DATA ITEM DESCRIPTION (DID)

Title: COST BREAKDOWN STRUCTURE (CBS) DETAIL REPORT (DI-FNCL-80166B)

Number: DI-FNCL- 80166B **Approval Date**: **3 Feb 05**
AMSC NUMBER: **N7546** **Limitation: N/A**
DTIC Applicable: **N/A** **GIDEP Applicable: N/A**
Office of Primary Responsibility: G/DA012
Applicable Forms: **N/A**

Use/Relationship:

The Cost Breakdown Structure (CBS) Detail Report provides the contractor's cost associated with a government contract issued to acquire a program, project, or system.

This DID is applicable to all cost-reimbursement, and all incentive based Research, Development, Test, and Evaluation (RDT&E) and Procurement funded contracts over $25 million in total value. Costs shall be reported through cost to the Government including General & Administrative (G&A) cost, but shall not include cost of money or Fee, unless otherwise noted.

Report all contract costs to a level of detail comparable to that presented in the standard CBS in Format 1. This CBS is applicable to a single program or increment. Costs included in these reports shall be current as of sixty calendar days (or less) prior to submission. While the cost data provided in the DID is not Earned Value Management (EVM) data, it must come from the same internal financial accounting system as that used to generate EVM reports. This DID requirement and all associated reporting requirements shall be flowed down to all subcontractors and other company divisions whose efforts comprise over $10 million ***or*** more than 25% of the total prime contract value. One consolidated report (Formats 1, 2 and 3) shall be delivered to the government including the prime contractor and all subcontractors/inter-divisions. In addition, all Formats 1 through 3 that were completed by subcontractors or other company divisions must also be provided to the government.

The initial submission of the DID is due within sixty days of contract award (or exercise of option) of each increment. The initial submission shall contain the latest Estimate-At-Completion (EAC). The final report is due within sixty days or less after each increment is delivered and accepted by the Government or is terminated by the Government. The final submission will include all costs for the increment to include any outstanding obligations/commitments. The reporting period for the DID shall be from the start date of the current increment to the end date of the current increment. Each submission will be subject to Government approval, which includes approval of methods used to map lower level costs into end item costs for both the prime and the subcontractors.

Requirements:

The CBS Detail Report employs three types of formats containing cost and technical data elements. Format 1 (see Figure 1 - Cost Breakdown Structure) provides data to capture costs by CBS elements, while Format 2 (see Figure 2 – Computer Software Configuration Item (CSCI)) provides technical data on software and Format 3 (see Figure 3 – Hardware Configuration Items) provides data on hardware elements.

1. Format 1.
The CBS contained in Format 1 was derived per MIL HDBK-881 guidance with high interest and critical items of interest to the Government expanded. For clarification and to expand any CBS elements to lower levels that depicted in Format 1, refer to MIL HDBK-881. Since MIL HDBK-881 is Development and Production oriented and does not address Operations and Support in sufficient detail, the Cost Element Structure found in the DoD Automated Information System Guide is utilized for the O&S sub elements of CBS 13.0.

If the Contract Work Breakdown Structure (CWBS) is not structured as a product-oriented CBS (i.e., identifies hardware and software end items) or does not isolate end item costs at the level of detail described above, then the CWBS shall be mapped into the CBS provided in this DID. In order to provide this information the costs shall be isolated for each applicable hardware end item, software end item, and supporting elements. Information shall be provided tracing the downward allocation, upward aggregation, or lateral transfer of costs from the CWBS to the resulting end items, such that the contributions of each CWBS element to each end item are shown. Additionally, information tracing costs from the CWBS to the standard CBS shall be provided. Software end item Computer Software Configuration Item (CSCI) or major single function costs shall be reported at CBS level five at a minimum. Systems Engineering Integration and Test, and Program Management costs are reported at various CBS levels, as displayed in the standard CBS. A CWBS dictionary to the lowest level available shall be provided by the contractor.

Although the Format 1 in this DID depicts the complete Lifecycle of a product, only fill in Format 1 entries for those CBS indented items pertinent to the Program, Project, or System phase supported by this contract/increment. Two submissions are required per increment, one at the beginning of the increment (Estimate at Completion (EAC)) and the second at the end of the increment (Actual Final Costs).

1.1 Format 1 Section A, Contractor/Project Information

1. Contractor: Contractor's name and company division in 1A, primary work location in 1B.

2. Contract: Name as assigned to the contract by the procuring agency in 2A, Contract designated number from the procuring agency in 2B, brief description of the contract in 2C, and Type (as CPIF, CPAF, etc.) in 2D.

3. Program: Name of the program, project, or system, if different from the name entered in 2A, shall be entered into block 3A, otherwise enter "same". Place in block 3B a brief description of the program.

4. Report Period: Enter in 4A and 4B year, day, and month of the time interval covered by this report. Start Date in 4A should be the start date of the contract/Increment being reported.

5. Date Submitted: Actual date of report completion and release of information to the Government Program Office.

6. Dollars in: Denote entries that are factored down by thousands (K) or millions (M).

7. Price to Sell: Shall enter total price to include burdens, COM and Fee

1.2 Format 1 Section B, Functional elements by CBS

Labor Hours - Enter the number of staff-hours (to include subcontractor, interdepartmental work transfer authorizations (IWTAs), consultant and purchase labor hours) devoted to each CBS item. Total labor hours (EAC for first submittal, final actual hours for last submittal) for each CBS may be further broken down into prime, subcontractor, IWTA, consultant, or purchase hours if desired but it is not required so long as the total includes them.

Labor Costs - Enter the direct fully burdened labor cost (EAC for first submittal, final actual cost for last submittal) for each CBS item. Labor costs shall be reported consistently with Labor Hours as described in the above paragraph. When reporting

other than prime labor costs (e.g., subcontractor, IWTA, consultant, purchased labor) in this element do not include any non-prime, non-labor dollars (i.e., subcontractor or IWTA ODCs or material). These other than labor non-prime costs should be reported with the prime contractor dollars for the same elements (see material and ODC elements below). Do not enter values that include cost of money or Fee.

Material Costs - Enter the material cost (EAC for first submittal, final actual cost for last submittal)to include applicable burdens, for each CBS element. Material costs for subcontractors, consultants, and IWTAs should also be captured here. This includes the cost of all COTS products purchased or licensed to develop this system. Do not enter values that include cost of money or Fee.

Other Direct Costs (ODCs) - Enter the total "other direct costs", to include applicable burdens, for each CBS element. ODC costs for subcontractors, consultants, and purchase labor should also be captured here. Identify the ODCs in block 8 Remarks, including burdens (i.e., Travel = $X,XXX)**.** Do not enter values that include cost of money or Fee.

Total Costs - Enter the total cost (summation of labor, material, and ODCs) by CBS element. Do not enter values that include cost of money or Fee.

1.3 Format 1 Section B (cont.): CBS Definitions

0.0 Total System
Includes the top summary of all equipment, hardware, software, data, services, and facilities costs required to develop, produce, upgrade, sustain and dispose of a product system. This is the total aggregation of the complex of equipment (hardware/software), data, services, and facilities required to develop and produce an electronic, automated, or software system capability such as a command and control system, radar system, communications system, information system, sensor system, navigation/guidance system, electronic warfare system, support system, etc.

1.0 Prime Mission Product (PMP)
Includes the hardware and software used to accomplish the primary mission of the Defense materiel item. This includes 1.) all integration, assembly, test and checkout, as well as all technical and management activities associated with individual hardware/software elements; 2.) the integration, assembly, test and checkout associated with the overall PMP, and when the electronic/automated software system comprises several PMPs, each will be listed separately at level 2; 3.) any whole and partial prime contractor, subcontractor, and vendor breadboards, brass boards, and qualification test units; 3.) the design, development and production of complete units (i.e., the prototype or operationally configured units that satisfy the requirements of their applicable specification(s), regardless of end use); and 4.) all factory special test equipment, special tooling, and production planning required fabricating the PMP.

1.1 Hardware
This element includes the hardware and integrated software components of the specific electronic/automated software system. Integrated software is all software that is an integral part of any specific hardware -system specification. This element also includes: 1.) all associated special test equipment, special tooling, production planning, and all technical and management activities; 2) all in-plant integration, assembly, test, and checkout of hardware components and software into an electronic/automated software subsystem, including the subsystem hardware and software integration and test; 3.) interface materials and parts required for the in-plant integration and assembly of other level 4 components into the electronic/automated software subsystem and all materials and parts or other mating equipments furnished by/to an integrating agency or contractor; 4.) cables, conduits, connectors, shelters, and other devices associated with the operational electronic/automated software subsystem; and 5.) the design, development, production, and assembly efforts to provide each electronic/automated software subsystem as an entity.

1.1.1…1.1.98 Hardware Configuration Item 1...n (Specify Name of HWCI)
This element refers to those HWCIs in each system. This element should be replicated as required for each HWCI within the system. This CBS item includes all hardware units and components required to have functioning HWCIs, and conform to the planned configuration. Includes commercial-off-the-shelf (COTS) and government-off-the shelf (GOTS) hardware incorporated into the design. Any integration, assembly, test, and checkout of hardware within the HWCI is included. Also, includes internal HWCI interface materials and parts required for the integration and assembly of hardware components. Element captures the engineering design, development, prototype/brass board manufacturing, and assembly efforts to provide each HWCI as an entity.

1.1.99 Integration, Assembly, and Checkout of Hardware Configuration Items (HWCIs)
Includes all effort directly associated with the hardware subsystem HWCIs integration, assembly, test and checkout as they are brought together into the subsystem.

1.2 Prime Mission Product Applications Software
This element includes software that is specifically produced for the functional use by a computer. This includes: 1.) battle management, weapons control, and data base management. 2.) all effort required to design, develop, integrate, and checkout the PMP applications computer software configuration items (CSCI's), not including the non-software portion of PMP firmware development and production.

1.2.1…1.2.98 Applications Software Build 1…n (Specify Name for Build/Spiral)
Includes in each software build an aggregate of one or more CSCIs that satisfies a specific set or subset of requirements. When incremental, spiral, or another software development method is used, multiple builds may be necessary to meet program requirements. A build is a separately tested and delivered product. Within builds are CSCIs. When a build is complete, a portion or all of one or more CSCIs will be completed. Therefore, a CSCI may appear in more than one build, but will be successively more functional as each build is completed.

1.2.1.1…1.2.1.96 CSCI 1...96 (Specify Name for CSCI)
Shall include the aggregation of software or any of its discrete portions, which satisfies an end use function and has been designated by the government for configuration management. CSCIs are the major software products of a system acquisition and are developed in accordance with standard DoD or commercial practices and process. This includes: 1.) reusable software components, such as commercial off-the-shelf software, government furnished software, or software specifically developed for reuse. 2.) glue code, which is used to ensure proper interface and operation of the COTS and custom software. 3.) effort associated with the requirements analysis, design, coding and testing, CSC's integration and testing, CSCI formal qualification testing, and software problem resolution of each CSCI. 3.) Computer Software Components (CSCs) which are functionally or logically a distinct part of a CSCI, distinguished for convenience in designing and specifying a complex CSCI as an assembly of subordinate elements. Note that often CSC's may be further decomposed into Computer Software Units (CSUs) by a software organization.

1.2.1.1.1 Requirements Analysis
This element contains all the resources associated with the software configuration item requirements analysis. This is the process by which a complete set of engineering and interface requirements are defined for each configuration item. Representative activities include: analysis of preliminary software requirements, identification and allocation of software requirements into configuration items, analysis of preliminary interface requirements, and identification and resolution of interface requirements.

1.2.1.1.2 Design
This element contains all the resources associated with the software configuration item design. This is the process of decomposing a high-level abstract requirement into lower level software elements. Preliminary design and detailed design activities are accomplished to map out high-

level as well as low-level strategies for allocating requirements for each configuration item to design entities, e.g. objects, classes, modules, CSC(s), etc. Representative activities include: creating and maintaining SDFs, analysis of preliminary software design(s), derive and map out high (top) level software design specifications, devise and map out low level (detail) software design specifications, analysis of preliminary interface design specifications, define and describe interface design specifications, generate input to software test planning, and formalize test requirements for design entities.

1.2.1.1.3 Coding and Design Entity Testing
This element contains all the resources associated with the software configuration item coding and design entity testing. This is the process of implementing the software design in a program language that may then be converted mechanically (i.e. by compilation) to an acceptable machine-executable representation. Each design entity is coded and subsequently tested to ensure that it satisfies its specific requirement. Representative activities include: maintaining SDFs, coding and compiling activities, conducting testing and analysis, code walk-through activities, performing compliance checks to coding conventions, and developing lower level design entities test and integration procedures.

1.2.1.1.4 Design Entity Integration and Test
This CBS element contains all the resources associated with the software configuration item design entity integration and testing. This is the process of integrating or building design entities into a configuration item and the testing of lower level threads to verify that the algorithms and data employed in interfacing each design entity are correctly specified and implemented. Representative activities include: perform design entity integration analysis, perform design entity build and lower level thread testing, record test results, and perform dry run of formal qualification tests.

1.2.1.1.5 Configuration Item Testing
This CBS element contains all the resources associated with the software configuration item testing. This is the process of demonstrating that the configuration item can perform correctly under the full range of operating conditions and that it satisfies its requirements. Representative activities include: conduct formal qualification tests, conduct test analysis and record test results, and generate input to the STRs.

1.2.1.1.6 CSCI COTS Software
This CBS element contains all the resources associated with the procurement, integration, verification and test of any COTS software. Representative activities include: purchasing COTS software packages and evaluating their suitability to meet program requirements regardless of whether the COTS package is ultimately used in the end deliverable.

1.2.1.97 System Engineering of Applications Software Build 1...n CSCIs - This CBS element contains the resources associated with all engineering from functional specialists (excluding checkout/ test and evaluation) in support of the Applications Software Build custom and COTS software. This engineering support includes: System Engineering, Quality Assurance, Reliability and Maintainability, and Human Engineering, that is associated with the Application Software Build custom and COTS software. This CBS element contains all the resources associated with integration and test verification and validation of the Applications Software Build custom and COTS software. Representative activities include: generate input to software test plans, descriptions, and procedures; define software test cases; perform custom and COTS software integration analysis; perform software build and test; and update SDFs. This element also includes any Build-specific algorithm development performed by the scientific/engineering/ mathematical team. Generally, the scientific/engineering/ mathematical algorithm development and any rudimentary coding is performed as a level of effort within the system engineering function. If mapping cannot be made at this level, then the algorithm development should be booked at the next higher level of System Engineering.

1.2.1.98 Program Management of Applications Software Build 1…n CSCIs - This CBS element includes the Applications Software Build software PM, which includes management,

direction and control of all effort contributing to the development, production, custom and COTS procurement, and integration of the Applications Software Build software. It includes overall administration, project controls, product effectiveness, configuration management, warranty administration, subcontract management, vendor liaison, logistics management, and security management.

1.2.1.99 Integration and Checkout of Application Software Build 1…n CSCIs - In those instances in which an integration, assembly, test, and checkout element is needed at the build level, this element includes the effort of technical and functional activities associated with the design, development, and production of software required to assemble this level of equipment software elements into the next higher CBS level and not directly a part of any other individual CBS element.

1.2.99 Integration, Assembly, Test & Checkout of All Application SW Builds
Includes all effort directly associated with the software CSCI's integration, assembly, test and checkout of all software builds as they into a complete built software system.

1.3 Prime Mission Product System Software
Includes the software designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system and associated programs. Includes: 1.) the operating systems, compilers, and utilities. 2.) all effort required to design, develop, integrate, and checkout the PMP system software, including all software developed to support PMP-Applications-Software Development. 3.) PMP system software required to facilitate development, integration, and maintenance of any PMP software build and CSCI.

For lower level 1.3 elements, follow the same breakout as 1.2 PMP Applications Software.

1.4 PMP Integration, Assembly, Test and Checkout
This element includes the effort of technical and functional activities associated with the design, development, and production of hardware and software required to assemble both the PMP Hardware, Application and System Software packages that is not directly a part of any other individual CBS element.

2.0 System Engineering/Program Management
This element is a rollup of CBS subelements that contain the overall management and technical direction costs associated with the program. Functions included in this element are: business, administrative, and engineering associated with planning, organizing, directing, coordinating, controlling, and accomplishing overall program objectives.

2.1 System Engineering
The system engineering function is responsible for the analysis, derivation, allocation, and traceability of derived system requirements and interfaces. This effort includes establishing system-level requirements; system/subsystem or subsystem/subsystem interfaces; system analyses, models and simulations; quality; configuration control; while also assessing performance of, and technically managing the design, development, production, and integration of the system.

This element includes: 1.) Effort to define the system and the integrated planning and control of the technical program efforts of design engineering, specialty engineering, production engineering, and integrated test planning. 2.) Effort to transform an operational need or statement of deficiency into a description of system requirements and a preferred system configuration. 3.) Technical planning and control effort for planning, monitoring, measuring, evaluating, directing, and replanning the management of the technical program. 4.) Where applicable - value engineering, configuration management, human factors, maintainability, reliability, survivability/vulnerability, system safety, environmental protection, standardization, system analysis, logistic support analysis, etc.

This element also includes support functional specialists (excluding checkout/test and evaluation), who provide technical planning, technical management, analysis, and support efforts for the development of the system. This engineering support typically includes: Systems Engineering, Quality Assurance, Reliability and Maintainability, and Human Engineering associated with the overall system.

Algorithms developed specifically for and in support of the system that cannot be mapped to a lower level system engineering element are included in this element.

2.2 Program Management
Includes the program and administrative planning, organizing, directing, coordinating, controlling, and approval actions designated to accomplish overall program objectives which are not associated with specific hardware elements and are not included in systems engineering. Includes: 1.) Cost, schedule, performance measurement management, warranty administration, contract management, data management, vendor liaison, subcontract management, etc. 2.) Support element management, defined as the logistics tasks management effort and technical control, and the business management of the support elements. The logistics management function encompasses the support evaluation and supportability assurance required to produce an affordable and supportable defense materiel system. 3.) Planning and management of all the functions of logistics such as maintenance support planning and support facilities planning; other support requirements determination; support equipment; supply support; packaging, handling, storage, and transportation; provisioning requirements determination and planning; training system requirements determination; computer resource determination; organizational, intermediate, and depot maintenance determination management; and data management.

2.3 Mission Assurance (MA)
Includes Mission Assurance (MA) planning and generation of measures intended to provide a defined and accepted level of confidence that the delivered system can meet expected mission objectives under a wide range of operating conditions. Also, identified is the level of residual risk and the documentation of the conscious decision to accept that risk. As a minimum, MA involves:

A. Defining Critical Objectives
B. Providing a Continuum of Capabilities
C. Defining the Range of Operating Conditions
D. Measuring Confidence
E. Defining quantitative Measures of Mission Assurance
F. Enhancing Readiness planning
G. Continuity of Operations Planning/Preplanning, often referred to as COOP
H. Critical Infrastructure Protection (CIP) planning

2.4 Information Assurance
Includes the effort involved in assuring that information systems are protected from unauthorized access and that the confidentiality, integrity and availability of the systems is maintained. This includes providing for restoration of information systems by incorporating protection, detection and reaction capabilities.

3.0 System Integration, Assembly, Test & Checkout
Includes the effort involved in providing technical and engineering services to the host developer, manufacturer or integrator during the installation and integration of the PMP into the host's system. Support provided by the producer/integrator of the PMP is included in this CBS element. This includes I&T management – requirements definition, planning and scheduling; development of test plans and procedures; test preparations, conduct and teardown; development of software for supporting system I&T; and review, analysis and documentation of test results.

4.0 System Test and Evaluation
The use of prototype, production, or specifically fabricated hardware/software to obtain or validate engineering data on the performance of the system. This includes detailed planning, conduct, support, data reduction and reports (excluding the Contract Data Requirements List data) from

such testing, and all hardware/software items which are consumed or planned to be consumed in the conduct of such testing, as well as all effort associated with the design and production of models, specimens, fixtures, and instrumentation in support of the system level test program. Please note that test articles which are complete units (i.e., functionally configured as required by specifications) are excluded from this work breakdown structure element.

4.1 Development Test and Evaluation
Testing effort that is planned, conducted and monitored by the developing agency of the DoD component. This includes test and evaluation conducted to demonstrate that the engineering design and development process is complete, the design risks have been minimized, the system will meet specifications, as well as estimate the system's military utility when introduced, determine whether the engineering design is supportable (practical, maintainable, safe, etc.) for operational use, provide test data with which to examine and evaluate trade-offs against specification requirements, life cycle cost, & schedule, and perform the logistics testing efforts to evaluate the achievement of supportability goals, the adequacy of the support package for the system. This element shall also include any contractor coordination effort with Operational Testing Authority (OTA) and Joint Interoperability Test Command (JTIC).

4.2 Operational Acceptance Test and Evaluation (OT&E)
The test and evaluation conducted by agencies or elements other than the developing activity to assess the prospective system's military utility, operational effectiveness, operational suitability, logistics supportability (including compatibility, inter-operability, reliability, maintainability, logistic requirements, etc.), cost of ownership, need for any modifications, and complete acceptance testing. Included also is Test and Evaluation Support and Test Facilities. This shall include all contractor support to OTA and JITC prior and during OT&E.

4.3 Test and Evaluation Support
The support elements necessary to operate and maintain, during test and evaluation, systems and subsystems which are not consumed during the testing phase and are not allocated to a specific phase of testing. Also includes repairable spares, repair of repairable, repair parts, warehousing and distribution of spares and repair parts, test and support equipment, test bed vehicles, drones, surveillance aircraft, tracking vessels, contractor technical support, etc.

4.4 Test Facilities
The special test facilities required for performance of the various developmental tests necessary to prove the design and reliability of the system or subsystem. Also includes test tank test fixtures, propulsion test fixtures, white rooms, and test chambers.

4.5 Independent Verification and Validation
Includes all contract support to Independent Verification and Validation. Independent Verification and Validation (IV&V) ensures that the software meets all performance standards and adheres to all stated requirements of the functional description and documentation procedures. Also includes reviewing and evaluating deliverable items, such as system specification, user manuals, program specifications, maintenance manuals, test plans, and operations manuals; accomplishing configuration audits; conducting validation testing; and establishing and maintaining the test database.

5.0 Training
Includes deliverable training services, devices, accessories, aids, equipment, and parts used to facilitate instruction through which personnel will learn to operate and maintain the system with maximum efficiency. Includes also Equipment, Services, and Facilities devoted solely to training.

6.0 Data
All deliverable data required to be listed on a Contract Data Requirements List, DD Form 1423. It includes acquiring, writing, assembling, reproducing, packaging and shipping the data, as well as transforming it into government format. Subcategories included are creation and production of Technical Publications, Engineering Data related to scientific or technical information and computer software documentation, Management Data covering configuration management, cost,

schedule, contractual data management, program management, etc., required by the government, Support Data related to data items designed to document support planning in accordance with functional categories selected, and the operation of a Data Depository as custodian to maintain a master engineering specification and establish a drawing depository service for government approved documents that are the property of the U.S. Government.

7.0 Peculiar Support Equipment
Includes the design, development, and production of those deliverable items and associated software required to support and maintain the system or portions of the system while the system is not directly engaged in the performance of its mission, and which are not common support equipment (See below). Also includes the peculiar or unique testing and measurement equipment and the peculiar Support and Handling Equipment.

8.0 Common Support Equipment
Includes the items required to support and maintain the system or portions of the system while not directly engaged in the performance of its mission, and which are presently in the DoD inventory for the support of other systems. This includes DoD inventory items of Test and Measurement Equipment, as well as DoD inventory Support and Handling Equipment.

9.0 Operational Site Activation
The real estate, construction, conversion, utilities, and equipment to provide all facilities required to house, service, and launch prime mission equipment at the organizational and intermediate level. Includes: 1.) conversion of site, ship, or vehicle. 2.) system assembly, checkout, and installation (of mission and support equipment) into site facility or ship to achieve operational status. 3.) contractor support in relation to operational/site activation. It also includes System Assembly, Installation, and Checkout on Site, Contractor Technical Support, Site Construction, and Site/Ship/Vehicle Conversion.

10.0 Industrial Facilities
The construction, conversion, or expansion of industrial facilities for production, inventory, and contractor depot maintenance required when that service is for the specific system. Includes equipment acquisition or modernization, where applicable, maintenance of these facilities or equipment, and industrial facilities for hazardous waste management to satisfy environmental standards.

11.0 Initial Spares and Repair Parts
Includes deliverable spare components, assemblies and subassemblies used for initial replacement purposes in the materiel system equipment end item. This includes any repairable spares and repair parts required as initial stockage to support and maintain newly fielded systems or subsystems during the initial phase of service, including pipeline and war reserve quantities, at all levels of maintenance and support. Note it does not include development test spares and spares provided specifically for use during installation, assembly, and checkout on site.

13.0 Operations and Support
Operations and Support of a system at all sites. Includes the management and maintenance of all hardware and software elements throughout the life cycle after close of development and production or deployment fielding. Shall include the rollup of all sub-elements. Operation and Maintenance efforts start with development of O&M operational procedures, training aids and courses and continue through system shut down and retirement. It includes the overall on-site and off-site maintenance of all system hardware and software.

13.1 Hardware Maintenance
This element is a summation of elements containing all costs to maintain and repair the system hardware. It includes personnel for functions such as troubleshooting, debugging, etc. plus long-term planning, and ground hardware and documentation updates. It includes maintenance personnel providing periodic, scheduled and unscheduled, maintenance of COTS or custom hardware. It also includes all maintenance materials and consumables used in the operation and maintenance.

13.1.1 Licenses/Vendor Maintenance Agreements (COTS/GOTS).
This element includes the cost of licenses/vendor maintenance agreements associated with the maintenance and repair of system hardware.

13.1.2 Spares/Repair Parts.
Includes all costs of spares and parts that contribute to the maintenance and repair of system hardware

13.1.3 Fixes/Repairs.
Includes all costs of fixes/repairs to system hardware.

13.1.4 Minor Enhancements
Includes costs of minor enhancements to the system hardware.

13.2 Software Maintenance.
Includes all costs for software maintenance (e.g., version updates).

13.2.1 Licenses/Vendor Maintenance Agreements (COTS/GOTS)
Includes costs of licenses/vendor maintenance agreements (e.g., annual software license fees for COTS software) and associated with the maintenance and repair of system software.

13.2.2 Fixes (Software)
Includes all costs of fixes to system software.

13.2.3 Minor Enhancements.
Includes all costs of minor enhancements to the system software.

13.2.4 Security Testing and Integration.
This element includes all costs of integration and testing of COTS/GOTS software-based security capabilities developed for a system, in order to operate and support a system.

13.3 Data Maintenance.
Includes all costs to keep the system data current.

13.4 Unit/Site Operations.
Includes all personnel and equipment specifically functioning to operate the system. Representative activities include training, security, fuel, and facilities maintenance.

13.4.1 Personnel.
Includes cost of Government (civilian & military) and contractor personnel required to operate the system.

13.4.2 Consumables.
This element includes the cost of all expendable material or supply items (excluding replenishment spares and items included in Site Operations) consumed in operating and maintaining a system.

13.4.3 Infrastructure Maintenance.
Includes the cost of maintenance required to operate and support the infrastructures communications, facilities, etc. It includes expenditures made to maintain physical plant, such as utilities, rents, etc.

13.4.4 System Security.
Includes all security related costs involved in operating the system that can be separately identified and are not contained within a discrete end item (HWCI or CSCI).

13.4.5 Recurring Training.

Includes costs related to training and retraining on system specific tasks after the fielding of the operating system.

13.5 Replenishment Spares.
This element includes components, assemblies and subassemblies required to stock the maintenance warehouse in support of the operation and maintenance of the system. These costs include the transportation and storage of these supplies and spares.

13.6 Sustaining Engineering
This element includes the performance of test, validation, and installation for system hardware/software changes. Includes test-engineering functions for operations integration of changes.

13.7 Communications
This cost element aggregates the cost of leasing and maintenance for the system communication costs. This element shall include leased long lines, long distance networks for data and voice, and other costs to interconnect components of the system. When communications are shared with other systems, costs will be prorated.

13.8 Interim Contractor Logistics Support
This element includes contractor services for labor, material and overhead to provide logistics support to the system. This element is usually a temporary service between IOC and FOC until the government logistics support system can take ownership of the system.

13.9 Contractor Logistics Support
Shall include all costs to provide logistics support to the system. This element includes costs from FOC for the operational life of the system.

13.10 Mission Assurance Peculiar Activity Support
Includes Mission Assurance (MA) Peculiar Activity Support in this phase involved in the conduct of Continuity of Operations or COOP. COOP in this phase is the testing, and implementation of the previously planned processes and procedures in CBS item 1.3 to recover essential operational capability in the event of a failure. COOP refers to a specific category of the Mission Assurance activities undertaken in the operations phase of the mission to increase resilience (equivalently to a decrease in the Mean-Time-To-Recover or MTTR). COOP includes O&S costs related to the recovery from major damage, and capability loss not captured elsewhere in this CBS.

13.11 Other
This element includes other uncategorized Operations and Support expenditures (As fully described and justified and not part of any other O&S CBS Element).

14.0 Disposal. Includes all costs required to terminate the commission and dispose of the system at the end of its life cycle. Includes any costs for demilitarization and decommission.

The remainder of this page intentionally left blank

2. Format 2A/B– Technical Report: Software

This report is to be delivered for each contract increment for each CSCI. Two submissions are required per increment, one at the beginning (Format 2A) of the increment representing the estimate and the second (Format 2B) at the end of the increment representing the actual. The Government will provide as GFI both report formats in Microsoft EXCEL along with a script to assist collection for Format 2B.

2.1 Format 2A, Technical Report: Estimate (Due at increment start)

2.1.1 Section A: CSCI/Project Information

1. Contractor: Enter contractor's name and company division in 1a, primary work location in 1b, contractor role (prime/subcontractor) in 1c, and designated prime contract number from the procuring agency in 1d.

2. Schedule: Enter in 2a year, month, and day of the estimated/actual start date for increment being reported. Enter in 2b year, month, and day of the estimated/actual completion date for increment being reported. Enter in 2c the date of report completion. Enter in 2d the current version number of the software being developed if applicable.

3. Name of Computer Software Configuration Item (CSCI) or Project: Enter the name of the CSCI, program, project or system being reported.

4. Corresponding CBS Number: Enter the corresponding contract breakdown structure (CBS) number for the software configuration item (CSCI) being reported.

2.1.2 Section B: Process Information/Requirements

5. Application Type: If CSCIs are designated, describe the planned primary generic function for the CSCI being reported, otherwise describe the planned primary generic function of the program. Possible descriptions could include: Web, MIS, Database, Signal Process, Analytical, Cryptologic, etc.

6. Development Method: Describe the planned development method(s). Possible descriptions could include: None, Evolutionary, Incremental, COTS Integration, Prototype, Rapid Application Development, Spiral, Waterfall, etc.

7. Percent Code Designed for Re-Use: Enter the percentage of code that is planned to be developed for reuse in the future on other systems. Use in subsequent releases on the existing system shall not be considered reuse. If no formal requirement is documented for the CSCI or project, requiring the development of reusable code, enter 0%.

8. CMM/CMMI Level: Enter the Software Engineering Institute's (SEI) Capability Maturity Mode (CMM) or Capability Maturity Model Integrated (CMMI) level, at the time of report generation, for the primary contractor responsible for the CSCI or project. Possible responses are: Level 1, Level 2, Level 3, Level4, Level 5 ONLY.

9. CMM/CMMI Level based on Self Assessment or External Assessment: Enter Self-Assessment or External-Assessment to support SEI level entered in Format 2, Section B element 8.

10. Total Number of Requirements: Enter an integer number representing the planned TOTAL number of requirements approved and allocated to this CSCI.

2.1.3 Section C: Software Metrics - Complete number 11 to describe the primary approach used to estimate the size of this CSCI or project. Provide SLOC and Object Oriented (00) metrics in numbers 12 and 13.

11. Project Metrics - Describe the method used to estimate the size of this CSCI or project.

    a. Sizing Methodology - Enter the primary estimating method (e.g. COSMIC, Mark II, IFPUG, Top-Down, Bottom-Up, ObjectMetrix, Object Points, Analogy, Expert Opinion, etc.) used to estimate the software size for this CSCI or program. If the method used to estimate this CSCI or program is Analogy, identify the analogous data point and provide its corresponding metrics in #15, Software Size Remarks.

    b. Unit of Measure – Shall enter the unit of measure (e.g. function points, SLOC, Use Cases, Object Points, etc.) for the methodology being used to estimate the software size of this CSCI or program.

    c. Size - Shall enter the estimated size in the same units-of-measure indicated in 11b.

12. SLOC Estimate - Quantify the estimated SLOC for this CSCI. If the unit of measure provided in 11b above is SLOC then the information in 12a should be consistent with the size provided in 11c.

    a. SLOC-Count Technique - Enter the estimating methodology used to estimate the software size for this CSCI or project. For analogy estimates, identify the analogous data point and its corresponding metrics in #15, Software Size Remarks. If the method used to estimate SLOC is a conversion of the size metric provided in 11c, the conversion algorithm shall be provided in #15, Software Size Remarks.

    b. Estimated New Source Lines of Code (SLOC) - Enter the estimated new source lines of code required to for this CSCI.

    c. Estimated Re-Used SLOC - Enter the estimated re-used source lines of code that this CSCI will contain.

13. 00 Metrics - Provide the additional sizing metrics in numbers 13a through 13d if they are applicable to this CSCI or project.

    a. Number of Use Cases – Enter the number of use cases. A use case describes a dialogue between an external actor and the system. A use case comprises a sequence of activities performed by an end user in the course of their work that results in measurable business benefit for the user. Essentially it forms a complete business transaction. The (Unified Modeling Language UML) specification states that a use case is a kind of classifier representing a coherent unit of functionality provided by a system, as manifested by sequences of messages exchanged among the system and one or more outside inter-actors (called actors) together with actions performed by the system.

    b. Number of Classes – Enter the number of classes. Classes are the embodiment of key business concepts, which can be expressed in the vocabulary of the business domain. A class can model a physical real-world entity or an abstract concept. Classes contain information, support behavior and maintain relationships with other classes. Classes enforce business rules and provide integrity checking. The UML specification states that a class is the descriptor for a set of objects with similar structure, behavior, and relationships. Also describe the intention of the class, that is, the rules that define it.

c. Number of Web Pages – Enter the number of web pages developed to support this CSCI or project. If dynamic web pages are used, only count the base template as one web page.

d. Number of Interfaces –Enter the number of interfaces. Interfaces provide a well-defined programming specification to access the capabilities of a component, subsystem, or class. An interface separates external services from internal implementation, and defines a protocol that permits polymorphic design. The UML specification states that an interface is a specifier for the externally visible operations of a class, component, or other classifier (including subsystems) without specification of internal structure. Each interface often specifies only a limited part of the behavior of an actual class. Interfaces do not have implementation. They lack attributes, states, or associations; they only have operations. (An interface may be the target of a one-way association, however, but it may not have an association that it can navigate). Interfaces may have generalization relationships. An interface is formally equivalent to an abstract class with no attributes and no methods and only abstract operations, but interface is a peer of class within the UML metamodel.

14. Estimated Effort. – Enter the estimated hours and cost associated with developing, designing, coding, integrating and testing this CSCI or project.

15. Software Size Remarks: - Describe the methodology used to estimate this CSCI or project. Identify analogous data points used for this estimate and provide their corresponding metrics.

16. Programming Language – Enter the programming language(s) that are planned for use in developing this CSCI.

17. COTS/GOTS - Replicate elements 17a -17e for each COTS/GOTS package used in this CSCI or project.

    a. Package Name - Enter the name of the COTS package planned for use in this CSCI.

    b. Version Number - Enter the version number of the COTS package.

    c. Number of Licenses - Enter the number of licenses that will be required for the COTS package

    d. Cost per License - Enter the cost per license for the l COTS package

    e. Yearly Maintenance Cost – Enter the yearly maintenance cost associated with the COTS package

18. Remarks - This space is provided to include any comments necessary to clarify the entries on this form.

2.2 Format 2B, Technical Report: Actuals (Due at end of increment)

2.2.1 Section A: CSCI/Project Information

1. Contractor: Enter contractor's name and company division in 1a, primary work location in 1b, contractor role (prime/subcontractor) in 1c, and designated contract number from the procuring agency in 1d.

2. Schedule: Enter in 2a year, month, and day of the estimated/actual start date for increment being reported. Enter in 2b year, month, and day of the estimated/actual completion date for increment being reported. Enter in 2c the date of report completion. Enter in 2d the current version number of the software being developed if applicable.

3. Name of Computer Software Configuration Item (CSCI) or Project: Enter the name of the CSCI, program, project or system being reported.

4. Corresponding CBS Number: Enter the corresponding contract breakdown structure (CBS) number for the software configuration item (CSCI) being reported.

2.2.2 Section B: Process Information

5. Application Type. - If CSCIs are designated, describe the planned primary generic function for the CSCI being reported, otherwise describe the planned primary generic function of the program. Possible descriptions could include: Web, MIS, Database, Signal Process, Analytical, Cryptologic, etc.

6. Development Method. - Describe the planned development method(s). Possible descriptions could include: Evolutionary, Incremental, COTS Integration, Prototype, Rapid Application Development, Spiral, Waterfall, etc.

7. Percent Code Designed for Re-Use. - Enter the percentage of code that that was developed for reuse in the future on other systems. Use in subsequent releases on the existing system shall not be considered reuse. If no formal requirement is documented for the CSCI or project, requiring the development of reusable code, enter 0%.

8. CMM/CMMI Level. –Enter the SEI CMM/CMMI level, at the time of report generation, for the primary contractor responsible for the CSCI or project. Possible responses are: Level 1, Level 2, Level 3, Level4, Level 5 ONLY.

9. CMM/CMMI Level based on Self-Assessment or External Assessment. –Enter Self-Assessment or External-Assessment to support SEI level entered in Format 2, Section B element 8.

2.2.3. Section C: Requirements

10a. Total Number of Requirements. - Enter an integer number of actual total requirements implemented by this CSCI or project.

10b. Requirements Volatility – Describes the requirements volatility for the CSCI or project. Shall describe the volatility by entering one of the following definitions:

Extra High – Requirement Volatility >25%
Very High – Requirement Volatility > 15% and <= 25%
High+ - Requirement Volatility > 10% and <= 15%
High – Requirement Volatility > 3% and <= 10%
Nominal – Requirement Volatility > 1% and <= 3%
Low – Requirement Volatility <= 1%

Percentages used to quantify the requirements volatility for the CSCI or project shall be calculated based on the equation below.

$$requirementvolatility = \left(\frac{added + deleted + changed}{total\,\#\,ofrequirements} \right) * 100$$

Requirements volatility represents the total change traffic applied to an agreed-to release plan. It can be greater than 100% if more changes occurred than the number of requirements originally planned for the release.

2.2.4 Section D : Software Metrics - Complete number 11 to describe the primary approach used to measure the size of this CSCI or project. Provide SLOC and Object Oriented (00) metrics in numbers 12 and 13.

11. Project Metrics - Describe the method used to measure the size of this CSCI or project.

    a. Sizing Methodology - Enter the primary software sizing method (e.g. COSMIC, Mark II, IFPUG, Top-Down, Bottom-Up, ObjectMetrix, Object Points, SLOC count, etc.), used during the development of this CSCI or project.

    b. Unit of Measure – Shall enter the unit of measure (e.g. function points, SLOC, Use Cases, Object Points, etc.) for the methodology being used.

    c. Size - Shall enter the actual software size in the same units-of-measure indicated in 11b.

12. SLOC Metrics - Quantify the SLOC for this CSCI. If the unit of measure provided in 11b above is SLOC then the information in 12a should be consistent with the size provided in 11c.

    a. SLOC-Count Technique - Enter the methodology used to measure the SLOC generated for this CSCI or project. If the method used to estimate SLOC is a conversion of the size metric provided in 11c, the conversion algorithm shall be provided in #15, Software Size Remarks. If a code-counting tool was used to measure SLOC, reference what tool was used. 12b. New Source Lines of Code (SLOC) - Enter the actual new source lines of code created for this CSCI. The SLOC entered for this element shall be consistent with the SLOC provided by programming language in section 16A of this form.

    b. Re-Used SLOC - Enter the actual re-used source lines of code for this CSCI. The SLOC entered for this element should equal the summation of the SLOC provided by programming language in section 17C of this form.

13. 00 Metrics - Provide the additional sizing metrics in numbers 13a through 13d if they are applicable to this CSCI or project.

    a. Number of Use Cases – Enter the number of use cases. A use case describes a dialogue between an external actor and the system. A use case comprises a sequence of activities performed by an end user in the course of their work that results in measurable business benefit for the user. Essentially it forms a complete business transaction. The (Unified Modeling Language UML) specification states that a use case is a kind of classifier representing a coherent unit of functionality provided by a system, as manifested by sequences of messages exchanged among the system and one or more outside inter-actors (called actors) together with actions performed by the system.

    b. Number of Classes – Enter the number of classes. Classes are the embodiment of key business concepts, which can be expressed in the vocabulary of the business domain. A class can model a physical real-world entity or an abstract concept. Classes contain information, support behavior and maintain relationships with other classes. Classes enforce business rules and provide integrity checking. The UML specification states that

a class is the descriptor for a set of objects with similar structure, behavior, and relationships. Also describe the intention of the class, that is, the rules that define it.

c. Number of Web Pages – Enter the number of web pages developed to support this CSCI or project. If dynamic web pages are used, only count the base template as one web page.

d. Number of Interfaces –Enter the number of interfaces. Interfaces provide a well-defined programming specification to access the capabilities of a component, subsystem, or class. An interface separates external services from internal implementation, and defines a protocol that permits polymorphic design. The UML specification states that an interface is a specifier for the externally visible operations of a class, component, or other classifier (including subsystems) without specification of internal structure. Each interface often specifies only a limited part of the behavior of an actual class. Interfaces do not have implementation. They lack attributes, states, or associations; they only have operations. (An interface may be the target of a one-way association, however, but it may not have an association that it can navigate). Interfaces may have generalization relationships. An interface is formally equivalent to an abstract class with no attributes and no methods and only abstract operations, but interface is a peer of class within the UML metamodel.

14. Actual Effort – Enter the actual effort in hours and dollars for this CSCI or project. This information must correspond to the lowest level CSCI breakout in Format 1. For example, the hours and dollars for CSCI 1 include WBS elements 1.2.1.1.1 through 1.2.1.1.6.

15. Software Size Remarks - Provide any amplifying comments necessary to clarify the information provided in Format 2B.

16. Programming Language - Provide System Measures (16A), OO Measures (16B) and Re-Use Measures (16C) for each language (i.e. programmatic, shell, scripting, html, embedded, etc.) used to develop this CSCI or project. 16B, OO Measures, is only required if the language is object oriented (i.e., Java, C++, etc.). ***If the automated version of this DID is not used, replicate sections 16A, 16B and 16C for each programming language.*** Otherwise, the automated spreadsheet provided will facilitate the input of data by language.

Example: A program consists of two languages, JAVA and C++, with no reused code for JAVA and reused code for C++. Elements 16A-16C will be replicated twice, once for each language. For Java, element 16A and 16B will be completed with the actual values and element 16C will be completed with N/A. For C++ code, element 16A will be completed and element 16C will be completed for the reused part of the C++ code.

16A. System Measures - This element reports the total system measures by language for the entire CSCI or project, including reused code.

i. Number of Files. – Enter the number of files in the system for the specified language.

ii. Delivered Source Instructions – Enter the sum of all Delivered Source Instructions (DSI) in the system. DSI includes executable and declarative lines of code such as control statements, mathematical statements, conditional statements, input/output & formatting, data declarations, function declarations, etc.

iii. Comment Lines - Enter the sum of all lines of code in the system that contains some form of comments.

iv. Comment Density – Enter the sum of all comment lines that actually contain a comment (i.e. words) and are not simply separators.

v. Blank Lines - Enter the sum of all Blank lines of code in the system that contain no instruction or comment characters.

vi. Total Lines –Enter the total number of all lines.

16B. OO Measures - This element, if applicable, reports the TOTALS by language for the entire CSCI or project, including reused code. 16B parameters only apply if object oriented code was used for this CSCI or project.

i. Number of Classes – Enter the number of classes. Classes are the embodiment of key business concepts, which can be expressed in the vocabulary of the business domain. A class can model a physical real-world entity or an abstract concept. Classes contain information, support behavior and maintain relationships with other classes. Classes enforce business rules and provide integrity checking. The UML specification states that a class is the descriptor for a set of objects with similar structure, behavior, and relationships. Also describe the intention of the class, that is, the rules that define it.

ii. Number of Sub-Systems – Enter the number of subsystems. A subsystem is a logical partitioning of the business domain. It is a grouping of related system software that supports one or more complete business processes. A subsystem provides a meaningful and useful subset of end user functionality and typically supports the job activities for one or more user roles within an organization. A subsystem typically provides a consistent and well-formed user interface. The UML specification states that a subsystem represents a behavioral unit in the physical system, and hence in the model. A subsystem offers interfaces and has operations, and its contents are partitioned into specification and realization elements. The specification of the subsystem consists of operations on the subsystem, together with specification elements such as use cases and stat machines. Apart from defining a namespace, a subsystem serves as a specification unit for the behavior of its contained model elements.

iii. Number of Components – Enter the number of components. A component is a physical and replaceable part of a system that conforms to, and implements, a set of interfaces. Components form the building blocks for systems. They can encapsulate environment specific details, implement business frameworks or wrap traditional technology. The UML specification states that a component represents a modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces. A component conforms to the interfaces that I exposes, where the interfaces represents services provided by elements that reside on the component. One or more artifacts, such as binary, executable, or script files, may implement a component. A component may be deployed on a node.

16C. Re-Use Measures - This element is strictly for reused code; code used but not developed for this CSCI or project. Elements 16C is Not Applicable if no reused code exists for a specific language. Do not include COTS packages in these totals. Code is not considered reused simply because it is used in subsequent releases on the existing system. Element 16C refers only to code developed outside this CSCI or project. Element 16C is a subset of the totals reported in element 16A.

i. Number of Files. – Enter the number of files in the system for the specified language.

ii. Delivered Source Instructions – Enter the sum of all Delivered Source Instructions (DSI) in the system. DSI includes executable and declarative lines of code such as control statements, mathematical statements, conditional statements, input/output & formatting, data declarations, function declarations, etc.

iii. Comment Lines - Enter the sum of all lines of code in the system that contains some form of comment.

iv. Comment Density - Enter the sum of all comment lines that actually contain a comment (i.e. words) and are not simply separators

v. Blank Lines – Enter the sum of all blank lines in the system that contain no instruction or comment characters.

vi. Total Lines. –Enter the total number of source lines of code.

vii. Percent Re-design effort . – Enter the amount of any re-design, re-work or coding modification done to the reused package for this CSCI/project. Express the amount as a percentage of the total code base of the reused package. For example, if the reuse package contains 1000 LOC and 125 LOC were modified then 12.5% would be entered ((# LOC modified / Total # of LOC) * 100)

17. COTS/GOTS - Replicate elements 17a -17e for each COTS/GOTS package used in this CSCI or project.

    a. Package Name - Enter the name of the COTS package planned for use in this CSCI.

    b. Version Number - Enter the version number of the COTS package.

    c. Number of Licenses - Enter the number of licenses that will be required for the COTS package

    d. Cost per License - Enter the cost per license for the l COTS package

    e. Yearly Maintenance Cost – Enter the yearly maintenance cost associated with the COTS package

18. Remarks - This space is provided to include any comments necessary to clarify the entries on this form.

The remainder of this page intentionally left blank

3. Format 3 - Technical Report: COTS and Custom Hardware

3.1. Format 3A Technical Reports: COTS Hardware

1. Program: Name of the program, project, or system.

2. Contractor: Provide name of prime contractor and name of hardware contractor if different.

3. Prime Contract Number - Enter the designated prime contract number from the procuring agency.

4. Dollars In: Denote entries that are factored down by thousands (K) or millions (M).

5. COTS Hardware Information - The following elements provide details for each hardware item with a unit cost greater than or equal to $2,000

   a. CBS No: Indicate the CBS number where the HWCI is used.

   b. Element: Provide the Hardware Item name.

   c. Part Description: Provide the hardware nomenclature at appropriate indentures of the CBS (e.g., Command & Control (CBS element)/Server (HWCI) /Sun 450 (Specific Part).

   d. Supplier: Identify the supplier/vendor of the item.

   e. Quantity: Indicate the quantity required to satisfy the function, including initial spares.

   f. Unit Cost: Provide the cost for each item.

   g. Total Cost: Calculation of total cost based on quantity and unit cost provided.

   h. Year: Indicate the year in which the item will be purchased or acquired.

   i. Parameters: Provide parameters/values for key technical descriptors or descriptive data for each element (e.g., for the server example above, this could be processor speed, # of processors, memory.) Some examples of desired parameters by hardware type are shown in the table below. Other parameters may also be provided and other relevant parameters may be added for hardware items not shown in the table (e.g., switches, routers).

PART DESCRIPTION	PARAMETERS
Server	Processor speed, # of processors, memory
Workstation	Processor speed, max # of processors, hard drive capacity
Storage Device (note tape vs. disk)	Tape or disk, storage capacity (GB), rpm, IO (request/sec)

3.2. Format 3B Technical Reports: Custom Hardware - A Custom Hardware Data Sheet should be completed for all non-COTS/GOTS hardware elements of the proposed system. The elements listed on the Custom Hardware Data Sheet are provided as guidance for an example of a newly developed electronic box/chassis (terms used interchangeably here). Provide a similar

level of detail for any newly developed hardware elements (i.e., relevant descriptive parameters). Enter N/A for any item within the data sheet that is not applicable to a particular entry. Use as many sheets as necessary to incorporate all boards within a box; use a separate sheet for each box. The terminology used herein is that boards roll up into boxes, which roll up into racks, which roll up into subsystems, which roll up into ground elements. Costs at each level should be traceable up to the next higher level.

1. Program: Name of the program, project, or system.

2. CBS Number: Indicate the CBS number where the HWCI is used.

3. Contractor: Provide name of prime contractor and name of hardware contractor if different.

4. Rack Name: Provide the name of the rack into which the boxes roll up (e.g., telemetry processor)

5. Box Name: Provide the name of the electronic box being developed (e.g., mass memory unit)

6. Application: Briefly describe the function of the hardware item being developed

Product Description

7. Identify Rack versus Chassis Power Supply: Does the item have a built-in power supply or does it obtain power from a rack-mounted power supply?

   a. Installed Power Supply (watts): Identify the source and amount of power required. Also identify vendor and part number

   b. Total Populated Weight: Identify total weight of the rack/item in pounds

   c. Rack Dimensions (inches): Identify dimensions of the rack in inches.

8a. Box Dimensions (inches): Identify dimensions of the box in inches.

8b. Installed Power Supply (watts): Provide installed power in watts.

9. Custom Boards/Cards (complete one column for each board/card)

   a. Board name: Enter the name of the board.

   b. Number of boards/cards per box: Enter the number of boards or cards per box.

   c. Typical Power (watts): Amount of power used per board for heat dissipation.

   d. Board/Card Dimensions (inches): Enter the dimensions for the board or card in inches.

   e. Parts Technology Allocation (Percent of Area): Provide the percent of total board area that each part technology occupies (e.g., 50% analog, 13% VHSIC/VLSI, 2.2% SAW-Crystal, .8% GaAs MMIC C band, 34% unoccupied)

   f. Number of board layers: Enter the number of board layers.

   g. Input type and speed (Mbps max): Analog or digital input? Identify key signal characteristics (e.g., bandwidth for analog; speed for digital).

h. Output type and speed (Mbps max): Analog or digital output? Identify key signal characteristics (e.g., bandwidth for analog; speed for digital).

i. Field Programmable Gate Array (FPGA) vendor: Provide vendor, if board includes FPGA(s).

j. FPGA part number: Enter the FPGA part number.

k. FPGA cost: Enter the FPGA cost (thru G&A, not including Fee or COM)

l. FPGA gates/pins: Enter the FPGA gates and PINS

m. Embedded Computer vendor: Provide vendor, if board includes an embedded computer.

n. Embedded Computer part number: Enter the part number for the embedded computer.

o. Embedded Computer cost: Enter the cost for the embedded computer (thru G&A, not including Fee or COM)

10. COTS Boards: (complete entries below if the box includes COTS boards, as well as custom boards)

a. Vendor: Enter the name of the vendor for the COTS board(s).

b. Part Number: Enter the part number for the COTS board(s).

c. Number of boards per box: Enter the number of boards per box.

d. Cost per board: Enter the cost per board (thru G&A, not including Fee or COM).

11. Board/Module Cost: Total cost of each board (thru G&A, not including Fee or COM).

12. Box Cost: Roll up of cost of all boards in the box named above (thru G&A, not including Fee or COM).

13. Rack Cost: Roll up of cost of all boxes in the rack named above (thru G&A, not including Fee or COM).

4. End of DI-FNCL-80166B

COST BREAKDOWN REPORT Format 1

SECTION A

1. CONTRACTOR	A. NAME & DIVISION	B. LOCATION (Address and Zip Code)	4. REPORT PERIOD A. FROM (YYMMDD)
2. CONTRACT	A. NAME B. NUMBER	C. DESCRIPTION	D. CONTRACT TYPE (select from list) B. TO (YYMMDD) 5. DATE SUBMITTED (YYMMDD)
3. PROGRAM	A. NAME	B. DESCRIPTION	6. DOLLARS IN (select from list) 7. PRICE TO DATE

SECTION B

CBS # & Level						NOMENCLATURE	Labor Hours	Labor $	Material $	ODC $	Total $
1	2	3	4	5	6						
0.0						Total System	0		0	0	0
	1.0					Prime Mission Product (PMP)	0		0	0	0
		1.1				Hardware Subsystem	0		0	0	0
			1.1.1...98			HWCI 1...n (Specify Name for HWCI x)					0
			1.1.99			Integration, Assembly, & Checkout of subsystem a...w HWCI's					0
		1.2				PMP Application Software	0		0	0	0
			1.2.1			Application Software Build 1 (Specify Name for Build/Spiral)	0		0	0	0
				1.2.1.1		CSCI 1 (Specify Name for CSCI 1)	0		0	0	0
					1.2.1.1.1	CSCI Requirements Analysis					0
					1.2.1.1.2	CSCI Design					0
					1.2.1.1.3	CSCI Coding and Design Entity Testing					0
					1.2.1.1.4	CSCI Design Entity Integration and Test					0
					1.2.1.1.5	Configuration Item Testing					0
					1.2.1.1.6	CSCI COTS Software					0
				1.2.1.2		CSCI 2 (Specify Name for CSCI 2)					0
				1.2.1.3...98		CSCI 3...n (Specify Name for CSCIs 3...n)					0
				1.2.1.97		System Engineering of Application S/W Build 1...9 CSCI's					0
				1.2.1.98		Program Management of Application S/W Build 1...9 CSCI's					0
				1.2.1.99		Integration & Checkout of Application S/W Build 1...9 CSCI's					0
			1.2.2...98			Application Software Build 2...n (Specify Name for Build/Spiral)					0
			1.2.99			Integration, Assembly, Test & Checkout of All Application S/W Builds					0
		1.3				PMP System Software	0	0	0	0	0
			1.3.1			System Software Build 1 (Specify Name for Build/Spiral)	0	0	0	0	0
				1.3.1.1		CSCI 1 (Specify Name for CSCI 1)	0	0	0	0	0
					1.3.1.1.1	CSCI Requirements Analysis					0
					1.3.1.1.2	CSCI Design					0
					1.3.1.1.3	CSCI Coding and Design Entity Testing					0
					1.3.1.1.4	CSCI Design Entity Integration and Test					0
					1.3.1.1.5	Configuration Item Testing					0
					1.3.1.1.6	CSCI COTS Software					0
				1.3.1.2		CSCI 2 (Specify Name for CSCI 2)					0
				1.3.1.3...98		CSCI 3...n (Specify Name for CSCIs 3...n)					0
				1.3.1.97		System Engineering of System S/W Builds 1...9 CSCI's					0
				1.3.1.98		Program Management of System S/W Builds 1...9 CSCI's					0
				1.3.1.99		Integration & Checkout of System S/W Builds 1...9 CSCI's					0
			1.3.2...98			System Software Build 2...n (Specify Name for Build/Spiral)					
			1.3.99			Integration, Assembly, Test & Checkout of All System S/W Builds					0
		1.4				PMP Integration, Assembly, Test & Checkout					0
	2.0					System Engineering/Program Management	0	0	0	0	0
		2.1				System Engineering					0
		2.2				Program Management					0
		2.3				Mission Assurance Planning and Assessment					0
		2.4				Information Assurance					0

1	CBS # & Level 2	3	4	5	6	NOMENCLATURE	Labor Hours	Labor $	Material $	ODC $	Total $
	1.0					System Integration, Assembly, Test & Checkout					0
	4.0					System Test and Evaluation	0	0	0	0	0
		4.1				DT&E					0
		4.2				OT&E					0
		4.3				Test & Eval Support					0
		4.4				Test Facilities					0
		4.5				Independent Verification and Validation					0
	5.0					Training					0
	6.0					Data					0
	7.0					Peculiar Support Equipment					0
	8.0					Common Support Equipment					0
	9.0					Operational/Site Activation					0
	10.0					Industrial Facilities					0
	11.0					Initial Spares & Repair Parts					0
	12.0					Government System Engineering/Program Management	0	0	0	0	0
		12.1				System Engineering					0
		12.2				Program Management					0
		12.3				Mission Assurance Planning					0
		12.4				Information Assurance					0
	13.0					Operations & Support	0	0	0	0	0
		13.1				Hardware Maintenance	0	0	0	0	0
			13.1.1			Licenses/Vendor Maintenance Agreements (COTS/GOTS)					0
			13.1.2			Spares/Repair Parts					0
			13.1.3			Fixes/Repairs					0
			13.1.4			Minor Enhancements					0
		13.2				Software Maintenance	0	0	0	0	0
			13.2.1			Licenses/Vendor Maintenance Agreements (COTS/GOTS)					0
			13.2.2			Fixes					0
			13.2.3			Minor Enhancements					0
			13.2.4			Security Testing and Accreditation					0
		13.3				Data Maintenance					0
		13.4				Unit/Site Operations	0	0	0	0	0
			13.4.1			Personnel					0
			13.4.2			Consumables					0
			13.4.3			Infrastructure Maintenance					0
			13.4.4			System Security					0
			13.4.5			Recurring Training					0
		13.5				Replenishment Spares					0
		13.6				Software Engineering					0
		13.7				Communications					0
		13.8				Interim Contractor Logistics Support					0
		13.9				Contractor Logistics Support					0
		13.10				Mission Assurance Peculiar Activity Support					0
		13.11				Other					0
	14.0					Disposal					0

Figure 1 – Format 1 : Cost Breakdown Structure

FORMAT 2A - TECHNICAL REPORT: ESTIMATES (Due at start of each increment)					
Section A: CSCI/Project Information					
1. CONTRACTOR					
a. Name & Division	b. Location		c. Contractor Role	d. Prime Contract Number	
2. SCHEDULE					
a. From (YY/MM/DD)	b. To (YY/MM/DD)		c. Date Submitted	d. Version Number	
3. Name of Computer Software Configuration Item (CSCI) or Project:					
4. Corresponding CBS Number:					
Section B: Process Information/Requirements					
5. Application Type	6. Development Method		7. Percent Code Designed for Reuse		
8. CMM/CMMI Level:					
9. CMM/CMMI Level based on self assessment or external assessment?					
10. Total Number of Requirements					
Section C: Software Metrics					
11. Project Metrics			13. OO Metrics		
a. Sizing Methodology			a. Num of Use Cases		
b. Unit of Measure			b. Num of Classes		
c. Size			c. Num of Web Pages		
			d. Num of Interfaces		
12. SLOC Metrics					
a. SLOC-Count Technique			14. Estimated Effort	Hours	Dollars
b. NEW SLOC					
c. Re-Used SLOC					
15. Software Size Remarks:					
16. Programming Language:					
a. Primary					
17. COTS/GOTS					
a. Package Name:			b. Version #:		
c. Number of Licenses:		d. Cost Per License:	e. Yearly Maintenance Cost:		
18. Remarks:					

Figure 2 - Format 2A: Software Technical Report: Estimate

FORMAT 2B - TECHNICAL REPORT: ACTUALS (Due at end of each increment)						
Section A: CSCI/Project Information						
1. CONTRACTOR						
a. Name & Division	b. Location		c. Contractor Role		d. Prime Contract Number	
2. SCHEDULE						
a. From (YY/MM/DD)	b. To (YY/MM/DD)		c. Date Submitted		d. Version Number	
3. Name of Computer Software Configuration Item (CSCI) or Project:						
4. Corresponding CBS Number:						
Section B: Process Information						
5. Application Type	6. Development Method		7. Percent Code Designed for Reuse			
8. CMM/CMMI Level:						
9. CMM/CMMI Level based on self assessment or external assessment?						
Section C: Requirements						
10. Total Number of Requirements						
11. Requirements Volatility						
Section D: Software Metrics						
11. Project Metrics			13. OO Metrics			
a. Sizing Methodology			a. Num of Use Cases			
b. Unit of Measure			b. Num of Classes			
c. Size			c. Num of Web Pages			
			d. Num of Interfaces			
12. SLOC Metrics						
a. SLOC-Count Technique			14. Actual Effort	Hours	Dollars	
b. NEW SLOC						
c. Re-Used SLOC						
15. Software Size Remarks:						
16. Programming Language:						
A. System Measures		B. OO Measures		C. Re-use Measures		
i. # of Files		i. # of Classes		i. # of Files		
ii. # of Delivered Source Instructions		ii. # of Subsystems		ii. # of Delivered Source Instructions		
iii. # of Comment Lines		iii. # of Components		iii. # of Comment Lines		
iv. Comment Density				iv. Comment Density		
v. # of Blank Lines				v. # of Blank Lines		
vi. Total # of Lines of Code				vi. Total # of Lines of Code		
				vii. Percent Re-design effort		
17. COTS/GOTS						
a. Package Name:				b. Version #:		
c. Number of Licenses:		d. Cost Per License:		e. Yearly Maintenance Cost:		
18. Remarks:						

Figure 3 - Format 2B: Software Technical Report: Actuals

FORMAT 3A - COTS HARDWARE INFORMATION

All Top level Major Hardware Configuration Items (HWCI)

Program:

Contractor:

WBS No.	Element	Part Description	Supplier	QTY	Unit Cost	Total Cost	Year	Parameters
	COTS Hardware Item #1	server						processor speed, # of processors, memory
	COTS Hardware Item #2	workstation						processor speed, max # of processors, hard drive capacity, slots, ports, bays
	COTS Hardware Item #3	router						
	COTS Hardware Item #4	integration material						
	COTS Hardware Item #5	storage device (disk/tape)						tape or disk, storage capacity (GB), rpm, IO (request/sec)
	COTS Hardware Item #6	switches						

Figure 4 - Format 3A: COTS Hardware Information

FORMAT 3B - Custom HARDWARE INFORMATION
All Top level Major Hardware Configuration Items (HWCI)

1. Program:
2. CBS Number:
3. Contractor:

4. Rack Name						
5. Box Name						
6. Application						
Product Description						
7. Identify Rack versus Chassis Power Supply						
a. Installed Power Supply (i.e. converter) (max watts)						
Vendor and part number						
Power Supply Cost						
b. Total Populated Weight (pounds)						
c. Rack Dimensions (inches WxDxH)						
8. Box						
a. Box Dimension (inches WxDxH)						
b. Installed Power Supply (max watts)						
9. Custom Boards/Cards	Board 1	Board 2	Board 3	Board 4	Board 5	Board 6
a. Board Name						
b. Number of boards/cards per box						
c. Typical Power (watts) (the value used for heat dissipation)						
d. Board/Card Dimensions (inches WxD)						
e. Parts Technology Allocation (% of Area by board)						
GaAs MMIC RF (include frequency)						
Silicon - RF (include frequency)						
SAW - Digital						
SAW - Crystal						
VHSIC/VLSI (Custom - i.e. ASICs)						
FGPAs						
Embedded Computers						
Digital - Lower complexity Silicon						
Analog - (non-RF)						
Unoccupied						
f. Number of board layers						
g. Input type and speed (Mbps max)						
h. Output type and speed (Mbps max)						
i. FPGA vendor						
j. FPGA part numbers						
k. FPGA cost						
l. FPGA gates/pins						
m. Embedded Computer vendor						
n. Embedded Computer part number						
o. Embedded Computer cost						
10. COTS Boards						
a. Vendor						
b. Part number						
c. Number of boards per box						
d. Cost per board (BYXX$K)						
11. Board/Module Cost (BYXX$k)						
12. Box Cost (BYXX$k)						
13. Rack Cost (BYXX$k)						

Figure 5 - Format 3B: Custom Hardware Information