

SOFTWARE DESCRIPTION ANNOTATED OUTLINE

(See DoD 5000.4-M for additional guidance)

GENERAL INSTRUCTIONS

Describe the characteristics of the system software. Supply requested data for both the top level and each Computer Software Configuration Item (CSCI) (and CSC when available). Information presented at the top level should apply to all the levels below.

Other data that could affect system costs should be provided at the appropriate level of detail. This includes any information not requested below but which is necessary to prepare a cost estimate. Other input data that are used in a software cost model should be included as an appendix to the Cost Analysis Requirements Description (CARD) submission.

In each question, if a response pertains only to selected software items, identify those items in the "Additional Comments" block.

Section I - Top-Level Characteristics. Above the CSCI Level. Information provided in this section should apply across the system's software, including each CSCI (and each CSC when available) and each software build.

Section II - Lower Level Characteristics. Complete for each CSCI (or each CSC when available) and each build.

SECTION I - TOP-LEVEL CHARACTERISTICS (Above CSCI Level)

1. SYSTEM REQUIREMENT VOLATILITY

a. LEVEL OF DEFINITION AND UNDERSTANDING OF SYSTEM REQUIREMENTS (*X one*)

- (1) Very little
- (2) Questionable
- (3) Fairly complete
- (4) Very complete

(5) Additional Comments

b. HOW WILL OVERALL TECHNOLOGY ADVANCES DURING DEVELOPMENT AFFECT THE PROJECT? (*X one*)

- (1) Significant advances; more than one system upgrade
- (2) Between one and three significant system modifications
- (3) Minor modifications
- (4) No changes to system or requirements

(5) Additional Comments

c. REQUIREMENTS VOLATILITY DURING DEVELOPMENT (*X one*)

- (1) No changes
- (2) Small noncritical changes
- (3) Frequent noncritical changes
- (4) Occasional moderate changes
- (5) Frequent moderate changes
- (6) Many large changes

(7) Additional Comments

2. SYSTEM INTEGRATION DIFFICULTY

a. EXPECTED LEVEL OF DIFFICULTY OF INTEGRATING AND TESTING THE CSCI'S TO THE ELEMENT LEVEL (*X one*)

- (1) Very little integration, no complex interfaces
- (2) Average degree of system integration/interface complexity
- (3) Several system interfaces, some complex
- (4) Complex, time-intensive integration process anticipated

(5) Additional Comments

3. USE OF COMMERCIAL OFF-THE-SHELF SOFTWARE (COTS)

a. EXPECTED IMPACT OF INTEGRATING COMMERCIAL OFF-THE-SHELF SOFTWARE INTO THE SYSTEM (*X one*)

- (1) Some impacts on the design/development effort to ensure that vendor-supplied COTS software interfaces correctly with the developed operational software
- (2) Few impacts created by the COTS software packages to support the operating environment of the applications software; COTS is in multiple releases and is relatively stable
- (3) No impacts; purchased software will be used only for operating environment support functions (*i.e., operating system*)

(4) Additional Comments

4. SOFTWARE SIZE ESTIMATE OF CSCIs (CSCs). (*Classify each CSCI into appropriate basing modes (e.g., space, air, etc.). Identify the low, most likely, and high (L, M, H) KSLOC estimates for each. Refer to glossary in DoD 5000.4-M for definitions.*)

MODE (1)	Total KSLOC (2)			Percent New SLOC (3)	Percent Reused SLOC (4)	Percent Modified SLOC (5)	Program- ming Language* (6)	Basis of Estimate** (7)	Reuse Library %*** (8)
	L	M	H						
a. SPACE									
b. AIR									
c. GROUND-MOBILE									
d. GROUND-FIXED									

* Computer language used.

** Basis of size estimate: analogy, function points, or other.

*** Percent added to library for future reuse of other activities.

5. ADDITIONAL SYSTEM SOFTWARE FACTORS *(Describe any additional factors that could affect the cost and/or size of the software being developed for the system.)*

SECTION II - LOWER-LEVEL CHARACTERISTICS

6. CSCI (CSC)-LEVEL CHARACTERISTICS

a. CSCI (CSC) NAME

b. FUNCTIONAL DESCRIPTION *(When available, this description should map to the functional allocation document)*

7. GENERAL INFORMATION

a. APPLICATION TYPE *(X all that apply)*

- (1) Prototype to be discarded later
- (2) Prototype to be built into delivered program
- (3) Complete stand-alone program
- (4) Component within a system
- (5) Reusable component for multiple programs
- (6) System with multiple components

b. APPLICATIONS DOMAIN *(Enter percentage of all that apply)*

- | | |
|---|--|
| <input type="checkbox"/> (1) Command and Control | <input type="checkbox"/> (8) Environment/Tools |
| <input type="checkbox"/> (2) Graphics, Image Processing | <input type="checkbox"/> (9) Training Software |
| <input type="checkbox"/> (3) Communications | <input type="checkbox"/> (10) Other Support Software |
| <input type="checkbox"/> (4) Signal Processing | <input type="checkbox"/> (11) Avionics |
| <input type="checkbox"/> (5) Process Control | <input type="checkbox"/> (12) Other <i>(Specify)</i> |
| <input type="checkbox"/> (6) Interface Systems | |
| <input type="checkbox"/> (7) Test Systems | |

(7) Additional Comments

(13) Additional Comments

c. SOURCE CODE MIX *(Enter percentage of all that apply)*

- | | | |
|--|--|---|
| <input type="checkbox"/> (1) Operating Systems | <input type="checkbox"/> (4) Mathematical Operations | <input type="checkbox"/> (7) String Manipulation |
| <input type="checkbox"/> (2) Real-Time Command & Control | <input type="checkbox"/> (5) Interactive Operations | <input type="checkbox"/> (8) Other <i>(Specify)</i> |
| <input type="checkbox"/> (3) Data Storage and Retrieval | <input type="checkbox"/> (6) On-Line Communications | |

(9) Additional Comments

d. DEVELOPMENT METHOD

- (1) Ada Development
- (2) Ada Incremental
- (3) Ada Full Use
- (4) Prototype
- (5) Spiral
- (6) Traditional Incremental
- (7) Waterfall

(8) Additional Comments

e. SOFTWARE INTENDED USE *(X one)*

- (1) Embedded - identify associated hardware system(s)
- (2) Other *(Specify)*

f. SOFTWARE NOVELTY. Is this the first CSCI or CSC of its kind, or are the functions and characteristics well understood and used elsewhere in the system? *(X one)* Yes No

g. PROGRAMMING PERSONNEL CAPABILITIES AND EXPERIENCE

(1) Does programming personnel have analysis capabilities experience? *(Indicate yes or no; indicate number of years experience.)*

(3) Identify staff programming capabilities.

(2) Does programming personnel have analysis application experience? *(Indicate yes or no; indicate number of years experience.)*

(4) Identify programmer language experience *(by language and number of years experience)*

h. SOFTWARE SCHEDULE

- (1) Attach software schedule to this form
- (2) Identify start date for requirements phase

i. SCHEDULE AND STAFFING CONSTRAINTS *(X one)*

- (1) Accelerated schedule
- (2) Normal schedule
- (3) Extended schedule

j. SECURITY CLASSIFICATION *(DoDD 5200.28 (reference (g)) classification)* *(X one)*

- (1) Class D
- (2) Class C1
- (3) Class C2
- (4) Class C3
- (5) Class B1
- (6) Class B2
- (7) Class B3

k. REQUIRED DOD-STDS

- (1) Complete 2167A *(reference (h))* documentation
- (2) Subset of 2167A *(reference (h))*
- (3) Other *(Specify)*

8. BASIS OF SIZE ESTIMATE		
a. (X as applicable) <input type="checkbox"/> (1) From lower level <input type="checkbox"/> (2) Function points <input type="checkbox"/> (3) Analogy with (Specify) <input type="checkbox"/> (4) Other (Specify)	b. IF SYSTEM WAS SIZED USING FUNCTION POINTS, ENTER NUMBER OF: <input type="text"/> (1) Inputs (Unique major data types that enter the system) <input type="text"/> (2) Outputs (Unique logical major report formats generated by system) <input type="text"/> (3) Inquiries (Types of queries that result in informational searches and responses) <input type="text"/> (4) External interfaces <input type="text"/> (5) Internal files (Unique logical files/databases used by the application)	
(5) Additional Comments		
9. SYSTEM HARDWARE ENVIRONMENT		
a. AVAILABILITY OF TARGET PROCESSING HARDWARE (X one) <input type="checkbox"/> (1) To be developed; will be completed before software is ready <input type="checkbox"/> (2) To be developed under contract concurrently with software; can/will have major impact <input type="checkbox"/> (3) To be developed under contract concurrently with software; will have little impact <input type="checkbox"/> (4) No new hardware to be developed	b. VIRTUAL MACHINE VOLATILITY OF TARGET SYSTEM (Based on number of major/minor changes) (if different from development system) <input type="checkbox"/> (1) Low - major and minor changes rarely <input type="checkbox"/> (2) Medium - major changes 2/year, minor 2/month <input type="checkbox"/> (3) High - major changes 4 or more times/year, minor often	
(5) Additional Comments		
9. SYSTEM HARDWARE ENVIRONMENT		
c. TARGET SYSTEM ARCHITECTURE (If different from development system) (X one) <input type="checkbox"/> (1) Centralized (Single processor) <input type="checkbox"/> (2) Tightly coupled (Multiple processor) <input type="checkbox"/> (3) Loosely coupled (Multiple processor) <input type="checkbox"/> (4) Functional processors communicating via bus <input type="checkbox"/> (5) Distributed (Centralized database) <input type="checkbox"/> (6) Distributed (Distributed database)	d. REHOSTING IMPACT (Effort to convert from host to target system, if necessary) (X one) <input type="checkbox"/> (1) None <input type="checkbox"/> (2) Minor language and/or system change <input type="checkbox"/> (3) Major language or system change <input type="checkbox"/> (4) Major language and system change	
(7) Additional Comments		
e. MAIN STORAGE CONSTRAINT (1) Percentage of main storage expected to be used by all CSCIs or CSCs sharing main storage hardware (Refers to random access storage, such as core, integrated-circuit, or plated-wire. Excludes drums, disks, tapes or bubble storage.) (2) Additional Comments	f. EXECUTION TIME CONSTRAINTS (1) Percentage of available execution time expected to be used by all CSCIs or CSCs sharing consumption of execution time resource (2) Additional Comments	g. SOFTWARE FUNCTIONS TO BE IMPLEMENTED IN FIRMWARE (1) Percentage (2) Additional Comments
10. SOFTWARE COMPLEXITY		
a. SOFTWARE INTERFACE COMPLEXITY (1) With how many CSCIs or CSCs does this CSCI or CSC interface? (2) Additional Comments	b. EXPECTED LEVEL OF DIFFICULTY OF INTEGRATING AND TESTING COMPONENTS TO THE CSCI OR CSC LEVEL (X one) <input type="checkbox"/> (1) No internal integration <input type="checkbox"/> (2) Very little integration, no complex interfaces <input type="checkbox"/> (3) Average degree of CSCI or CSC integration and interface complexity <input type="checkbox"/> (4) Several CSCI or CSC interfaces, some complex <input type="checkbox"/> (5) Complex, time-intensive CSCI or CSC integration process anticipated (6) Additional Comments	
c. DIFFICULTY OF PROCESSING LOGIC (X one) <input type="checkbox"/> (1) Simple logic, straightforward I/O <input type="checkbox"/> (2) Difficult, highly nested logic, real-time processing <input type="checkbox"/> (3) Routine nesting, minimal interface with operating system, standard I/O <input type="checkbox"/> (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments	d. MATHEMATICAL COMPLEXITY (X one) <input type="checkbox"/> (1) Simple algorithms and simple calculations <input type="checkbox"/> (2) Majority of simple algorithms and calculations <input type="checkbox"/> (3) Algorithms and calculations of average complexity <input type="checkbox"/> (4) Some difficult or complex calculations <input type="checkbox"/> (5) Many difficult algorithms and complex calculations (6) Additional Comments	
e. DEGREE OF REAL-TIME (X one) <input type="checkbox"/> (1) No tasking, essentially batch response <input type="checkbox"/> (2) Interactive with limited (Ada) tasking <input type="checkbox"/> (3) Interrupt drive, tasking in milliseconds <input type="checkbox"/> (4) Concurrent tasking, rendezvous in milliseconds <input type="checkbox"/> (5) Concurrent tasking, rendezvous in nanoseconds (6) Additional Comments	f. PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) <input type="text"/> (1) Real-time <input type="text"/> (4) On-line <input type="text"/> (2) Time-constrained <input type="text"/> (5) Other (Specify) <input type="text"/> (3) Non-time-critical (6) Additional Comments	

g. DISPLAY REQUIREMENT (X all that apply) <input type="checkbox"/> (1) Simple I/O <input type="checkbox"/> (2) User-friendly, menu driven <input type="checkbox"/> (3) Pressure-sensitive devices (<i>touch screen, joystick</i>) (5) Additional Comments	h. SOFTWARE TESTABILITY (X one) <input type="checkbox"/> (1) Very difficult <input type="checkbox"/> (2) Difficult <input type="checkbox"/> (3) Time insensitive <input type="checkbox"/> (4) Easy (5) Additional Comments
--	--

11. SOFTWARE RELIABILITY

a. EFFECT OF SOFTWARE FAILURE (X as applicable) <input type="checkbox"/> (1) Inconvenience <input type="checkbox"/> (2) Easily recoverable loss <input type="checkbox"/> (3) Moderate loss (<i>Recoverable</i>) <input type="checkbox"/> (4) Major loss (<i>High financial loss</i>) (5) Additional Comments	b. BACKUP CONSIDERATIONS (X one) <input type="checkbox"/> (1) Data protection beyond regular backup required <input type="checkbox"/> (2) No special backup requirements <input type="checkbox"/> (3) Alternative methods need to be developed in case of software failure (4) Additional Comments	c. RECOVERY CONSIDERATIONS (X one) <input type="checkbox"/> (1) Alternative methods need to be developed in case of software failure <input type="checkbox"/> (2) No special recovery requirements (3) Additional Comments
--	---	--

11. SOFTWARE RELIABILITY

a. DATABASE SIZE (1) Kilobytes (2) Additional Comments	b. PHYSICAL DATA FILES (1) Number of Files (2) Additional Comments	c. DATABASE COMPLEXITY (X one) <input type="checkbox"/> (1) Simple data, few files, low capacity <input type="checkbox"/> (2) Simple, numerous variables <input type="checkbox"/> (3) Multiple files, fields data interactions <input type="checkbox"/> (4) Complex file structure <input type="checkbox"/> (5) Highly complex (6) Additional Comments
---	---	---

13. SOFTWARE REUSE (If applicable)

a. LOGICAL COMPLEXITY OF CODE REUSED FROM OTHER PROGRAMS (X one) <input type="checkbox"/> (1) Simple algorithms and simple calculations <input type="checkbox"/> (2) Majority of simple algorithms and calculations <input type="checkbox"/> (3) Algorithms and calculations of average complexity <input type="checkbox"/> (4) Some difficult or complex calculations <input type="checkbox"/> (5) Many difficult algorithms and complex calculations (6) Additional Comments	b. STRUCTURAL COMPLEXITY OF CODE REUSED FROM OTHER PROGRAMS (X one) <input type="checkbox"/> (1) Nonprocedural (<i>Generated, query, spreadsheets, etc.</i>) <input type="checkbox"/> (2) Well structured with usable modules <input type="checkbox"/> (3) Fair structure, some complex paths and modules <input type="checkbox"/> (4) Poor structure, many complex paths and modules (5) Additional Comments
c. COMPLEXITY OF DATABASE REUSED FROM OTHER PROGRAMS (If applicable) <input type="checkbox"/> (1) Simple data, few variables, little complexity <input type="checkbox"/> (2) Several data elements, simple data relationships <input type="checkbox"/> (3) Multiple files, switches, and data interactions <input type="checkbox"/> (4) Complex data elements, complex data interactions <input type="checkbox"/> (5) Very complex data elements and interactions (6) Additional Comments	d. IF PLANNING TO REUSE THIS CSCI IN ANOTHER PROGRAM, SELECT INTENDED USE (X one) <input type="checkbox"/> (1) None <input type="checkbox"/> (2) Reuse within element <input type="checkbox"/> (3) Reuse across element <input type="checkbox"/> (4) Reuse in another DoD program application (5) Additional Comments

14. SOFTWARE MAINTENANCE

(1) Indicate number of years maintenance will be required (2) Indicate number of separate maintenance sites (3) Indicate estimated maintenance/software growth over life	(4) Indicate annual change rate for software (5) Additional Comments
--	---

15. ADDITIONAL CSCI (CSC) FACTORS (Describe any additional factors that could affect the cost and/or size of the CSCI/CSC software being developed (e.g., known contractor-specific information))