

**CLEARED  
For Open Publication**

**Mar 20, 2018**

Department of Defense  
OFFICE OF PREPUBLICATION AND SECURITY REVIEW

**DEPARTMENT OF DEFENSE**  
**SOFTWARE RESOURCE DATA REPORT (SRDR)**  
**VERIFICATION AND VALIDATION (V&V) GUIDE**  
**VERSION 4.0**



**02 February 2018**

**Primary Authors:**

**Nicholas Lanham, OSD CAPE**

**Marc Russo, NCCA**

**Dan Strickland, MDA**

**Ron Cipressi, AFCAA**

**Stephen Palmer, NAVAIR 4.2**

**Crystal Rudloff, Technomics**

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.

**18-S-1070**

<b>Software Resource Data Report (SRDR) Unified Review Function (SURF) Team Charter .....</b>	<b>1</b>
<b>Verification and Validation (V&amp;V) Guide Purpose.....</b>	<b>2</b>
<b>Summary of V&amp;V Guide Changes .....</b>	<b>3</b>
1.0 Review of an SRDR submitted to DCARC .....	4
1.1. Question Template Common Heading and Summary Data.....	4
1.2. Legacy SRDR Reports .....	4
1.3. SRDR Development Reports .....	5
1.4. SRDR Maintenance Reports .....	5
1.5. SRDR ERP Reports .....	6
1.6. Additional Review Guidelines .....	6
1.6.1. Super Domain (SD) and Application Domain (AD) Designation: .....	6
1.6.2. Operating Environment (OE) Designation: .....	7
1.6.3. Effort .....	8
1.6.4. Schedule .....	9
1.6.5. Estimate at Completion (EAC) Values .....	9
1.6.6. CSDR Plan Review .....	9
2.0 Quality Tagging .....	10
3.0 Solutions for Common Findings .....	10
3.1. Allocation.....	10
3.2. Combining.....	11
3.3. Early Acquisition Phase Combining .....	12
4.0 Pairing Data .....	12
5.0 Possible Automation .....	12
6.0 Process Improvement Initiatives .....	13
6.1. Comments .....	13
6.2. Data Collection on the SURF Process .....	13
6.3. Process Review Meetings .....	13
Appendix A – Super Domain and Application Domain Categories .....	15
Appendix B – Productivity Quality Tags.....	24
Appendix C – Schedule Quality Tags.....	26
Appendix D – CSDR Plans in CADE.....	27
Appendix E – Future V&V Questions .....	29

## **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. Previously, the primary Government review occurred after final DCARC acceptance. This adjustment to the process significantly enhances the Government's review effort and reduces the amount of Government and contractor processing time.

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, Cost Assessment and Program Evaluation (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.

The establishment of a joint SURF team that includes members from each DoD service cost organization and allows for Government review prior to final DCARC acceptance, provides 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. The SURF team will manage and update the existing SRDR dataset hosted to the Defense Acquisition Automated Cost Information Management System (DACIMS) application in the Cost Assessment Data Enterprise (CADE) 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. In addition to early and consistent reviews, the SURF team will work with the CADE and DCARC teams to capture key metrics and lessons learned to continue to improve the review of SRDR submissions.

## **Verification and Validation (V&V) Guide Purpose**

The purpose of this SRDR data review guide is to provide general directions, key 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 DCARC submission process. These submissions undergo a rigorous 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 data and information. 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, a set of SRDR V&V Question Templates have been developed to highlight data inconsistencies, erroneous values, and/or substantial deviations from historical reports (i.e., prior software builds and/or increments). Without following V&V guidance, the 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 and the accompanying V&V Question Lists provide 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:

- 1.0 Review of an SRDR submitted to DCARC
- 2.0 Quality Tagging
- 3.0 Solutions for Common Findings
- 4.0 Pairing Data
- 5.0 Possible Automation
- 6.0 Process Improvement Initiatives

## **Summary of V&V Guide Changes**

The following list summarizes the key additions and updates that were made to the SRDR V&V Guide in Version 4.0:

- Creation of V&V Question Templates to support the review and evaluation of current and legacy Development, Maintenance, and Enterprise Resource Planning (ERP) SRDR submissions
- Edits and updates to the V&V Guide based on lessons learned, process changes, and policy updates (e.g., 2017 SRDR Data Item Description (DID) update)
- Cost and Software Data Reporting (CSDR) Plan review guidance and instructions
- Identification of SURF process improvement initiatives

## 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 SURF V&V Question Templates were created to aid SURF team analysts in their review of the submitted data. Initially, a single V&V Question Template was created to support the review of all software reports, however, with the recent updates to the SRDR DID and the creation of separate Development, Maintenance and ERP reporting forms, it was determined that templates should be created for each of the different reporting types. The SURF V&V Question Templates are available on the CADE public website ([www.cade.osd.mil/policy/software](http://www.cade.osd.mil/policy/software)). The templates also include standardized responses that can be submitted to DCARC with the completed review. The templates have been designed such that every “no” response represents data that is missing or data that requires further explanation from the contractor. For every “no” response, SURF team analysts should include specific details that are expected to be included in the updated data submission. This information can be incorporated into the template by editing the response or using the comment field. This guide and the questions in the templates are to be utilized at the lowest level Computer Software Configuration Item (CSCI) or Contractor Work Breakdown Structure (CWBS) element provided. In addition to the completion of the question template for each field, Release, and CSCI, the SURF reviewer should provide a summary assessment of the report and the required data and information required for an acceptance of the submission. Additionally, feedback from DCARC indicates that the most helpful reviews include additional detail within the comments section for each question. SURF analysts are encouraged to comment within the question template to provide DCARC with further insight to aid them in their discussion with the contractor.

### 1.1. Question Template Common Heading and Summary Data

Each review using the question template requires that the SURF analyst provide the following common heading and summary data:

- Contractor Name – Provide the name of the contractor that submitted the software report.
- Contract Number – Provide the contract number of the software report that is being evaluated.
- DCARC Identification (ID) Number – Provide the DCARC ID number.
- SURF Analyst Name – Provide the name of the analyst that is completing the review of the software report.
- SURF Analyst Email – Provide the email of the analyst that is completing the review of the software report.
- SURF Analyst Recommendation – Provide a summary of the overall recommendation and assess if the data submission should be accepted or rejected by DCARC. If a rejection is recommended, provide a summary of the data and information that is required in the resubmittal.
- Super Domain and Application Domain– Provide the SURF team analyst’s assessment of the super domain and application domain. Section 1.6.1 provide additional guidance and definitions associated with selecting a super domain.
- Operational Environment – Provide the SURF team analyst’s assessment of the operational environment of the software. Section 1.6.2 provides additional guidance and definitions.

### 1.2. Legacy SRDR Reports

When evaluating submissions that are based on Legacy requirements (i.e., 2011 SRDR DID), analysts should use the SURF Legacy Question Template for the review. The template is organized as follows:

- Reporting Event
- Demographic and Common Heading Information
- Software Characterization and Development Process
- Super Domain and Application Domain Designation
- Operating Environment (OE) Designation
- Development Process
- Personnel
- Sizing and Language
- Requirements
- Source Lines of Code (SLOC)
- Non-SLOC Based Software Sizing
- Product Quality Reporting
- Effort
- Schedule

### 1.3. SRDR Development Reports

When evaluating submissions which use the DD Form 3026-1, analysts should use the SURF Development Question Template for the review. The template is organized as follows:

- Common Heading
- Release-Level – the questions included in this section should be completed for each reported release
- CSCI-Level – the questions included in this section should be completed for each reported CSCI
- Source Lines of Code (SLOC)
- Agile Software Development Sizing
- Non-SLOC Based Software Sizing
- Product Quality Reporting
- Effort
- Schedule
- Estimate at Complete (EAC)

### 1.4. SRDR Maintenance Reports

When evaluating submissions that are based on the DD Form 3026-2, analysts should use the SURF Maintenance Question Template for the review. The template is organized as follows:

- Common Heading
- Resource Reporting
- Software Maintenance Technical Data
- Release-Level Data – the questions included in this section should be completed for each reported release
- Labor Hours
- Product Size Reporting
- Software Maintenance Effort Data
- Schedule
- Estimate at Complete (EAC)

### 1.5. SRDR ERP Reports

When evaluating submissions that are based on the DD Form 3026-3, analysts should use the SURF ERP Question Template for the review. The question list is organized as follows:

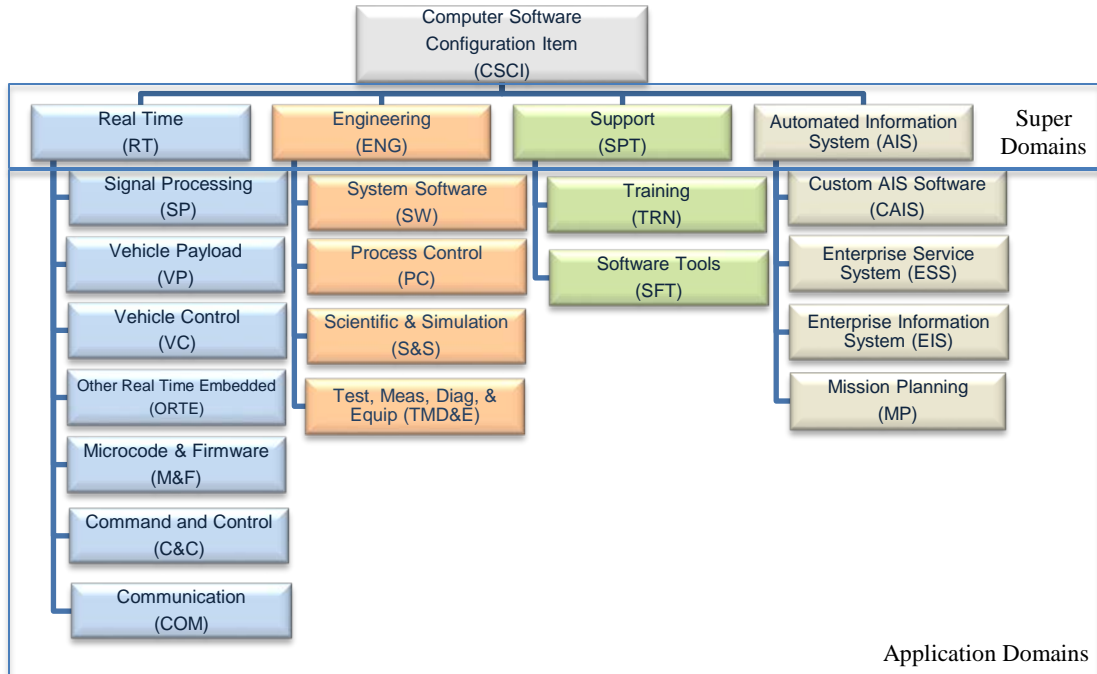
- Common Heading
- ERP Software Development Technical Data (Project)
- Application Domain Designation
- Product Size Reporting
- Agile Measures
- Implementation
- ERP Activities with Effort and Schedule

### 1.6. Additional Review Guidelines

#### 1.6.1. Super Domain (SD) and Application Domain (AD) Designation:

When assessing the AD designation chosen by the submitting organization, the reviewing Government analyst must begin by tracing the given CSCI into an SD category (Real Time, Engineering, Support, or Automated Information System) using the mapping tree illustrated within Figure 1. During this process, the reviewing analyst must be sure to review the SRDR data dictionary (if a Legacy form) 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 – Super Domain and Application Domain Categories.





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

After the analyst has reviewed the supporting functional description, 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- Super Domain and Application Domain Categories.

Another method of mapping a CSCI to SD and AD category is to analyze the Work Breakdown Structure (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 – Super Domain and Application Domain Categories. Once the analyst has determined what SD and AD categories the CSCI falls under, they should note it within the question template. These designations will assist the SRDR database team in capturing all relevant data.

#### 1.6.2. Operating Environment (OE) Designation:

The latest version of the SRDR DID requires the submitting organization to select the appropriate operating environment specific to the system. In order to confirm the most applicable OE has been selected, the reviewing Government analyst must compare the system description included in the form, SRDR data dictionary (for Legacy reports), 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
<b>Surface Fixed (SF)</b>	Fixed (SF)	software is at a fixed site

Operating Environment (OE)		Examples
<b>Surface Mobile (SM)</b>	Manned (SMM)	software is moved somewhere and set up
	Unmanned (SMU)	
<b>Surface Portable (SP)</b>	Manned (SPM)	software is in a handheld device
<b>Surface Vehicle (SV)</b>	Unmanned (SVM)	software is embedded in as part of a moving ground vehicle
	Unmanned (SVU)	
<b>Air Vehicle (AV)</b>	Manned (AVM)	software is embedded as part of an aircraft
	Unmanned (AVU)	
<b>Sea System (SS)</b>	Manned (SSM)	software is embedded as part of a surface or underwater boat/ship or boat
	Unmanned (SSU)	
<b>Missile System (MS)</b>	Unmanned (MSU)	Software is embedded as part of a missile system
<b>Ordnance System (OS)</b>	Unmanned (OSU)	software is embedded as part of an ordnance system
<b>Space System (SPS)</b>	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 against the list of OE descriptions in Table 1. The functional description is included as part of the SRDR data dictionary for Legacy forms or in the Release-level data in the Development, Maintenance and ERP forms. Each of the provided OE designations has been derived from the list of “major systems” or “end products” listed within MIL-STD-881C<sup>1</sup>, 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 certain cases, such as a radio that is handheld and used within a weapon systems, a CSCI may have two OE designations. In that case, the analyst should capture the OE for both uses and note why they made that selection within the comment section of SURF question template.

### 1.6.3. Effort

<sup>1</sup> “Work Breakdown Structures for Defense Materiel Items,” MIL-STD-881C, 3 October 2011.

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.”

#### 1.6.4. 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.

#### 1.6.5. 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) on all initial and interim reports. These fields will provide Government analysis teams with the ability to trace estimated hours up to the parent WBS element. Effort data is summarized monthly and includes Actuals To Date (ATD), Estimate At Complete Hours (EAC<sub>H</sub>), and Total Hours.

#### 1.6.6. CSDR Plan Review

The CSDR Plan provides the contractor, the DCARC, and the SURF team with detailed information regarding the structure and frequency of the data that is expected to be reported in each SRDR submission. The CSDR Plan details the Work Breakdown Structure (WBS) to be

used to report the costs associated with the software development or maintenance activities. Because each Release and CSCI is defined by the contractor, a CSDR Plan Supplement has been created to help track the names, numbers, start dates, and end dates associated with each Release and each CSCI. For development reports, the CSDR Plan Supplement also includes a mapping between the Releases and the CSCIs. This data is used to ensure that software data is appropriately tracked and reported. As a result, the SURF analyst should review each SRDR submission to ensure that it includes all of the data that was requested by the CSDR Plan. Appendix D – CSDR Plans in CADE includes instructions on how to access CSDR Plans in CADE

## 2.0 Quality Tagging

The purpose of conducting an SRDR V&V review is to determine if the data point(s) can be used when performing software data analysis or building future estimates and cost models. Historically, only a relatively small percentage of the submissions fall into the category of being “Good” data 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, a roll-up, 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 – Productivity Quality Tags provides Government analysts with a list of ways that data can be tagged for productivity (size and effort information) and Appendix C – Schedule Quality Tags for schedule usage.

## 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 Technology Demonstration (TD) portion with the subsequent Engineering Manufacturing and Development (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 for the Management and Support hours 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  $(23,900 / 218,085) = 10.96\%$  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 descriptions 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 3.1 and 3.2, respectively), the same process must have occurred on the Initial as well the Final.

### 5.0 Possible Automation

The establishment of a standardized Extensible Markup Language (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, the funding to implement it, and the prioritization of this work within other CADE efforts. 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 data have been included for all applicable WBS elements, Releases and CSCIs as required for SRDR reporting by the CSDR Plan
- 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 Full-Time Equivalent(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

## 6.0 Process Improvement Initiatives

As the automation and CADE tools that support the SURF process mature, evolve and become available, the team will need to reevaluate the V&V Question Templates and the supporting business processes. Because there are no validation tools today, much of the V&V Question Templates have been focused on ensuring that the appropriate fields include data and the submissions are consistent with the CSDR Plan. In the future, the tool should be able to perform these checks, thereby allowing the SURF to focus more on the completeness and the technical credibility of the data. For example, the team can look at the data reported across submissions (initial, interim, final) to ensure consistency, as well as, compare the data reported to known historical benchmarks. Appendix D – Future V&V Questions includes questions that may be added to the V&V guide and question templates when some level of automation is available in CADE. In addition, as SURF reviews become less of a manual process, the analysts should continue to review all submissions (i.e. initial and resubmissions) and provide feedback for all of the iterations throughout the reporting life of the program. The SRDR DID and its forms have been updated and some of the reporting requirements are new to the community. For these reasons, the SURF team has implemented several process improvement initiatives to help address any lessons learned and identify issues with current documentation and policy. These initiatives include SURF reviewer comments, SURF process data collection, and annual process review meetings.

### 6.1. Comments

In an effort to provide clear and actionable guidance to the DCARC and the contractor, the V&V Question Templates include a section for overall comments that, when filled out by the SURF analyst, include an overall evaluation of the report, as well as, information regarding what needs to be clarified or included in order for the report to be accepted. Also, for responses where data is missing, the Yes or No responses are acceptable, but where more clarification is required because the data reported does not make sense, the SURF analyst should provide more specific comments to state what is needed in the report. These efforts combined should decrease the number of resubmissions required for SRDRs.

### 6.2. Data Collection on the SURF Process

The team also plans to use CADE to help capture data and information regarding the questions within the templates that have the most negative responses. Over time, these questions may help identify any issues with the V&V Question Templates, the DID or the forms that may be causing confusion for the reporting contractor. As this information is known and identified, the SURF team will evaluate and provide recommendations if the language in the V&V Question Template and/or the DID and its forms need to be updated. All recommendations for updates will be vetted through the SURF team, the DCARC, and the SRDR Working Group.

### 6.3. Process Review Meetings

In addition to the monthly email correspondence, the SURF team members and key DCARC representatives plan to meet yearly to discuss key findings, issues, and lessons learned. Because the SRDR DID and its forms have only recently been approved, any needed changes or improvements to

the guide should be tracked and discussed amongst SURF team members. As issues arise, the SURF team lead, will capture the information in a watch list and determine if and when updates are recommended and required. Any information available from the data collection initiative or feedback received by DCARC will also be addressed in this forum.



## Appendix A – Super Domain and Application Domain Categories

Super Domain (SD) Descriptions and Categories:	
<b><i>Real Time (RT)</i></b>	
Description	<p>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.,</p> <ul style="list-style-type: none"> <li>• May have guaranteed execution requirements i.e. missed deadline means catastrophic results</li> <li>• May have to be compact and efficient due to limited storage capacity and high throughput requirements</li> <li>• Could have very high reliability requirements (life critical, manned mission)</li> <li>• Might have tightly coupled interfaces</li> <li>• Program code may be imprinted on hardware devices</li> <li>• May process sensor inputs and directs actuator outputs</li> <li>• Sometimes executed on special-purpose processors</li> </ul>
Application Domain Mapping	Signal Processing, Vehicle Control, Vehicle Payload, Other Real Time Embedded, Command and Control, Communication, and Microcode & Firmware
<b><i>Engineering (ENG)</i></b>	
Definition	<p>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.,</p> <ul style="list-style-type: none"> <li>• May have a fast response time requirement</li> <li>• May have more storage capacity</li> <li>• Might need to be highly reliable but not life critical</li> <li>• May have multiple interfaces with other systems</li> <li>• May implement complex algorithms, models or protocol</li> <li>• Program code can be modified or uploaded</li> <li>• Executes on general purpose processors that may be embedded in special purpose hardware</li> </ul>
Application Domain Mapping	System Software, Process Control, Scientific and Simulation, and Test, Measurement, and Diagnostic Equipment
<b><i>Automated Information System (AIS)</i></b>	
Definition	<p>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.,</p> <ul style="list-style-type: none"> <li>• Must have acceptable response time</li> <li>• Fewer storage and throughput constraints</li> <li>• Must be reliable enough to prevent data loss</li> <li>• May consist of a single COTS / GOTS solution or multiple products coordinated with customer software</li> <li>• Algorithms, models, and protocols are well understood</li> <li>• Code may not be available for modification</li> <li>• Software restarts are acceptable</li> <li>• Executes on commercial processing hardware</li> </ul>
Application Domain Mapping	Mission Planning, Enterprise Service Systems, Custom AIS Software, and Enterprise Information Systems

<b>Support (SPT)</b>	
Definition	<p>Support software assists with operator training and software testing. This software has few constraints, e.g.,</p> <ul style="list-style-type: none"> <li>• Has to have an acceptable response time most of the time</li> <li>• Less limited by storage or throughput</li> <li>• Less stringent reliability requirement</li> <li>• Software restarts are acceptable</li> <li>• Fewer interfaces</li> <li>• Relatively low complexity algorithms, models or protocols</li> <li>• Program code can be modified and uploaded</li> <li>• Executes on general purpose processors on general purpose computer boards</li> </ul>
Application Domain Mapping	Training, and Software Tools

Application Domain Definitions and Categories:				
<b>Signal Processing</b>				
<i>Definition</i>	<i>Software that requires timing-dependent device coding to enhance, transform, filter, convert, or compress data signals</i>			
<i>Source Definition</i>	<b>Signal Processing:</b> <i>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</i>			
<i>Typical examples</i>	<i>Lasers</i>	<i>Sonar</i>	<i>Acoustic</i>	<i>Electromagnetic</i>
	<i>Signal Processor</i>	<i>Radar Altimeter</i>	<i>Photographic Sensors</i>	<i>Motion Sensors</i>
	<i>Infrared Sensors</i>	<i>Sensor Assembly</i>	<i>Electronic Sensors</i>	<i>Seeker Assembly</i>
	<i>Signal Electronics</i>	<i>Optical Assembly</i>	<i>Tracking Sensors</i>	<i>Antenna Assembly</i>
<b>Vehicle Payload</b>				
<i>Definition</i>	<i>Software which controls and monitors vehicle payloads and provides communications to other vehicle subsystems and payloads</i>			
<i>Source Definition</i>	<b>Vehicle Payload:</b> <i>Software used to manage and control payload functions (experiment control, sensor management, etc.) for manned or unmanned space applications</i>			
<i>Typical examples</i>	<i>Fire Control</i>	<i>Mine Warfare</i>	<i>Electronic Attack subsystem controller</i>	<i>Weapons Delivery and Control</i>
	<i>Gun fire control system</i>	<i>Missile fire control systems</i>	<i>Antisubmarine warfare fire control and torpedo fire control systems</i>	<i>Pointing, Command, &amp; Control Interface</i>
	<i>Payload Flight Software</i>	<i>Armament</i>	<i>Survivability Payload</i>	<i>Reconnaissance Payload</i>
	<i>Electronic Warfare Payload</i>	<i>Armament/Weapons Delivery</i>	<i>Intelligence, Surveillance, Reconnaissance Payload</i>	<i>Mission Payload</i>
<b>Vehicle Control</b>				
<i>Definition</i>	<i>Software necessary for the control of vehicle primary and secondary mechanical devices and surfaces</i>			
<i>Source Definition</i>	<b>Flight Systems:</b> <i>Onboard software used for various functions associated with the operation and control of airborne platforms (e.g., airplanes, helicopters, missiles, and spacecraft)</i> <b>Avionic:</b> <i>Software that is on-board &amp; 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:</i>			
<i>Typical examples</i>	<i>Flight Control</i>	<i>Electrical Power</i>	<i>Hydraulic</i>	<i>Fuel Subsystem</i>
	<i>Propulsion</i>	<i>Attitude Control System</i>	<i>Structures &amp; Mechanisms</i>	<i>Bus Flight Software</i>
	<i>Thermal Control</i>	<i>Landing Gear</i>	<i>Controls software</i>	<i>Thrust Vector Actuation</i>
	<i>Executive</i>			

<b>Other Real Time Embedded</b>				
<i>Definition</i>	<p><i>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).</i></p> <p><i>Real-time data processing unit responsible for directing and processing sensor input/output</i></p>			
<i>Source Definition</i>	<p><b>Radar:</b> <i>Software used in the operation and control of radar systems</i></p> <p><b>Embedded Electronics:</b> <i>An application that is very hardware-specific and often embedded in the firmware of electronic devices and other hardware</i></p> <p><b>Robotics:</b> <i>Software that provides logic and control for robotic or automation equipment</i></p> <p><b>Real Time:</b> <i>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</i></p>			
<i>Typical examples</i>	<i>Embedded Electronics/ Appliance</i>	<i>Robotics</i>	<i>PDAs</i>	<i>Telemetry, Tracking, &amp; Command (TT&amp;C)</i>
	<i>Guidance, Navigation and Control</i>	<i>Controls and Displays</i>	<i>Data Links</i>	<i>Radios (device)</i>
	<i>Remote Control</i>	<i>Receiver</i>	<i>Transmitter</i>	<i>Exciter</i>
	<i>Bombing Computer</i>	<i>Video and recorders</i>	<i>Telephones (device)</i>	<i>Built-in-Test</i>
<b>Microcode and Firmware</b>				
<i>Definition</i>	<p><i>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</i></p>			
<i>Source Definition</i>	<p><b>Microcode and Firmware:</b> <i>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</i></p> <p><b>Microcode and Firmware:</b> <i>“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.”</i></p>			
<i>Typical examples</i>	<i>Field Programmable Gate Arrays (FPGAs)</i>	<i>Microwave controllers</i>	<i>Application Specific Integrated Circuit (ASIC)</i>	<i>Programmable Read-Only Memory (PROM)</i>
	<i>Erasable Programmable Read- Only Memory (EPROM)</i>	<i>Electrically Erasable Programmable Read-Only Memory (EEPROM)</i>	<i>Complex Programmable Logic Device (CPLD)</i>	<i>Programmable Array Logic (PAL)</i>
	<i>Electronic Programmable Logic Device (EPLD)</i>	<i>Field Programmable Logic (FPL)</i>		

<b>Command and Control</b>				
<i>Definition</i>	<i>Software that allows humans to manage a dynamic situation and respond in human real time</i>			
<i>Source Definition</i>	<i>Vehicle onboard master data processing unit(s) responsible for coordinating and directing the major mission systems</i> <b>Command and Control:</b> <i>An application that provides commands and monitoring between users (and other systems) and hardware (or hardware-embedded software)</i> <b>Command and Control:</b> <i>Software that allows humans to manage a dynamic situation and respond in human real time</i>			
<i>Typical examples</i>	<i>Mission Management</i>	<i>Mission Computer Processing</i>	<i>Mission Control</i>	<i>Command processing</i>
	<i>Air traffic control</i>	<i>Data reduction/analysis</i>	<i>Telemetry Processing</i>	<i>Battlefield command</i>
	<i>Battle management</i>			
<b>System Software</b>				
<i>Definition</i>	<i>Layers of software that sit between the computing platform and applications [1]</i>			
<i>Source Definition</i>	<b>Device Driver:</b> <i>An application that provides low level connectivity services for a particular device (hardware or software) attached to or loaded onto a computer</i> <b>System and Device Utilities:</b> <i>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</i> <b>System:</b> <i>Layers of software that sit between the hardware and applications programs</i>			
<i>Typical examples</i>	<i>Operating Systems</i>	<i>Infrastructure</i>	<i>Framework</i>	<i>Middleware</i>
	<i>Device Driver</i>	<i>Display Drivers</i>	<i>File management</i>	<i>Image Processing</i>
	<i>Interface Driver</i>	<i>Utilities</i>		
<b>Process Control</b>				
<i>Definition</i>	<i>Software that manages the planning, scheduling and execution of a system based on inputs, generally sensor driven</i>			
<i>Source Definition</i>	<b>Process Control:</b> <i>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</i> <b>Process Control:</b> <i>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</i>			
<i>Typical examples</i>	<i>Temperature control</i>	<i>Manufacturing process control</i>	<i>Device or instrument control</i>	

<b>Scientific and Simulation</b>				
<i>Definition</i>	<i>Non real time software that involves significant computations and scientific analysis</i>			
<i>Source Definition</i>	<p><b>Artificial Intelligence:</b> Machine learning algorithms or software that often seeks to mimic human intellectual processes</p> <p><b>Computer Aided Design:</b> An application for creating, editing, and analyzing graphical models &amp; representations</p> <p><b>Expert System:</b> An application that emulates the decision-making ability of a human expert</p> <p><b>Math and Complex Algorithms:</b> An application dominated by complex mathematical operations or algorithms, numerical methods or complex analysis</p> <p><b>Simulation:</b> 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</p> <p><b>Graphics:</b> An application using custom or advanced image rendering (ray tracing, smoothing, fractals, etc.</p> <p><b>Scientific:</b> 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</p>			
<i>Typical examples</i>	<i>System Integration Lab (SIL) Simulation</i>	<i>Simulators</i>	<i>Offline Data Analysis</i>	<i>Expert Systems</i>
	<i>Math &amp; Algorithm Intensive</i>	<i>Graphics</i>	<i>Statistical Analysis</i>	<i>Artificial Intelligence</i>
	<i>Simulation &amp; Modeling</i>	<i>Engineering &amp; Science</i>	<i>3D Modeling &amp; Animation</i>	<i>Trainer Simulations</i>
		<i>Computer Aided Design (CAD)</i>	<i>Model Based Systems Engineering (MBSE)</i>	<i>Weather models</i>
<b>Communications</b>				
<i>Definition</i>	<i>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</i>			
<i>Source Definitions</i>	<p><b>Communications:</b> An application involved in the transmission and receipt of data across networks</p> <p><b>Message Switching:</b> Transport layer software performing packet and circuit switching, handling electronic mail and implementing file transfer protocols</p> <p><b>Network Management:</b> Software that monitors and reports on the status of all components of telecommunication networks, including communication links and nodes</p> <p><b>Voice Provisioning:</b> 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</p> <p><b>Telecommunication:</b> Software that facilitates the transmission of information from one physical location to another</p>			
<i>Typical examples</i>	<i>Switches</i>	<i>Routers</i>	<i>Integrated circuits</i>	<i>Multiplexing</i>
	<i>Encryption</i>	<i>Broadcasting</i>	<i>Transfer modes</i>	<i>Radios (networks)</i>

	<i>Network management</i>	<i>Network Operations</i>	<i>Satellite communications</i>	<i>Telecommunications</i>
	<i>Networks (WAN/LAN)</i>	<i>Protocols (VOIP, TCP/IP, PKI, etc.)</i>		
<b>Test, Measurement, and Diagnostic Equipment (TMDE)</b>				
<i>Definition</i>	<p><i>Software used for testing, measuring, diagnosing, emulating, and evaluating operational hardware and software systems</i></p> <p><i>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</i></p> <p><i>This does not include built-in-test (BIT)</i></p>			
<i>Source Definitions</i>	<p><b>Diagnostics:</b> <i>An application that performs a comprehensive series of built-in tests on core components and reports the results of each test</i></p> <p><b>Test and Measurement Equipment:</b> <i>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</i></p>			
<i>Typical examples</i>	<i>Test equipment software</i>	<i>Equipment emulators</i>	<i>Test driver</i>	<i>Maintenance and Diagnostic</i>
	<i>Fault Tolerance</i>	<i>Diagnostic</i>		
<b>Mission Planning</b>				
<i>Definition</i>	<p><i>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</i></p>			
<i>Source Definition</i>	<p><b>Mission Planning and Analysis:</b> <i>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</i></p> <p><b>Command and Control:</b> <i>Software that allows humans to manage a dynamic situation and respond in human real time</i></p>			
<i>Typical examples</i>	<i>Scenario generation</i>	<i>Planning &amp; Analysis</i>	<i>Target planning</i>	<i>Route planning</i>
	<i>Fuel planning</i>	<i>Cargo load planning</i>		
<b>Training</b>				
<i>Definition</i>	<p><i>Hardware and software that are used for educational and training purposes</i></p>			
<i>Source Definition</i>	<p><b>Training/CBT/CAI:</b> <i>An application that delivers education or training</i></p>			
<i>Typical examples</i>	<i>Computer Based Training (CBT)</i>	<i>Computer Aided Instruction (CAI)</i>	<i>Tutorial Applications</i>	<i>Courseware</i>

<b>Software Tools</b>				
<i>Definition</i>	<i>Software that is used for analysis, design, construction, or testing of computer programs</i>			
<i>Source Definitions</i>	<b>Testing Software:</b> Software for testing and evaluating hardware and software systems <b>Software Development Tools:</b> Software used for analysis, design, construction, or testing of computer programs			
<i>Typical examples</i>	<i>Compilers</i>	<i>Linker/loaders</i>	<i>Debuggers</i>	<i>Editors</i>
	<i>Assemblers</i>	<i>Requirements analysis &amp; design tool aids</i>	<i>Code generators</i>	<i>Programming aids</i>
	<i>Report generators</i>	<i>Code auditors</i>	<i>Test case data recording</i>	<i>Test case data reduction/analysis</i>
	<i>Test case generation</i>			
<b>Enterprise Service Systems</b>				
<i>Definition</i>	<i>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 .</i>			
<i>Source Definition</i>	<i>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.</i> <b>Electronic Data Exchange:</b> An application specialized in the structured transmission of business data or documents between separate parties (companies, organizations, etc.) without human intervention			
<i>Typical examples</i>	<i>Enterprise service management</i>	<i>Machine-to-machine messaging</i>	<i>Service discovery</i>	<i>People and device discovery</i>
	<i>Metadata discovery</i>	<i>Mediation service</i>	<i>Service security</i>	<i>Content discovery and delivery</i>
	<i>Federated search</i>	<i>Enterprise catalog service</i>	<i>Data source integration</i>	<i>Enterprise content delivery network</i>
	<i>Session management</i>	<i>Presence and awareness</i>	<i>Text collaboration</i>	<i>White boarding and annotation</i>
	<i>Application sharing</i>	<i>Application broadcasting</i>	<i>Virtual spaces</i>	<i>Identity management</i>
	<i>Content discovery</i>	<i>Collaboration</i>	<i>User profiling and customization</i>	
<b>Custom AIS Software</b>				
<i>Definition</i>	<i>Software needed to build a custom software application to fill a capability gap not captured by COTS/GOTS software packages</i>			
<i>Source Definition</i>	<b>Graphical User Interface:</b> 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) <b>Multimedia:</b> 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 <b>Internet Server Applet:</b> Platform-independent software which executes in the browser, typically JavaScript and its libraries. This may also include server-side scripting for example using PHP			



<i>Typical examples</i>	<i>Glue code</i>	<i>External system interfaces</i>	<i>Data transformation</i>	<i>Inter-COTS/GOTS data exchange</i>
	<i>Graphical User Interface</i>	<i>Internet Server Applet</i>	<i>Website</i>	
<b><i>Enterprise Information Systems</i></b>				
<i>Definition</i>	<i>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</i>			
<i>Source Definition</i>	<i>See SEER-SEM Enterprise Information Systems Definition See QSM Enterprise Information Systems Definitions</i>			
<i>Typical examples</i>	<i>Enterprise resource planning</i>	<i>Enterprise data warehouse</i>	<i>General ledger</i>	<i>Accounts payable</i>
	<i>Revenue and accounts receivable</i>	<i>Funds control and budgetary accounting</i>	<i>Cost management</i>	<i>Financial reporting</i>
	<i>Real property inventory and management</i>	<i>Document management</i>	<i>Logistic or Supply Planning &amp; Control</i>	<i>Transaction Processing</i>
	<i>Management Performance Reporting</i>	<i>Office Information System</i>	<i>Reservation System</i>	<i>Geographic or spatial information system</i>
	<i>Financial Transactions</i>	<i>Database management</i>	<i>Data Warehousing</i>	<i>Executive Information System</i>
	<i>Internet Server Applet</i>	<i>Report Generation</i>	<i>Office Automation</i>	<i>Data Mining</i>

## Appendix B – Productivity Quality Tags

**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 Quality Assurance (QA), Configuration Management (CM), Software (SW) Program Management (PM) and integration that were reported at the total effort level back to the lowest level CSCIs or WBS.

**Good - Alteration:** This is a data point that meets the criteria of Good but has some sort of alteration from the original reported SRDR. The entering analyst must include notes and rationale as to why the data was changed from the DCARC accepted SRDR submission.

**Good – Combined.** This is a data point that the government has created by combining multiple build submissions. The report’s supporting information 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 - Roll Up:** This is a data point that meets the criteria of Good, is at the Roll Up level, and lower level data points do not meet the qualification for Good. This normally occurs when a CSCI is on the same WBS reporting level as software support efforts that are specific to work, such as Product Engineering, CM, or IATCO. Normally Product Engineering, CM, and IATCO do not have SLOC counts associated with them, but do have hours that relate to the CSCI. Thus, the data point at the Roll Up Level represents the complete data point.

**Good – TD Proto.** This is a data point from the Technology Development phase of software development. 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 contractors can identify a size (SLOC) and development hours for the TD code, but while the code may be functional, it has not been developed to completion.

**Good – TD Proto Allocation.** This is a data point from a Technology Development effort 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 data point from a 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 data point 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 phase and follows the rules of an overall Roll Up.

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

**Allocation:** This is a point whose hours are allocated to other CSCIs. These data points normally include things like QA, CM, SW PM and integration. There is normally no SLOC associated with these data points, so the hours need to be allocated ("peanut butter spread") back to the other CSCIs.

**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 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.

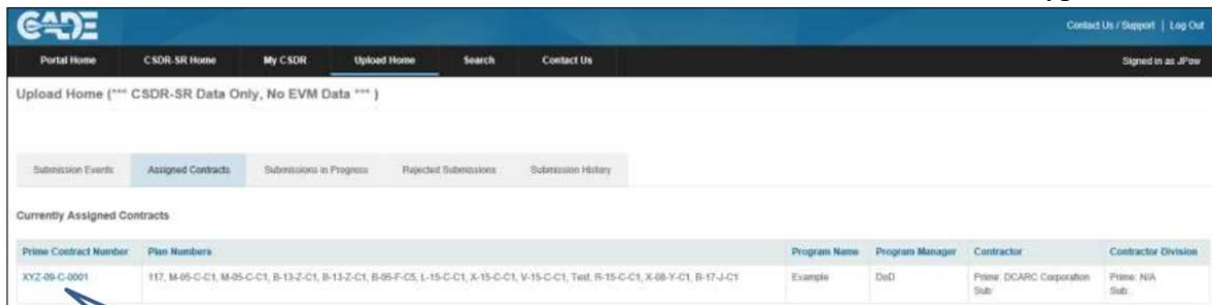
**Roll Up – Incomplete Lower Level** – This is a schedule at a parent level where it is determined that the data feeding the parent was not flagged as good, and at the roll up the data is determined to not be good as well. This can occur for interim build information, terminated contracts, missing SLOC 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 – CSDR Plans in CADE

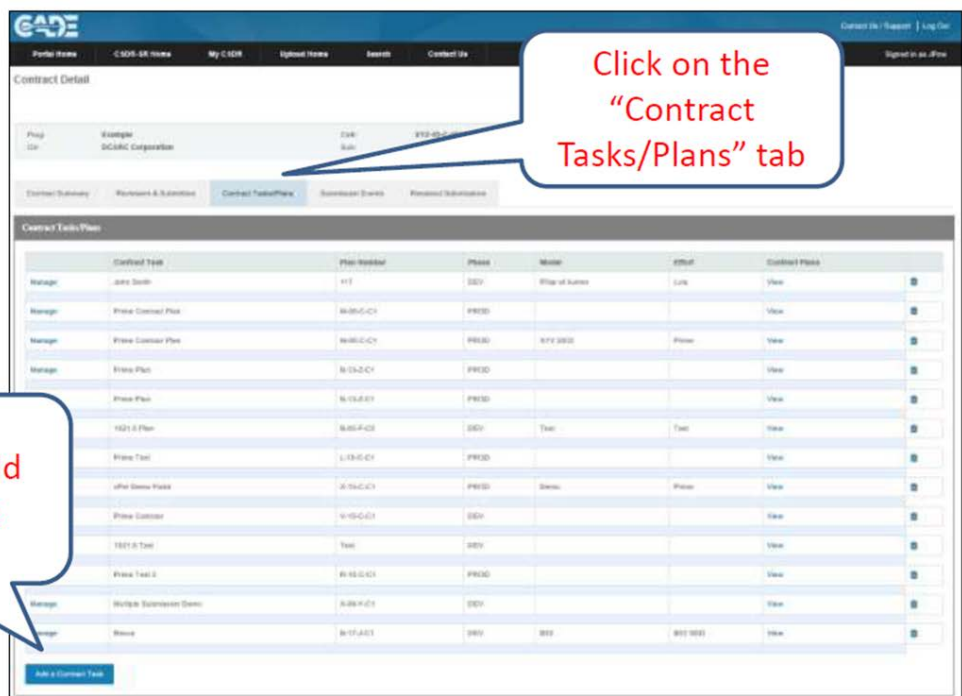
The following instructions are excerpts taken from the DCARC’s “CSDR Submit-Review Website Reviewer Guide” which can be found on the CSDR-SR Home page within CADE

- To access the CSDR plans the DCARC has on file for your contracts, you must access the Contract Detail screen for each contract
- You can access the Contract Detail screen by clicking on the contract number hyperlink on the “Assigned Contracts” tab on the Upload Home page
- You can also access the Contract Detail screen by navigating to Search > Search Contracts or Search > Search Submissions. Click on the contract number hyperlink



Click on the Prime Contract Number to view the Contract Details

- From the Contract Detail screen select the “Contract/Task Plans” tab



Click on the “Contract Tasks/Plans” tab

Click here to add contract tasks

- Selecting “View” will allow you to access both the XML and Excel version of the CSDR plans that are currently on file. Select the XML and Excel files with the most recent plan date.

The screenshot displays the GADe portal interface. At the top, there is a navigation bar with the GADe logo and links for Portal Home, CSDR-SR Home, My CSDR, Upload Home, Search, and Contact Us. On the right side of the navigation bar, there are links for Contact Us / Support and Log Out, along with a 'Signed in as jPow' indicator. Below the navigation bar, the page title is 'Contract Detail'. The main content area shows contract information: Prog: Example, Ctr: XYZ-09-C-0001, and Ctr Sub: DCARC Corporation. There are tabs for Contract Summary, Reviewers & Submitters, Contract Tasks/Plans (which is selected), Submission Events, and Received Submissions. Below the tabs is a table titled 'Contract Tasks/Plans' with columns for Contract Task, Plan Number, Phase, Model, Effort, and Contract Plans. The table contains two rows: one for 'Manage' by John Smith with Plan Number 117, Phase DEV, Model Pillar of Autnm, Effort Lots, and a 'View' link; and another for 'Manage' of a Prime Contract Plan with Plan Number M-05-C-C1, Phase PROD, and a 'View' link. A callout box with a blue border and red text points to the 'View' link in the first row, containing the text: 'Click “View” to download the XML and Excel versions of the approved CSDR Plan'.

Contract Task	Plan Number	Phase	Model	Effort	Contract Plans
Manage John Smith	117	DEV	Pillar of Autnm	Lots	<a href="#">View</a>
Manage Prime Contract Plan	M-05-C-C1	PROD			<a href="#">View</a>

## Appendix E – Future V&V Questions

1. Is the submission compliant with the CSDR Plan, i.e., a comparison of the submission to the plan requirement?
2. Is there consistency of nomenclature and WBS numbering, and definitions from submission to submission?
3. If there are prior submissions, does the latest submission clearly describe what report the prior submission is linked to?
4. If a prior submission exists, is the information that has changed readily identifiable and a reason for the change provided (in the comments section)?
5. If this is an Interim or Final Report, was there an Initial Report it can be traced to?
6. Are at least three precedents reported and consistent from submission to submission?
7. Is the Process Maturity rating reported with an associated date, and has it changed from a prior submission?
8. Were SLOC counts reported in another data submission and are they traceable from submission to submission or build to build, if applicable?