nomenclature and WBS numbering are consistent with previous submitted reports
- Flag fields that have changed between submissions of the same report
- Verify integer data in fields that require numbers, check that dollars are not reported
- Verify skill mix percentages add to 100%, if provided
- Verify peak head count is greater than or equal to computed average FTE (data must exist for Effort, Schedule, and hours per person month)
- Identify multiple records that have identical Size, Effort, or Schedule data
- Verify date data in fields that require dates
- Flag activities that do not have Effort data
- Flag activities that do not have Schedule data
- Verify that End Dates are not before respective Start Dates
- Verify that Child CSCI/CWBS elements properly sum to provided Parent-level values

Appendix A – Super Domain and Application Domain Categories:

Super Domain (SD) Descriptions and Categories:	
Real Time (RT)	
Description	RT is the most constrained type of software. These are specific solutions limited by system characteristics such as memory size, performance, or battery life. These projects take the most time and effort due to constraints e.g., • May have guaranteed execution requirements i.e. missed deadline means catastrophic results • May have to be compact and efficient due to limited storage capacity and high throughput requirements • Could have very high reliability requirements (life critical, manned mission) • Might have tightly coupled interfaces • Program code may be imprinted on hardware devices • May process sensor inputs and directs actuator outputs • Sometimes executed on special-purpose processors
Application Domain Mapping	Signal Processing, Vehicle Control, Vehicle Payload, Other Real Time Embedded, Command and Control, Communication, and Microcode & Firmware
Engineering (ENG)	
Definition	Engineering software operates under less severe constraints than real-time software. This software may take the outputs of real-time software and further process them to provide human consumable information or automated control of devices. Or the software may perform transformation and aggregation / distribution of data. These projects take more time and effort due to multiple factors, e.g., • May have a fast response time requirement • May have more storage capacity • Might need to be highly reliable but not life critical • May have multiple interfaces with other systems • May implement complex algorithms, models or protocol • Program code can be modified or uploaded • Executes on general purpose processors that may be embedded in special purpose hardware
Application Domain Mapping	System, Process Control, Scientific and Simulation, and Test, Measurement, Diagnostic, and Evaluation
Automated Information System (AIS)	
Definition	Automated Information System software provides information processing services to humans or software applications. These applications allow the designated authority to exercise control and have access to typical business / intelligence processes and other types of information access. These systems also include software that facilitates the interface and control among multiple COTS / GOTS software applications. This software has few constraints, e.g., • Must have acceptable response time • Fewer storage and throughput constraints • Must be reliable enough to prevent data loss • May consist of a single COTS / GOTS solution or multiple products coordinated with customer software • Algorithms, models, and protocols are well understood • Code may not be available for modification • Software restarts are acceptable • Executes on commercial processing hardware
Application Domain Mapping	Mission Planning, Enterprise Service Systems, Custom AIS Software, and Enterprise Information Systems

Support (SPT)	
Definition	Support software assists with operator training and software testing. This software has few constraints, e.g., • Has to have an acceptable response time most of the time • Less limited by storage or throughput • Less stringent reliability requirement • Software restarts are acceptable • Fewer interfaces • Relatively low complexity algorithms, models or protocols • Program code can be modified and uploaded • Executes on general purpose processors on general purpose computer boards
Application Domain Mapping	Training, and Software Tools

Application Domain Definitions and Categories:				
Signal Processing				
Definition	*Software that requires timing-dependent device coding to enhance, transform, filter, convert, or compress data signals*			
Source Definition	***Signal Processing**: Software dominated by functions that enhance, transform, filter, convert, or compress data signals. Large volumes of data are processed using complex algorithms, often with real time operating requirements*			
Typical examples	*Lasers*	*Sonar*	*Acoustic*	*Electromagnetic*
	Signal Processor	*Radar Altimeter*	*Photographic Sensors*	*Motion Sensors*
	Infrared Sensors	*Sensor Assembly*	*Electronic Sensors*	*Seeker Assembly*
	Signal Electronics	*Optical Assembly*	*Tracking Sensors*	*Antenna Assembly*
Vehicle Payload				
Definition	*Software which controls and monitors vehicle payloads and provides communications to other vehicle subsystems and payloads*			
Source Definition	***Vehicle Payload:** Software used to manage and control payload functions (experiment control, sensor management, etc.) for manned or unmanned space applications*			
Typical examples	Fire Control	Mine Warfare	Electronic Attack subsystem controller	Weapons Delivery and Control
	Gun fire control system	Missile fire control systems	Antisubmarine warfare fire control and torpedo fire control systems	Pointing, Command, & Control Interface
	Payload Flight Software	Armament	Survivability Payload	Reconnaissance Payload
	Electronic Warfare Payload	Armament/Weapons Delivery	Intelligence, Surveillance, Reconnaissance Payload	Mission Payload
Vehicle Control				
Definition	*Software necessary for the control of vehicle primary and secondary mechanical devices and surfaces*			
Source Definition	***Flight Systems**: Onboard software used for various functions associated with the operation and control of airborne platforms (e.g., airplanes, helicopters, missiles, and spacecraft)* ***Avionic:** Software that is on-board & controls the flight and operation of the aircraft . Please note that the "avionic" example is only inclusive of aviation based systems however; vehicle control software will also be included within sea and land based systems that rely on vehicle control software to manipulate the subsystem component examples provided below:*			
Typical examples	*Flight Control*	*Electrical Power*	*Hydraulic*	*Fuel Subsystem*
	Propulsion	*Attitude Control System*	*Structures & Mechanisms*	*Bus Flight Software*
	Thermal Control	*Landing Gear*	*Controls software*	*Thrust Vector Actuation*
	Executive			

Other Real Time Embedded				
Definition	*Interrupt-driven, embedded software in military and consumer appliances, devices, and products, possibly directing and processing sensor inputs/outputs, generally with a very small executive for an operating system interface to basic processor(s).* *Real-time data processing unit responsible for directing and processing sensor input/output*			
Source Definition	***Radar**: Software used in the operation and control of radar systems* ***Embedded Electronics**: An application that is very hardware-specific and often embedded in the firmware of electronic devices and other hardware* ***Robotics**: Software that provides logic and control for robotic or automation equipment* ***Real Time**: Software that must operate close to the processing limits of the CPU. This is interrupt-driven software and is generally written in C, Ada or Assembly language. It generally operates with a very small executive for an operating system interface to the basic processor*			
Typical examples	*Embedded Electronics/ Appliance*	*Robotics*	*PDAs*	*Telemetry, Tracking, & Command (TT&C)*
	Guidance, Navigation and Control	*Controls and Displays*	*Data Links*	*Radios (device)*
	Remote Control	*Receiver*	*Transmitter*	*Exciter*
	Bombing Computer	*Video and recorders*	*Telephones (device)*	*Built-in-Test*
Microcode and Firmware				
Definition	*Firmware/microcode is software stored on target hardware devices that do not have hard disks and use programmable logic devices. It is a combination of persistent memory and the program code and data stored in it*			
Source Definition	***Microcode and Firmware**: Software that is the architecture of a new piece of hardware or software that is burned into silicon and delivered as part of a hardware product. This software is the most complex because it must be compact, efficient, and extremely reliable* ***Microcode and Firmware**: "Firmware/microcode is the way software is stored in devices that do not have hard disks. It is a combination of persistent memory and the program code and data stored in it."*			
Typical examples	*Field Programmable Gate Arrays (FPGAs)*	*Microwave controllers*	*Application Specific Integrated Circuit (ASIC)*	*Programmable Read-Only Memory (PROM)*
	Erasable Programmable Read-Only Memory (EPROM)	*Electrically Erasable Programmable Read-Only Memory (EEPROM)*	*Complex Programmable Logic Device (CPLD)*	*Programmable Array Logic (PAL)*
	Electronic Programmable Logic Device (EPLD)	*Field Programmable Logic (FPL)*		

Command and Control				
Definition	*Software that allows humans to manage a dynamic situation and respond inhuman real time*			
Source Definition	*Vehicle onboard master data processing unit(s) responsible for coordinating and directing the major mission systems* ***Command and Control****: An application that provides commands and monitoring between users (and other systems) and hardware (or hardware-embedded software)* ***Command and Control****: Software that allows humans to manage a dynamic situation and respond in human real time*			
Typical examples	*Mission Management*	*Mission Computer Processing*	*Mission Control*	*Command processing*
	Air traffic control	*Data reduction/ analysis*	*Telemetry Processing*	*Battlefield command*
	Battle management			
System Software				
Definition	*Layers of software that sit between the computing platform and applications [1]*			
Source Definition	***Device Driver****: An application that provides low level connectivity services for a particular device (hardware or software) attached to or loaded onto a computer* ***System and Device Utilities****: Software to help analyze, configure, optimize or maintain a computer* ***OS/Executive****: Software that controls basic hardware operations, serves as a platform for applications to run, or that directly contributes to such a system. Multi-user operating systems provide management and security of system users* ***System****: Layers of software that sit between the hardware and applications programs*			
Typical examples	*Operating Systems*	*Infrastructure*	*Framework*	*Middleware*
	Device Driver	*Display Drivers*	*File management*	*Image Processing*
	Interface Driver	*Utilities*		
Process Control				
Definition	*Software that manages the planning, scheduling and execution of a system based on inputs, generally sensor driven*			
Source Definition	***Process Control****: Software that controls various processes by commanding devices, monitoring processes via sensor feedback, and modifying commands as a function of desired behavior versus feedback. Often associated with industrial environments* ***Process Control****: Software that controls an automated system. Generally sensor driven. Examples are software that runs a nuclear power plant, or software that runs an oil refinery, or a petrochemical plant*			
Typical examples	*Temperature control*	*Manufacturing process control*	*Device or instrument control*	

Scientific and Simulation				
Definition	*Non real time software that involves significant computations and scientific analysis*			
Source Definition	***Artificial Intelligence****: Machine learning algorithms or software that often seeks to mimic human intellectual processes* ***Computer Aided Design****: An application for creating, editing, and analyzing graphical models & representations* ***Expert System****: An application that emulates the decision-making ability of a human expert* ***Math and Complex Algorithms****: An application dominated by complex mathematical operations or algorithms, numerical methods or complex analysis* ***Simulation****: Software that evaluates numerous scenarios and summarizes processes or events to simulate physical, business or biological processes, complex systems or other phenomena that may not have simple empirical relationships* ***Graphics****: An application using custom or advanced image rendering (ray tracing, smoothing, fractals, etc.* ***Scientific****: Software that involves significant computations and scientific analysis. This type of software is often sensor driven with data capture schemes to accumulate data (from a spacecraft, say) then followed by extensive data analysis. Frequently written in FORTRAN*			
Typical examples	*System Integration Lab (SIL) Simulation*	*Simulators*	*Offline Data Analysis*	*Expert Systems*
	Math & Algorithm Intensive	*Graphics*	*Statistical Analysis*	*Artificial Intelligence*
	Simulation & Modeling	*Engineering & Science*	*3D Modeling & Animation*	*Trainer Simulations*
		Computer Aided Design (CAD)	*Model Based Systems Engineering (MBSE)*	*Weather models*
Communications				
Definition	*The transmission of information, e.g. voice, data, commands, images, and video across different mediums and distances. Primarily software systems that control or manage transmitters, receivers and communications channels*			
Source Definitions	***Communications****: An application involved in the transmission and receipt of data across networks* ***Message Switching****: Transport layer software performing packet and circuit switching, handling electronic mail and implementing file transfer protocols* ***Network Management****: Software that monitors and reports on the status of all components of telecommunication networks, including communication links and nodes* ***Voice Provisioning****: An application that provides clients with access to voice communications by providing accounts and appropriate access, unifying the common elements of end-user operations and management of call systems* ***Telecommunication****: Software that facilitates the transmission of information from one physical location to another*			
Typical examples	*Switches*	*Routers*	*Integrated circuits*	*Multiplexing*
	Encryption	*Broadcasting*	*Transfer modes*	*Radios (networks)*

	Network management	*Network Operations*	*Satellite communications*	*Telecommunications*
	Networks (WAN/LAN)	*Protocols (VOIP, TCP/IP, PKI, etc.)*		
Test, Measurement, and Diagnostic Equipment (TMDE)				
Definition	*Software used for testing, measuring, diagnosing, emulating, and evaluating operational hardware and software systems* *Software necessary to operate and maintain systems and subsystems which are not consumed during the testing phase and are not allocated to a specific phase of testing* *This does not include built-in-test (BIT)*			
Source Definitions	***Diagnostics****: An application that performs a comprehensive series of built-in tests on core components and reports the results of each test* ***Test and Measurement Equipment****: Software that supports the peculiar or unique testing and measurement equipment that allows an operator or maintenance function to evaluate operational conditions of a system or equipment by performing specific diagnostics, screening or quality assurance effort at an organizational, intermediate, or depot level of equipment support*			
Typical examples	*Test equipment software*	*Equipment emulators*	*Test driver*	*Maintenance and Diagnostic*
	Fault Tolerance	*Diagnostic*		
Mission Planning				
Definition	*Provides the capability to maximize the use of the platform. The system supports all the mission requirements of the platform and may have the capability to program onboard platform systems with routing, targeting, performance, map, and Intel data*			
Source Definition	***Mission Planning and Analysis****: Software used to support mission planning activities such as space mission planning, aircraft mission planning, scenario generation, feasibility analysis, route planning, and image/map manipulation* ***Command and Control****: Software that allows humans to manage a dynamic situation and respond in human real time*			
Typical examples	*Scenario generation*	*Planning & Analysis*	*Target planning*	*Route planning*
	Fuel planning	*Cargo load planning*		
Training				
Definition	*Hardware and software that are used for educational and training purposes*			
Source Definition	***Training/CBT/CAI:*** *An application that delivers education or training*			
Typical examples	*Computer Based Training (CBT)*	*Computer Aided Instruction (CAI)*	*Tutorial Applications*	*Courseware*

Software Tools				
Definition	*Software that is used for analysis, design, construction, or testing of computer programs*			
Source Definitions	***Testing Software**: Software for testing and evaluating hardware and software systems* ***Software Development Tools**: Software used for analysis, design, construction, or testing of computer programs*			
Typical examples	*Compilers*	*Linker/loaders*	*Debuggers*	*Editors*
	Assemblers	*Requirements analysis & design tool aids*	*Code generators*	*Programming aids*
	Report generators	*Code auditors*	*Test case data recording*	*Test case data reduction/analysis*
	Test case generation			
Enterprise Service Systems				
Definition	*Software needed for developing functionality or a software service that are unassociated, loosely coupled units of functionality that have no calls to each other embedded in them .*			
Source Definition	*Software needed for developing functionality or a software service that are unassociated, loosely coupled units of functionality that have no calls to each other embedded in them . COTS/GOTS services that are unassociated, loosely coupled units of functionality.* ***Electronic Data Exchange**: An application specialized in the structured transmission of business data or documents between separate parties (companies, organizations, etc.) without human intervention*			
Typical examples	*Enterprise service management*	*Machine-to-machine messaging*	*Service discovery*	*People and device discovery*
	Metadata discovery	*Mediation service*	*Service security*	*Content discovery and delivery*
	Federated search	*Enterprise catalog service*	*Data source integration*	*Enterprise content delivery network*
	Session management	*Presence and awareness*	*Text collaboration*	*White boarding and annotation*
	Application sharing	*Application broadcasting*	*Virtual spaces*	*Identity management*
	Content discovery	*Collaboration*	*User profiling and customization*	
Custom AIS Software				
Definition	*Software needed to build a custom software application to fill a capability gap not captured by COTS/GOTS software packages*			
Source Definition	***Graphical User Interface**: A general class of applications using windows, icons, menus and a pointing device, which are developed using standard features of a modern integrated development environment (IDE)* ***Multimedia**: An application that achieves enhanced user interaction by going beyond standard computing interfaces, for example, using graphics and input devices in ways that require custom programming* ***Internet Server Applet**: Platform-independent software which executes in the browser, typically JavaScript and its libraries. This may also include server-side scripting for example using PHP*			

Typical examples	*Glue code*	*External system interfaces*	*Data transformation*	*Inter-COTS/GOTS data exchange*
	Graphical User Interface	*Internet Server Applet*	*Website*	
Enterprise Information Systems				
Definition	*Software needed for building an enterprise information system that uses an integrated database to support typical business processes within business/functional areas and consistent information access across areas and systems. COTS/GOTS attributed to a specific software service or bundle of services*			
Source Definition	*See SEER-SEM Enterprise Information Systems Definition* *See QSM Enterprise Information Systems Definitions*			
Typical examples	*Enterprise resource planning*	*Enterprise data warehouse*	*General ledger*	*Accounts payable*
	Revenue and accounts receivable	*Funds control and budgetary accounting*	*Cost management*	*Financial reporting*
	Real property inventory and management	*Document management*	*Logistic or Supply Planning & Control*	*Transaction Processing*
	Management Performance Reporting	*Office Information System*	*Reservation System*	*Geographic or spatial information system*
	Financial Transactions	*Database management*	*Data Warehousing*	*Executive Information System*
	Internet Server Applet	*Report Generation*	*Office Automation*	*Data Mining*

Appendix B – Productivity Quality Tags:

Good – This is a data point that is complete for both hours and SLOC and has correct demographic, reporting event, personnel, and AD information. It also is not a TD or EMD effort (in other words the data point represents the totality of the software effort and does not have the artificial split created by TD/EMD), did not require an allocation of hours associated with support elements like CM, QA, SW Program Management or integration, or did not require combining build or phase information to make the data point complete.

Good – Allocation. This is a data point that meets the criteria of good, but it has allocated hours associated with it to distribute things like QA, CM, SW PM and integration that were reported at the total effort level back to the lowest level CSCIs or WBS.

Good – Combined. This is a data point that the government has created by combining build information if the data shows that the builds were not independent standalone events and that combining them will create a higher quality data point. It may also include the combination of TD and EMD reports. The information must pass the standards established for a Good data point in order to be combined.

Good – TD Proto. This is a final from the Technology Development phase of development and is for the code that is delivered as part of the TD. The reported hour information can be missing large portions of IEEE 12207 functions as the purpose of this code is only to demonstrate a TD function on the Concept Demonstrator (CD) and it will not continue on into EMD. Typically they will identify a size for the TD code. A final report for TD prototype code will almost always be tagged as good.

Good – TD Proto Allocation. This is a final from the Technology Development stage and is for code delivered as part of TD but required the allocation of support functions to make the data point good

Good – TD EMD. This is a final from the Technology Development phase that is effort performed on code that will be developed during the EMD phase of the program. Typically this will capture upfront design and architecture activities associated with getting to a Preliminary Design Review (PDR) level of understanding. Frequently this data point will not have SLOC associated with it and must be combined with a follow on EMD effort to make a complete data point.

Good – EMD. This is a final from the EMD phase of the program. It passes all of the quality screens associated with a Good data point; however it may be missing activities performed during the TD phase of the program. More often than not this data point will need to be combined with its TD phase for inclusion in productivity analysis.

Roll Up – Lower Level Good. This is a data point that is the summation, or at times duplication, of lower level WBS or CSCI reports and is comprised of efforts tagged as good (it may include reports that are allocation activities like CM, QA, or SW PM) . Inclusion of this data point in analysis would “double count” the effort when analyzing productivity.

Roll Up – Incomplete Lower Level – This is a schedule at a parent level where it is determined that the data feeding the parent was not flagged as good, and at the roll up the data is determined to not be good as well. This can occur for interim build information, terminated contracts, missing SLOC or effort data or any of the other reasons to potentially not use a data point

Roll Up – TD. This is a roll up for TD data points.

Roll Up – EMD. This is a Roll Up for the EMD phase and follows the rules of an overall Roll Up.

Interim Build – This is a data point that is an interim update on the effort and does not represent the final delivered code or hours. It can be associated both with a build or a programmatic event like PDR or CDR if reporting is required for them.

Missing SLOC. This is a final data point that did not report SLOC with the record. Typically this will be for effort associated with support activities or integration that are global to the effort and require allocation of the hours back to other CSCI or WBS elements.

Missing Hours. This is a data point that may have reported sizing information, but did not record hours at the reported level. These data points may be useful for sizing assistance.

Question – Size. This is a data point that has reported good hours, but information on size has anomalies that make the data point questionable for usage in analysis.

TD Missing SLOC. This is a TD data point that did not include SLOC. Typically this will be associated with a testing or support function for the TD specific code.

TD – Question Hours. This is a TD data point that included SLOC, but the hours that are reported appear to be missing significant content or do not make sense for the reported SLOC.

Question – Hours. This is a data point where the reported hours cannot be verified as inclusive of all the desired IEEE 12207 activities. Overall productivity and other quality reviews indicate the data point may be valid, but questions exist about content.

Terminated Contract. This is a report where the contract was canceled prior to delivery of the final software product.

Other. This is a report that has issues not categorized above. It will contain an explanatory note as to why it did not fit into the other categories.

Appendix C – Schedule Quality Tags:

Good – This is a data point where the productivity quality tag is marked as good and the schedule is clearly unique to that CSCI or WBS (the schedule does not duplicate schedules used for other CSCIs or WBS elements in the submittal). The total duration (not the durations of the sub-elements like design or requirements) falls within established benchmarks for similar size efforts.

Good – Combined –This is a data point that the government has created by combining build information or by adding TD data with EMD data in order to create a complete, or higher quality data point. In the case of schedule the durations for the build or phase are combined to establish a total duration for the effort.

Good – TD Proto. This is the schedule associated with developing and delivering prototype code during a TD phase. Because TD tends to be a fixed duration time span and all of the prototype code is developed in this time span there tends to be an identical schedule for all of the CSCIs or WBS elements in this phase.

Good – TD EMD. This is the schedule associated with software effort that will transition from TD into the EMD phase and will tend to match the corresponding TD EMD productivity flag. This is not a complete schedule to completion of the software but will typically only cover the requirements and design stage.

Good – EMD. This is the schedule associated with completing the software started in TD. In order to be flagged as good it needs to have a unique schedule for the CSCI or WBS element being examined during the EMD phase and the productivity flag must be good (any variation) as well.

Good – Roll Up. There are times when the only schedule that can be evaluated for schedule analysis is at the roll up level. This occurs when the schedule for all of the CSCIs or WBS elements are identical (or near identical), or if there is a separate WBS that reports integration activities and the schedule for that work cannot be traced back to the lower level elements (even though it may be possible to allocate the hours back). At this point the overall schedule reported at the roll-up level becomes the point that can be used (this is even if the productivity flag has been marked as a roll-up).

Roll Up – Lower Level Good. This is a schedule at a parent level where it is determined that the lower level schedules can be used for schedule analysis.

Roll Up – Incomplete Lower Level – This is a schedule at a parent level where it is determined that the data feeding the parent was not flagged as good, and at the roll up the data is determined to not be good as well. This can occur for interim build information, terminated contracts, missing SLOC or effort data or any of the other reasons to potentially not use a data point.

Not Usable – This is a schedule that is associated with any of the productivity flags that indicate the data is incomplete or missing. These include Interim Build, Missing SLOC, Missing hours, Question Size

Appendix D – SRDR Scorecard Process and Rating(s):

The criteria described within this appendix provides reviewing Government analysts with rating designations that will be included for each SRDR data submission (2630-2, 2630-3, DID). During the SRDR review process, each CSCI submission will be evaluated to validate that the submitted data aligns within one of the four scorecard rankings described below:

- Met: SRDR provides complete evidence that satisfies the entire criterion
- Partially Met: SRDR has some issues but the supplemental data allows for adjustments or normalization
- Not Met: SRDR fails to satisfy the criterion and supplement data does not provide information for adjustments or normalization
- No Data: Criterion is not applicable

It is important to note that the scorecard tag rankings are determined by the reviewing Government analyst and are specific to the Data Homogeneity and Data Reliability areas highlighted in Table 4 below. As the SRDR submission process incorporates more automation specific to evaluating data consistency, formatting, and missing content, the qualitative data review responsibility will fall to the designated Government review analyst. This process will include a review of each area in order to determine if the criteria are Met, Partially Met, or Not Met using the color code illustrated in the tables below.

Government Analyst Data Cleansing Scorecard and Criteria[3]:

Table 4: Characteristics

Characteristic	Description
Key Grouping	Key groupings normalize data by similar missions, characteristics, or operating environments by cost type or work content. Products with similar mission applications have similar characteristics and traits, as do products with similar operating environments. For example, space systems exhibit characteristics different from those of submarines, but the space shuttle has characteristics distinct from those of a satellite even though they may share common features. Data should also be grouped by company and location, to assist decision makers during source selection and special studies.
Data Applicability	Because cost estimates are usually developed with data from past programs, it is important to examine whether the historical data apply to the program being estimated. Over time, modifications may have changed the historical program so that it is no longer similar to the new program. For example, it does not make sense to use data from a technology demonstration prototype that omitted 4 of 6 software activities to estimate a full scale development project. Having good descriptive requirements of the data is imperative in determining whether the data available apply to what is being estimated.
Data Homogeneity	Using homogeneous groups normalizes for differences between historical and new software programs in order to achieve content consistency. To that end, DCARC and CIPT must judge the SRDR data for quality in terms of consistency. This may require adding and deleting certain items to get an apples-to-apples comparison. A properly defined DID is necessary to avoid inconsistencies. The main point is to clearly define what the sizing metric is so that the data can be converted to a common standard before being used in the estimate.
Data Reliability	All data collection activities must be documented as to source, work product content, time, units, and assessment of accuracy and reliability. Comprehensive documentation during data collection greatly improves quality and reduces subsequent effort in developing and documenting the estimate. For example, the auditor should review all critical data fields in the SRDR to determine whether there are any data anomalies. The auditor should also perform a sanity check to see if the productivity data even make sense by comparing it to historical data.

[3] SRDR Scorecard Characteristics and Criteria (Rosa, 2014)

Effort

Definition: Analysts should perform a sanity check to see if the effort data makes sense in terms of consistency, content and segregation. The analyst should review all effort related fields in the DD STD Form and DID to determine whether there are any content issues:

Condition(s)	Rating	Resolution
Effort data met all checklist items: ❑ Effort data captures **all** software activities as per the DID (effort may or may not be allocated down to the activity level) ❑ The final report is the last final (not an interim report). The initial report is the first initial for the build or effort being reported ❑ Effort captures hours for the entire team – prime and subcontractor	Met	Appropriate for effort estimation
Effort data showed one of the following issues: ❑ Effort data missing for one or two software activity ❑ Effort data reported in another CSCI. For example, FQT captured in another CSCI ❑ Effort can be made complete by allocating or combining techniques ❑ Effort captures hours for the entire team – prime and subcontractor ❑ Effort is an interim report that contains a mix of actual and estimate to complete but meets all other criteria for "Met"	Partially Met	Do not use for effort distribution unless certain activities have been added or deleted to get apples-to-apples
❑ Effort data not reported for three or more software activities ❑ Effort reported as cumulative from previous builds and cannot be made standalone ❑ Effort not reported for entire team – missing subcontractor or other effort ❑ Effort is reported as rounded numbers ❑ Two or more reports at the lowest level have identical effort	Not Met	Do not use for effort related analyses Do not use for effort estimation unless: (1) cumulative data have been normalized to actual work performed;

Schedule

Definition: Analysts should perform a sanity check to see if the schedule data makes sense in terms of consistency, content and segregation. The analyst should review all schedule related fields in the DD STD Form and DID to determine whether there are any content issues:

Condition(s)	Rating	Resolution
Schedule content met all checklist items: ❑ Schedule data captures **all** software activities as per the DID ❑ Schedule data reported in calendar dates or duration with a calendar date for the first month ❑ Schedule data is for the build or effort only and not cumulative from a prior event ❑ Schedule data appears unique to the CSCI or level being looked at	Met	Use for schedule estimation
Schedule data showed one or more of the following issues: ❑ Missing schedule for one software activity ❑ Missing schedule for one activity but captured in another CSCI, e.g., requirements and the schedule can be adjusted for it ❑ Missing start month for duration schedule	Partially Met	Do not use for schedule estimation unless certain items have been added or deleted to make the schedule whole
❑ Schedule data missing on two or more software activities ❑ Schedule reported as cumulative from previous builds ❑ Replicated schedule data (start and end dates) across components at the lowest levels – could be green at a higher level	Not Met	Do not use for schedule estimation unless replicated schedule data across components, have been normalized. That is, rolling up all CSCIs into a single record.

Size

Definition: Analysts should perform a sanity check to see if the reported SIZE data makes sense in terms of type, content and segregation. The analyst should review all schedule related fields in the DD STD Form and DID to determine whether there are any content issues:

Condition(s)	Rating	Resolution
Size reported met all checklist items: ❑ Size appropriately allocated by types (New, Modified, Reuse, Auto-Generated, Deleted) according to DID and SRDR guidelines ❑ Hand-coded and Auto-generated code shown separately ❑ Reuse code excludes COTS ❑ Carried over code has been properly identified. ❑ Deleted code has been properly identified ❑ Preexisting code split between Reuse and Modified ❑ Requirements and External Interfaces count provided ❑ Covers both prime and subcontractor effort if applicable ❑ Size is logical	Met	
❑ Size reported is non-comment or physical ❑ Discrepancy between form and dictionary on how code is reported ❑ All other criteria met	Partially Met	Determine if adjustment is necessary
Size reported did not meet the checklist requirements	Not Met	Do not use unless the DID provides information for normalization

Equivalent Size

Definition: Analysts should review all reported ESLOC parameters in the SRDR to determine whether there are any data anomalies or discrepancies.

Condition(s)	**Rating**	**Resolution**
Reported ESLOC data is reliable as it met all checklist items: ❑ ESLOC parameters (DM, CM, IM) reported for all size types (modified, reuse, etc.) ❑ ESLOC parameters, CM and IM, greater than 0 for Modified Code ❑ ESLOC parameters, DM and CM, equal to 0 for Reuse Code ❑ No discrepancies between reported ESLOC parameters and DID ❑ Developer provided their own ESLOC parameters and step-by-step calculation to allow analysts to replicate	Met	
Reported ESLOC data is NOT reliable as it failed one or more checklist items: ❑ ESLOC parameters (DM, CM, IM) NOT reported on one or more size type ❑ ESLOC parameters, CM or IM, equal to 0 for Modified Code ❑ ESLOC parameters, DM or CM, greater than 0 for Reuse Code ❑ Discrepancies between reported ESLOC parameters and DID ❑ Developer provided their own ESLOC parameters but failed to provide step-by-step calculation	Not Met	Correct submission prior to acceptance

DM = Percent Design Modified, CM = Percent Code Modified, IM = Percent Re-Test