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 Characteristic			•						
		I - TOP-I	LEVEL CHAR	ACTERISTIC	S (Above CS	CI Level)			
1. SYSTEM REQUIREMENT VOLATIL	ITY								
a. LEVEL OF DEFINITION AND UNDERSTANDING OF SYSTEM REQUIREMENTS (X one) b. HOW WILL OVERALL TECHNOLOGY ADVANCES DURING DEVELOPMENT AFFECT THE PROJECT? (X one)							lG		
(1) Very little (1) Significant advances; more than one system upgrade									
(2) Questionable (2) Between one and three significant system modifications								ıs	
(3) Fairly complete				(3) Min	or modificatio	ns			
(4) Very complete				(4) No	changes to sy	stem or requirem	nents		
(5) Additional Comments				(5) Addition	al Comments	3			
c. REQUIREMENTS VOLATILITY DUR	ING DEVEL	OPMEN	T (X one)	2. SYSTEM	I INTEGRATI	ON DIFFICULTY	•		
(1) No changes				a. EXPECT	ED LEVEL O	F DIFFICULTY C	OF INTEGRATING A	ND	
(2) Small noncritical changes				TESTING	G THE CSCI'S	S TO THE ELEM	ENT LEVEL (X one)		
(3) Frequent noncritical changes				(1) Ver	y little integra	tion, no complex	interfaces		
(4) Occasional moderate changes				(2) Ave	rage degree	of system integra	tion/interface comple	exity	
(5) Frequent moderate changes				(3) Sev	eral system in	nterfaces, some o	complex		
(6) Many large changes				(4) Cor	nplex, time-in	tensive integratio	n process anticipate	d	
(7) Additional Comments				(5) Addition	al Comments	3			
3. USE OF COMMERCIAL OFF-THE-SHELF SOFTWARE (COTS)									
a. EXPECTED IMPACT OF INTEGRAT				IELF SOFTW	ARE INTO TH	IE SYSTEM (X o	ne)		
(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 Total KSLOC Percent New			Percent Reused	Percent Modified	Program- ming	Basis of	Reuse Library		
(4)	(2)		SLOC	SLOC	SLOC	Language*	Estimate**	%***	
(1)	L M	Н	(3)	(4)	(5)	(6)	(7)	(8)	
a. SPACE									
b. AIR									
c. GROUND-MOBILE									
d. GROUND-FIXED									
** Basis of size estimate: analogy, function points, or other.								1	

^{***} 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.)							
zomg deresped ist the eyelenmy							
SECTION II -	I OWER-LE	EVEL CHARACTERISTICS					
6. CSCI (CSC)-LEVEL CHARACTERISTICS		TEL STRICKS LENGTH					
a. CSCI (CSC) NAME							
b. FUNCTIONAL DESCRIPTION (When available, this descrip	tion should n	nap to the functional allocation document)					
7. GENERAL INFORMATION							
a. APPLICATION TYPE (X all that apply)	b. APPLICA	ATIONS DOMAIN (Enter percentage of all that apply)					
(1) Prototype to be discarded later		Command and Control (8) Environment/Tools					
(2) Prototype to be built into delivered program		Graphics, Image Processing (9) Training Software					
(3) Complete stand-alone program (4) Component within a system		Communications (10) Other Support Software Signal Processing (11) Avionics					
(4) Component within a system (5) Reusable component for multiple programs		Process Control (12) Other (Specify)					
(6) System with multiple components		Interface Systems					
(7) Additional Comments		Test Systems					
	(13) Additio	onal Comments					
COURSE CORE NOW (Forter and of all that are had							
c. SOURCE CODE MIX (Enter percentage of all that apply) (1) Operating Systems (4) Mathe	matical Oper	rations (7) String Manipulation					
(2) Real-Time Command & Control (5) Interactive Operations (8) Other (Specify) (3) Data Storage and Retrieval (6) On-Line Communications							
(9) Additional Comments							
d. DEVELOPMENT METHOD		e. SOFTWARE INTENDED USE (X one)					
(4) Prototype	(7) Wat						
(2) Ada Incremental (5) Spiral		(1) Ellibouded Rothing accounted has a system,					
(3) Ada Full Use (6) Traditional Incremental		(2) Other (Specify)					
(8) Additional Comments							
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? (3) Identify staff programming capabilities.							
(Indicate yes or no; indicate number of years experience.)							
(a) Development and the second base and leading and leading and		(A) Literative and a second se					
(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)							
(maissing yes at the final and							
h. SOFTWARE SCHEDULE i. SCHEDULE AND STAFFING CONSTRAINTS (X one)							
(1) Attach software schedule to this form (1) Accelerated schedule (3) Extended							
(2) Identify start date for requirements phase (2) Normal schedule schedule							
j. SECURITY CLASSIFICATION (DoDD 5200.28 (reference (g)) classification) k. REQUIRED DOD-STDS (1) Complete 2167A (reference (h)) documentation							
(1) Class D (3) Class C2 (5) Class B1	(7) C	Class B3 (2) Subset of 2167A (reference (h))					
(4) Class C3 (6) Class B2		(3) Other (Specify)					

a. (X as applicable) (I) Financion Doints (I) Financion Doints (I) Analogy with (Rportly) (I) To be developed under contract concurrently with software; with away title impact (I) Analogy with (Rportly) (I) To be developed under contract concurrently with software; with away title impact (I) Analogy with (Rportly) (8. BASIS OF SIZE ESTIMATE								·		
Q : Durptus (privile) Q :	a. (X as applicable)		b. IF S	SYSTEM WAS	SIZE	D USING FUNCT	TION POIN	ITS, EN	TER NUMBER OF:		
G) Analogy with (Speechy) G) Analogy with (Speechy) G) Analogy with (Speechy) G) Additional Comments G) Internal files (Unique logical flies/databases used by the application)	(1) From lower level			(1) Inputs (U	Inique	major data types	that enter	the syst	em)		
G) Analogy with (Speechy) G) Analogy with (Speechy) G) Analogy with (Speechy) G) Additional Comments G) Internal files (Unique logical flies/databases used by the application)	(2) Function points			(2) Outputs	Uniqu	e logical major re	port forma	ts genei	rated by system)		
4 Additional Comments 4 Additional Comments 5 Several interfaces 5 Internal files (Unique logical files/databases used by the application) 5 Several interfaces 5 Internal files (Unique logical files/databases used by the application) 5 Several interfaces 5 Internal files (Unique logical files/databases used by the application) 5 Several interfaces 5 Internal files (Unique logical files/databases used by the application) 6 Several interfaces 5 Internal files (Unique logical files/databases used by the application) 6 Several interfaces 6 Several interfaces 5 Internal files (Unique logical files/databases used by the application) 7 Several interfaces 5 Internal files (Unique logical files/databases used by the application) 7 Several interfaces 5 Internal files (Unique logical files/databases) 7 Several interfaces 5 Several interfaces 5 Several interfaces 5 Internal files (Unique logical files/databases) 7 Several interfaces 7 Several i		i i i i i i i i i i i i i i i i									
S. YSTEM HARDWARE ENVIRONMENT						4					
SYSTEM HARDWARE ENVIRONMENT				1	•	ces					
9. SYSTEM HARDWARE ENVIRONMENT (1) To be developed, will be completed before software is ready (2) To be developed, will be completed before software is ready (3) To be developed, will be completed before software is ready (3) To be developed under contract concurrently with software; can'vell have little inpact (3) To be developed under contract concurrently with software; will have little inpact (3) How major and minor changes are all (4) Additional Comments (4) Additional Comments (5) Additional Comments (4) Additional Comments (4) Additional Comments (4) Additional Comments (5) Additional Comments (4) Additional Comments (5) Additional Comments (5) Additional Comments (6) Additional Comments (7) Additional Comments (8) Ad				1 ` ′			/database	s used b	y the application)		
(2) To be developed under contract concurrently with software; answill have major impact (2) To be developed under contract concurrently with software; answill have major impact (2) Medium - major and minor changes rarely (2) Medium - major changes zyear, minor z/month will have major impact (3) To be developed under contract concurrently with software; will have filled impact (3) To be developed under contract concurrently with software; will have filled impact (3) Medium - major changes zyear, minor z/month (3) High - major changes zyear, minor z/month order of the major changes zyear, minor z/month order zyear, minor z/month order zyear, minor z/month will zyear zyear, minor z/month zyear, zy	9. SYSTEM HARDWARE ENVIRONMENT			,	•	, 0			, , ,		
(2) To be developed under contract concurrently with software; answill have major impact (2) To be developed under contract concurrently with software; answill have major impact (2) Medium - major and minor changes rarely (2) Medium - major changes zyear, minor z/month will have major impact (3) To be developed under contract concurrently with software; will have filled impact (3) To be developed under contract concurrently with software; will have filled impact (3) Medium - major changes zyear, minor z/month (3) High - major changes zyear, minor z/month order of the major changes zyear, minor z/month order zyear, minor z/month order zyear, minor z/month will zyear zyear, minor z/month zyear, zy	a. AVAILABILITY OF TARGET PROCESSING	HARDWARE (X	(one)		h \/IE	THAL MACHINE	= VOLATII	ITY OF	TARGET SYSTEM		
Caralytis have major impact Caralytis have hardware to be developed Caralytis have hardware to b	l						-				
acavWill have major impact (3) Lose developed under contract concurrently with software: will have tills impact (4) No new hardware to be developed (5) Additional Comments (6) Additional Comments (6) Additional Comments (6) Additional Comments (7) Lose training and minor changes a for more times/year, minor others (8) Legisland Comments (9) Additional Comments (1) Centralized (Single processor) (1) Centralized (Single processor) (1) Tightly coupled (Multiple processor) (2) Tightly coupled (Multiple processor) (3) Loses (coupled (Multiple processor) (4) Functional processors communicating via bus (5) Distributed (Centralized database) (6) Distributed (Centralized database) (7) Additional Comments (8) Distributed (Centralized database) (8) Distributed (Centralized database) (9) Additional Comments (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, integrigetd-crocus, or placet-wire. Excludes drums, disks, tapes or bubble storage) (9) Additional Comments (1) Software Complexity (1) Percentage of main storage and sorage integrigetd-crocus, or placet-wire. Excludes drums, disks, tapes or bubble storage) (2) Additional Comments (2) Additional Comments (3) Additional Comments (4) Additional Comments (5) Additional Comments (6) Additional Comments (7) Percentage of main storage expected to be used by all CSCIs or CSCs sharing consumption of execution time expected to be used by all CSCIs or CSCs sharing consumption of execution time resource (8) Additional Comments (9) Additional Comments (1) Percentage of main storage and system change (1) Percentage of main storage and system change (1) Percentage of available execution time expected to be used by all CSCIs or CSCs or CSCs or CSC or CSC or CSC or CSC integration and interface complexity (1) Winth how many CSCIs or CSCs does this CSCI or CSC integration and interface complexity (1) Winth how many CSCIs or CSCs or CSCs integration a			-								
3		Territy With Softw	aie,		I						
(4) No new hardware to be developed (3) High - major changes 4 or more times/year, minor often oft	, · ·	roothy with coftw	oro:								
(4) No new hardware to be developed		Territy With Softw	aie,			-	_	-			
(5) Additional Comments C. TARGET SYSTEM ARCHITECTURE (If different from development system) (X one) (1) Centralized (Single processor) (2) Tightly coupled (Multiple processor) (3) Lossely coupled (Multiple processor) (4) Functional processors communicating via bus (5) Distributed (Centralized adtabase) (6) Distributed (Centralized adtabase) (7) Additional Comments (8) Major language and system change (9) Additional Comments (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 (2) Additional Comments (2) Additional Comments (3) Major language and system change (4) Major language and system change (5) Additional Comments (6) Additional Comments (7) Additional Comments (8) Report Integration of execution time expected to be used by all time expected to be used by all cSCIs or CSCs sharing consumption of execution time resource (2) Additional Comments (2) Additional Comments (2) Additional Comments (3) Major language and/or system change (4) No internal integration of execution time expected to be used by all time expected to be used	·						iariyes 4 0	i illore t	iiiles/year, miiloi		
. TARGET SYSTEM ARCHITECTURE (if different from development system) (X one) 1 (1) Centralized (Single processor) 2 (2) Tightly coupled (Multiple processor) 3 (3) Loosely coupled (Multiple processor) 4 (4) Functional processors communicating via bus (5) Solisributed (Centralized distabase) (6) Distributed (Distributed distabases) (7) Additional Comments 8 (5) Additional Comments 8 (6) Distributed (Distributed distabase) (8) Major Inaquage and system change (9) Major Inaquage and system change (10) Percentage of main storage and system change (11) Percentage of main storage expected to be used by all CSCls or CSCs sharing main storage hardware (Refers to random access storage, such as one, integrated-circuit, or piled-wire. Excludes drums, disks, lapser or bubble storage.) (10) Percentage of main storage capacited on the used by all CSCls or CSCs sharing consumption of execution time resource (21) Additional Comments (22) Additional Comments (23) Additional Comments (24) Additional Comments (25) Additional Comments (26) Additional Comments (27) Additional Comments (28) Additional Comments (29) Additional Comments (29) Additional Comments (29) Additional Comments (20) Additional Comments (29) Additional Comments (29) Additional Comments (29) Additional Comments (29) Additional Comments (20) Additional Comments (29) Additional Comments (20) Additional Comments (21) Additional Comments (22) Additional Comments (33) Additional Comments (4) Additional Comments (5) Additional Comments (6) Additional Comments (7) Additional Comments (8) Additiona					(A) A		nte				
(1) Centralized Gingle processor)	(a) Additional Comments				(4) /(113				
(1) Centralized Gingle processor)	C TARGET SYSTEM ARCHITECTURE (IL 1995	orant from dovel	onmon'	cuctom! (V =	10)	4 DELIGOTIVO	IMPACT	/F#	a convert from the set		
(2) Tightly coupled (Multiple processor)	· ·	ərəni ironi üeveli	υριτιθετίτ	syst e lli) (A 01	1 0)						
3) Loosely coupled (Multiple processor)							em, m mece	osary) (X One)		
(3) Major language or system change (3) Major language or system change (4) Major language and system change (4) Major language and system change (4) Major language and system change (5) Distributed (Distributed distabase) (5) Additional Comments (6) Additional Comments (7) Additional Comments (7) 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 (3) Major language and system change (4) Major language and system change (4) Major language and system change (6) Additional Comments (6) Additional Comments (7) Percentage of available execution (7) Percentage of available execution (8) Percentage of available execution (8) Additional Comments (9) Additional Comments (1) Percentage (2) Additional Comments (3) Average degree of CSCI or CSC integration and interface complexity (4) Several CSCI or CSC integration process anticipated (6) Additional Comments (6) Additional Comments (6) Additional Comments (7) Simple logic, straightforward I/O (8) Several CSCI or CSC integration and interface complexity (9) Simple logic, straightforward I/O (1) Simple logic, straightforward I/O (2) Majority of simple algorithms and calculations (3) Algorithms and calculations (3) Algorithms and calculations (3) Algorithms and calculations (4) Some difficult or complex calculations (5) Many difficult algorithms and calculations (6) Additional Comments (7) Simple algorithms and calculation						I⊫ ` ′		.1/	ta an ab an an		
(s) Distributed (Centralized database) (s) Additional Comments (
(6) Distributed (Distributed database)		/ia bus						-			
e. MAIN STORAGE CONSTRAINT f. EXECUTION TIME CONSTRAINTS g. SOFTWARE FUNCTIONS TO BE IMPLEMENTED IN FIRMWARE (1) Percentage of main storage expected to be used by all CSCIs or CSCs sharing main storage hardware (<i>Refers to random access storage, such as core, integrated-circuit, or plated-wire. Excludes drums, disks, lapes or bubble storage)</i> (2) Additional Comments (2) Additional Comments (2) Additional Comments (3) Additional Comments (4) SoFTWARE INTERFACE COMPLEXITY (5) With how many CSCIs or CSC shoes this CSCI or CSC integration, no complex interfaces (6) Additional Comments (7) Additional Comments (8) Additional Comments (9) Additional Comments (1) No internal integration (1) Sort integration and interface complexity (3) Average degree of CSCI or CSC integration and interface complexity (4) Several CSCI or CSC integration, no complex (5) Complex, time-intensive CSCI or CSC integration and interface complexity (6) Additional Comments (7) Simple logic, straightforward I/O (1) Simple logic, straightforward I/O (2) Difficut, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) Additional Comments (8) Additional Comments (8) Additional Comments (9) Additional Comments (1) Simple algorithms and calculations (1) Simple algorithms and calculations (2) Majorithm sand calculations (3) Algorithms and calculations or average complexity (4) Some difficult or complex calculations (5) Many difficult algorithms and calculations (6) Additional Comments (8) Additional Comments (9) Additional Comments (1) Percentage of excultion served to be used by all CSCI or CSC integration and interface complexity (4) Some difficult or complex calculations (5) Many difficult algorithms and calculations or average complexity (1) Simple algorithms and calculations (2) Majorithm and calculations (3) Add								d syster	n change		
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 (3) Additional Comments (4) Several CSCI or CSC integration and interface complexity (5) Complex, time-intensive CSCI or CSC integration process anticipated (6) Additional Comments (7) Percentage (1) Percentage (1) Percentage (2) Additional Comments (2) Additional Comments (3) Additional Comments (4) Additional Comments (5) Complex, time-intensive CSCI or CSC integration and interface complexity (6) Additional Comments (7) Simple logic, straightforward I/O (8) Routine nesting, minimal interface with operating system, standard I/O (9) Official, highly nested logic, real-time processing (9) Routine nesting, minimal interface with operating system, standard I/O (9) Official time processing (1) No tasking, essentially batch response (2) DEGREE OF REAL-TIME (X one) (1) Interactive with limited (Ada) tasking (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in nanoseconds (5) Concurrent tasking, rendezvous in nanoseconds						(5) Additional C	omments				
(1) Percentage of main storage expected to be used by all CSCls or CSCs sharing main storage hardware (Refers to random access storage, such as core, integrated or bubble storage.) (2) Additional Comments (2) Additional Comments (2) Additional Comments (2) Additional Comments (3) Additional Comments (4) Additional Comments (5) Complex, time-integration or CSC integration and interface complexity (6) Additional Comments (7) Simple algorithms and calculations (8) Additional Comments (8) Additional Comments (8) Additional Comments (8) Additional Comments (1) Simple algorithms and calculations (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (3) Additional Comments (4) Some difficult or complex calculations (5) Additional Comments (6) Additional Comments (7) Percentage of available execution time expected to be used by all (17) Percentage (17) Per	(7) Additional Comments										
(1) Percentage of main storage expected to be used by all CSCls or CSCs sharing main storage hardware (Refers to random access storage, such as core, integrated or bubble storage.) (2) Additional Comments (2) Additional Comments (2) Additional Comments (2) Additional Comments (3) Additional Comments (4) Additional Comments (5) Complex, time-integration or CSC integration and interface complexity (6) Additional Comments (7) Simple algorithms and calculations (8) Additional Comments (8) Additional Comments (8) Additional Comments (8) Additional Comments (1) Simple algorithms and calculations (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (3) Additional Comments (4) Some difficult or complex calculations (5) Additional Comments (6) Additional Comments (7) Percentage of available execution time expected to be used by all (17) Percentage (17) Per											
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 (2) Additional Comments (2) Additional Comments (2) Additional Comments (3) Additional Comments (4) SoFTWARE INTERFACE COMPLEXITY (5) With how many CSCIs or CSCs does this CSCI or CSC integration (6) Complex, time-intensive CSCI or CSC integration process anticipated (6) Additional Comments (6) Additional Comments (7) Percentage (1) Percentage (2) Additional Comments (2) Additional Comments (3) Additional Comments (4) Several CSCI or CSC integration and interface complexity (4) Several CSCI or CSC integration process anticipated (6) Additional Comments (6) Additional Comments (7) Percentage (2) Additional Comments (8) Additional Comments (9) Very little integration, no complex interfaces (9) Additional Comments (1) Simple logic, straightforward I/O (1) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) Percentage (2) Majority of Integration Process anticipated (8) Additional Comments (9) Additional Comments (1) Simple algorithms and calculations of average complexity (1) Simple algorithms and calculations of average complexity (2) Majority of simple algorithms and calculations of average complexity (3) Algorithms and calculations of average complexity (4) Some difficult algorithms and complex calculations (5) Additional Comments (6) Additional Comments (7) Percentage (7) Additional Comments (8) Additional Comments (9) Additional Comments (1) No tasking, essentially batch response (1) No tasking, essentially batch response (2) Interactive with limited (Ada) tasking (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in	e. MAIN STORAGE CONSTRAINT		f. EXE	CUTION TIM	E CON	STRAINTS	g. SOFT\	WARE F	UNCTIONS TO BE		
to random access storage, such as core, integrated-circuit, or plated-wire. Excludes drums, disks, tapes or bubble storage.) (2) Additional Comments (2) Additional Comments (2) Additional Comments (2) Additional Comments (3) Additional Comments (4) Additional Comments (5) Additional Comments (6) Additional Comments (7) Additional Comments (8) Additional Comments (8) Additional Comments (8) EXPECTED LEVEL OF DIFFICULTY OF INTEGRATING AND TESTING COMPONENTS TO THE CSCI OR CSC LEVEL (X one) (9) Additional Comments (1) With how many CSCIs or CSCs does this CSCI or CSC interface? (1) No internal integration (2) Very little integration, no complex interfaces (3) Average degree of CSCI or CSC integration and interface complexity (4) Several CSCI or CSC integration process anticipated (6) Additional Comments (6) Additional Comments (7) Additional Comments (8) Average degree of CSCI or CSC integration and interface complexity (9) Complex, time-intensive CSCI or CSC integration process anticipated (9) Additional Comments (9) Difficult, highly nested logic, real-time processing (1) Simple algorithms and simple calculations (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (6) Additional Comments (7) Additional Comments (8) Many difficult algorithms and calculations of average complexity (9) Many difficult algorithms and complex calculations (1) No tasking, essentially batch response (1) Interactive with limited (Ada) tasking (2) Interactive with limited (Ada) tasking (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in milliseconds (5) Other (Specify) (6) Concurrent tasking, rendezvous in nanoseconds	(1) Percentage of main storage expected to be	used by all	(1) Pe	rcentage of a	vailabl	railable execution IMPLI			D IN FIRMWARE		
circuit, or plated-wire. Excludes drums, disks, tapes or bubble storage.) (2) Additional Comments (2) Additional Comments (2) Additional Comments (2) Additional Comments (3) Additional Comments (4) Some and interface complexity (5) Complex, time-intensive CSCI or CSC integration process anticipated (6) Additional Comments (6) Additional Comments (7) With how many CSCIs or CSCs does this CSCI or CSC interface? (8) Additional Comments (9) Additional Comments (1) No internal integration (1) No internal integration (2) Very little integration, no complex interfaces complexity (4) Several CSCI or CSC interfaces, some complex (6) Additional Comments (7) Additional Comments (8) Additional Comments (9) Difficult, highly nested logic, real-time processing (1) Simple algorithms and simple calculations (1) Simple algorithms and calculations (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) Mith now many CSCI or CSC integration and interface complexity (8) Additional Comments (9) Difficult, highly nested logic, real-time processing (1) Simple algorithms and calculations (1) Simple algorithms and calculations (2) Majority of simple algorithms and calculations (3) Algorithms and calculations of average complexity (4) Complex dynamic resource allocation, multiple exception handles, for Many difficult algorithms and calculations (5) Additional Comments (6) Additional Comments (6) Additional Comments (7) Majority of simple algorithms and calculations (8) Additional Comments (9) Additional Comments (1) Simple algorithms and calculations (1) Simple algorithms and calculations (2) Majority of Simple algorithms and calculations (3) Algorithms and calculations (4) Some difficult or complex calculations (5) Many difficult algorithms and calculations (6) Additional Comments (8)	· · · · · · · · · · · · · · · · · · ·	•				, (1): 0:00:.tage					
C2 Additional Comments C3 Additional Comments C4 Additional Comments			1			• •					
(2) Additional Comments (2) Additional Comments (3) Additional Comments (4) SOFTWARE INTERFACE COMPLEXITY a. SOFTWARE INTERFACE COMPLEXITY (1) With how many CSCls or CSCs does this CSCl or CSC interface? (1) No internal integration (2) Very little integration, no complex interfaces (3) Additional Comments (4) Several CSCl or CSC interfaces, some complex (5) Complex, time-intensive CSCl or CSC integration process anticipated (6) Additional Comments (6) Additional Comments (7) Additional Comments (8) Additional Comments (9) Interface Complex interface complex interface complex interface complex interfaces, some complex (5) Complex, time-intensive CSCl or CSC integration process anticipated (6) Additional Comments (6) Additional Comments (7) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Algorithms and simple calculations (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) Many difficult algorithms and complex calculations (5) Many difficult algorithms and complex calculations (6) Additional Comments (8) Additional Comments (9) Interactive with limited (Ada) tasking (1) No tasking, essentially batch response (2) Interactive with limited (Ada) tasking (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in milliseconds (5) Concurrent tasking, rendezvous in nanoseconds (6) Concurrent tasking, rendezvous in nanoseconds		ks, tapes or	of	execution time	resource (2) Additional Comments				mments		
10. SOFTWARE COMPLEXITY a. SOFTWARE INTERFACE COMPLEXITY (1) With how many CSCIs or CSCs does this CSCI or CSC interface? (2) Additional Comments (3) Average degree of CSCI or CSC integration and interface complexity (4) Several CSCI or CSC integration process anticipated (6) Additional Comments c. DIFFICULTY OF PROCESSING LOGIC (X one) (1) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) Interactive with limited (Ada) tasking (Sum equals 100%) (7) Interactive with limited (Ada) tasking (Sum equals 100%) (7) Imme-constrained (5) Other (Specify) (6) Concurrent tasking, rendezvous in milliseconds (6) Concurrent tasking, rendezvous in nanoseconds				(2) Additional Comments			1`′				
a. SOFTWARE INTERFACE COMPLEXITY (1) With how many CSCls or CSCs does this CSCl or CSC interface? (2) Additional Comments (2) Additional Comments (3) Average degree of CSCl or CSC integration no complex interfaces complexity (4) Several CSCl or CSC integration process anticipated (5) Complex, time-intensive CSCl or CSC integration process anticipated (6) Additional Comments c. DIFFICULTY OF PROCESSING LOGIC (X one) (1) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments 6) Additional Comments 7) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in nanoseconds (5) Onter (Specify) (5) Concurrent tasking, rendezvous in nanoseconds	(2) Additional Comments			(2) Additional Comments							
a. SOFTWARE INTERFACE COMPLEXITY (1) With how many CSCls or CSCs does this CSCl or CSC interface? (2) Additional Comments (2) Additional Comments (3) Average degree of CSCl or CSC integration no complex interfaces complexity (4) Several CSCl or CSC integration process anticipated (5) Complex, time-intensive CSCl or CSC integration process anticipated (6) Additional Comments c. DIFFICULTY OF PROCESSING LOGIC (X one) (1) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments 6) Additional Comments 7) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in nanoseconds (5) Onter (Specify) (5) Concurrent tasking, rendezvous in nanoseconds											
CSCI OR CSC LEVEL (X one) this CSCI or CSC interface? (2) Additional Comments (3) Average degree of CSCI or CSC integration and interface complexity (4) Several CSCI or CSC integration process anticipated (5) Complex, time-intensive CSCI or CSC integration process anticipated (6) Additional Comments (a) MATHEMATICAL COMPLEXITY (X one) (b) Additional Comments (c) DIFFICULTY OF PROCESSING LOGIC (X one) (d) Additional Comments (d) MATHEMATICAL COMPLEXITY (X one) (e) Majority of simple algorithms and simple calculations (e) Difficult, highly nested logic, real-time processing (f) Routine nesting, minimal interface with operating system, standard I/O (g) Adjority of simple algorithms and calculations of average complexity (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) Majority of simple algorithms and calculations of average complexity (d) Some difficult or complex calculations (e) Many difficult algorithms and complex calculations (f) Many difficult algorithms and complex calculations (g) Many difficult algorithms and calculations (g) Majority of simple algorithms and calculations (g) Majority		l									
this CSCI or CSC interface? (1) No internal integration (2) Very little integration, no complex interfaces (3) Average degree of CSCI or CSC integration and interface complexity (4) Several CSCI or CSC integration process anticipated (5) Complex, time-intensive CSCI or CSC integration process anticipated (6) Additional Comments c. DIFFICULTY OF PROCESSING LOGIC (X one) (1) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (8) Interrupt drive, tasking in milliseconds (9) Interrupt drive, tasking in milliseconds (1) Real-time (4) On-line (4) Concurrent tasking, rendezvous in milliseconds (5) Concurrent tasking, rendezvous in nanoseconds (6) Onterreprized					TY OI	FINTEGRATING	AND TES	TING C	OMPONENTS TO THE		
C) Not internal integration, no complex interfaces (2) Additional Comments (3) Average degree of CSCI or CSC integration and interface complexity (4) Several CSCI or CSC integration and interface complexity (4) Several CSCI or CSC integration process anticipated (6) Additional Comments (6) Additional Comments (6) Additional Comments (7) Simple algorithms and simple calculations (7) Simple algorithms and simple calculations (7) Simple algorithms and calculations (8) Agorithms and calculations (9) Majority of simple algorithms and calculations (9) Majorithms and calculations (9) M	• •			, ,							
(3) Average degree of CSCI or CSC integration and interface complexity (4) Several CSCI or CSC interfaces, some complex (5) Complex, time-intensive CSCI or CSC integration process anticipated (6) Additional Comments c. DIFFICULTY OF PROCESSING LOGIC (X one) (1) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) Majorithms and calculations of average complexity (4) Some difficult or complex calculations (5) Many difficult algorithms and complex calculations (6) Additional Comments (8) Additional Comments (9) Additional Comments (1) No tasking, essentially batch response (2) Interactive with limited (Ada) tasking (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in milliseconds (5) Concurrent tasking, rendezvous in nanoseconds (3) Non-time-critical	this CSCI of CSC interface?	(1) No inter	nal inte	gration							
(4) Several CSCI or CSC interfaces, some complex (5) Complex, time-intensive CSCI or CSC integration process anticipated (6) Additional Comments c. DIFFICULTY OF PROCESSING LOGIC (X one) (1) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (8) Interrupt drive, tasking in milliseconds (9) Time-constrained (1) Real-time (1) Other (Specify) (3) Non-time-critical		(2) Very littl	le integr	ation, no com	plex ir	terfaces					
c. DIFFICULTY OF PROCESSING LOGIC (X one) (1) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (6) Additional Comments (7) Many difficult algorithms and complex calculations (5) Many difficult algorithms and complex calculations (6) Additional Comments (8) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (9) Interactive with limited (Ada) tasking (1) Real-time (4) Concurrent tasking, rendezvous in milliseconds (1) Real-time (2) Time-constrained (3) Non-time-critical	(2) Additional Comments	(3) Average degree of CSCI or CSC integration and interface complexity									
c. DIFFICULTY OF PROCESSING LOGIC (X one) (1) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (6) Additional Comments (7) Majorithms and calculations of average complexity (4) Some difficult or complex calculations (5) Many difficult algorithms and complex calculations (6) Additional Comments (8) Additional Comments (9) Majorithms and calculations of average complexity (4) Some difficult or complex calculations (6) Additional Comments (9) Majorithms and calculations of average complexity (4) Some difficult or complex calculations (6) Additional Comments (9) Majorithms and calculations of average complexity (4) Some difficult or complex calculations (5) Many difficult algorithms and complex calculations (6) Additional Comments (9) Majorithms and calculations of average complexity (4) Some difficult or complex calculations (5) Many difficult algorithms and complex calculations (6) Additional Comments (9) Majorithms and calculations of average complexity (4) Some difficult or complex calculations (5) Many difficult algorithms and calculations (6) Additional Comments (9) Majorithms and calculations of average complexity (4) Some difficult or complex calculations (5) Many difficult algorithms and calculations (6) Additional Comments (8) Additional Comments (9) Majorithms and calculations (2) Majorithms and calculations		(4) Several CSCI or CSC interfaces, some complex									
c. DIFFICULTY OF PROCESSING LOGIC (X one) (1) Simple logic, straightforward I/O (2) Difficult, highly nested logic, real-time processing (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (8) Interrupt drive, tasking in milliseconds (9) Interactive with limited (Ada) tasking (1) Real-time (1) Simple algorithms and simple calculations (2) Majority of simple algorithms and calculations of average complexity (3) Algorithms and calculations of average complexity (4) Some difficult or complex calculations (5) Many difficult algorithms and complex calculations (6) Additional Comments (7) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (8) Interrupt drive, tasking in milliseconds (9) Interactive with limited (Ada) tasking (1) Real-time (4) On-line (5) Other (Specify) (5) Concurrent tasking, rendezvous in nanoseconds		(5) Complex, time-intensive CSCI or CSC integration process anticipated									
□ (1) Simple logic, straightforward I/O □ (1) Simple algorithms and simple calculations □ (2) Difficult, highly nested logic, real-time processing □ (2) Majority of simple algorithms and calculations □ (3) Routine nesting, minimal interface with operating system, standard I/O □ (3) Algorithms and calculations of average complexity □ (4) Complex dynamic resource allocation, multiple exception handles, recursion □ (4) Some difficult or complex calculations (5) Many difficult algorithms and complex calculations □ (5) Many difficult algorithms and complex calculations (5) Additional Comments □ (5) Additional Comments e. DEGREE OF REAL-TIME (X one) □ (6) Additional Comments □ (1) No tasking, essentially batch response □ (A) COCH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) □ (2) Interactive with limited (Ada) tasking □ (1) Real-time □ (2) Time-constrained □ (3) Other (Specify) □ (4) Concurrent tasking, rendezvous in manoseconds □ (3) Non-time-critical □ (5) Other (Specify)		(6) Additional Comments									
□ (1) Simple logic, straightforward I/O □ (1) Simple algorithms and simple calculations □ (2) Difficult, highly nested logic, real-time processing □ (2) Majority of simple algorithms and calculations □ (3) Routine nesting, minimal interface with operating system, standard I/O □ (3) Algorithms and calculations of average complexity □ (4) Complex dynamic resource allocation, multiple exception handles, recursion □ (4) Some difficult or complex calculations (5) Many difficult algorithms and complex calculations □ (5) Many difficult algorithms and complex calculations (5) Additional Comments □ (5) Additional Comments e. DEGREE OF REAL-TIME (X one) □ (6) Additional Comments □ (1) No tasking, essentially batch response □ (A) COCH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) □ (2) Interactive with limited (Ada) tasking □ (1) Real-time □ (2) Time-constrained □ (3) Other (Specify) □ (4) Concurrent tasking, rendezvous in manoseconds □ (3) Non-time-critical □ (5) Other (Specify)											
□ (1) Simple logic, straightforward I/O □ (1) Simple algorithms and simple calculations □ (2) Difficult, highly nested logic, real-time processing □ (2) Majority of simple algorithms and calculations □ (3) Routine nesting, minimal interface with operating system, standard I/O □ (3) Algorithms and calculations of average complexity □ (4) Complex dynamic resource allocation, multiple exception handles, recursion □ (4) Some difficult or complex calculations (5) Many difficult algorithms and complex calculations □ (5) Many difficult algorithms and complex calculations (5) Additional Comments □ (5) Additional Comments e. DEGREE OF REAL-TIME (X one) □ (6) Additional Comments □ (1) No tasking, essentially batch response □ (A) COCH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) □ (2) Interactive with limited (Ada) tasking □ (1) Real-time □ (2) Time-constrained □ (3) Other (Specify) □ (4) Concurrent tasking, rendezvous in manoseconds □ (3) Non-time-critical □ (5) Other (Specify)	c. DIFFICULTY OF PROCESSING LOGIC (X	one)			d. N	IATHEMATICAL	COMPLEX	(ITY (X	one)		
(2) Difficult, highly nested logic, real-time processing (2) Majority of simple algorithms and calculations (3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Many difficult or complex calculations (5) Many difficult algorithms and complex calculations (5) Many difficult algorithms and complex calculations (6) Additional Comments (6) Additional Comments (7) Many difficult algorithms and complex calculations (8) Additional Comments	I										
(3) Routine nesting, minimal interface with operating system, standard I/O (4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (7) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (8) Interrupt drive, tasking in milliseconds (9) Interrupt drive, tasking, rendezvous in manoseconds (1) Real-time (2) Time-constrained (3) Algorithms and calculations of average complexity (4) Some difficult or complex calculations (5) Many difficult algorithms and complex calculations (6) Additional Comments (6) Additional Comments (7) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (8) Sum equals 100% (1) Real-time (2) Time-constrained (3) Non-time-critical											
(4) Complex dynamic resource allocation, multiple exception handles, recursion (5) Additional Comments (6) Additional Comments (6) Additional Comments (7) Many difficult algorithms and complex calculations (8) Additional Comments (9) Many difficult algorithms and complex calculations (1) No tasking, essentially batch response (1) No tasking, essentially batch response (2) Interactive with limited (Ada) tasking (3) Interrupt drive, tasking in milliseconds (4) Some difficult or complex calculations (5) Many difficult algorithms and complex calculations (6) Additional Comments (7) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (1) Real-time (2) Time-constrained (3) On-line (5) Other (Specify) (5) Concurrent tasking, rendezvous in nanoseconds											
recursion (5) Additional Comments (6) Additional Comments (6) Additional Comments (7) Additional Comments (8) Additional Comments (9) Additional Comments (1) No tasking, essentially batch response (2) Interactive with limited (Ada) tasking (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in milliseconds (5) Many difficult algorithms and complex calculations (6) Additional Comments (7) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (1) Real-time (2) Time-constrained (5) Other (Specify) (5) Concurrent tasking, rendezvous in nanoseconds											
(5) Additional Comments (6) Additional Comments (7) ERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in milliseconds (5) Concurrent tasking, rendezvous in nanoseconds (6) Additional Comments f. PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (1) Real-time (2) Time-constrained (3) Non-time-critical											
e. DEGREE OF REAL-TIME (X one) (1) No tasking, essentially batch response (2) Interactive with limited (Ada) tasking (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in milliseconds (5) Concurrent tasking, rendezvous in nanoseconds (5) Concurrent tasking, rendezvous in nanoseconds (6) PERCENTAGE OF TOTAL SOURCE CODE ALLOCATED TO EACH OPERATIONAL TIMING REQUIREMENT (Sum equals 100%) (1) Real-time (2) Time-constrained (3) Non-time-critical	(5) Additional Comments										
(1) No tasking, essentially batch response TO EACH OPERATIONAL TIMING REQUIREMENT (2) Interactive with limited (Ada) tasking (Sum equals 100%) (3) Interrupt drive, tasking in milliseconds (1) Real-time (4) On-line (4) Concurrent tasking, rendezvous in milliseconds (2) Time-constrained (5) Other (Specify) (5) Concurrent tasking, rendezvous in nanoseconds (3) Non-time-critical	(v) Additional Comments										
(1) No tasking, essentially batch response TO EACH OPERATIONAL TIMING REQUIREMENT (2) Interactive with limited (Ada) tasking (Sum equals 100%) (3) Interrupt drive, tasking in milliseconds (1) Real-time (4) On-line (4) Concurrent tasking, rendezvous in milliseconds (2) Time-constrained (5) Other (Specify) (5) Concurrent tasking, rendezvous in nanoseconds (3) Non-time-critical	o DECREE OF REAL TIME (Y and)				1			0			
(2) Interactive with limited (Ada) tasking (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in milliseconds (5) Concurrent tasking, rendezvous in nanoseconds (2) Time-constrained (3) Non-time-critical											
(2) Interrupt drive, tasking in milliseconds (3) Interrupt drive, tasking in milliseconds (4) Concurrent tasking, rendezvous in milliseconds (5) Concurrent tasking, rendezvous in nanoseconds (5) Concurrent tasking, rendezvous in nanoseconds (6) Concurrent tasking, rendezvous in nanoseconds (7) Real-time (8) On-line (9) On-line (9) Other (Specify)								WIING F	REQUIRENIENI		
(4) Concurrent tasking, rendezvous in milliseconds (2) Time-constrained (5) Other (Specify) (5) Concurrent tasking, rendezvous in nanoseconds (3) Non-time-critical					(3		<i>''</i>		1,000		
(5) Concurrent tasking, rendezvous in nanoseconds (3) Non-time-critical						(1) Real-time		<u> </u>	(4) On-line		
	(4) Concurrent tasking, rendezvous in milli					(0)			(=) 0.1 (0 ::::		
(6) Additional Comments (6) Additional Comments						→ ` ′			(5) Other (Specify)		
						(3) Non-time-	critical		(5) Other (Specify)		

g. DISPLAY REQUIREMENT (X all that appl	h. SOFTWARE TESTABILITY (X one)							
(1) Simple I/O (4) Graphics oriented			(1) Very difficult (3) Time insensitive					
(2) User-friendly, menu driven (5) Not applicable			(2) Difficult		(4) Easy			
(3) Pressure-sensitive devices (touch sca	reen, joystick)		(5) Additional Com	ments				
(5) Additional Comments								
11. SOFTWARE RELIABILITY	I			1				
a. EFFECT OF SOFTWARE FAILURE	b. BACKUP CONSIDERATION				Y CONSIDERATIONS			
(X as applicable)	(1) Data protection beyond	d regular	r backup	(X one)				
(1) Inconvenience	required		(1) Alternative methods need to be developed in case of software					
(2) Easily recoverable loss (3) Moderate loss (Recoverable)	(2) No special backup requ							
(4) Major loss (High financial loss)	(3) Alternative methods ne		e aevelopea	_	ecial recovery requirements			
(5) Additional Comments	(4) Additional Comments	u. 0	(3) Additional Comments					
(o) ridditional comments	(4) Additional Commonts		(b) Additional Comments					
11. SOFTWARE RELIABILITY				l .				
a. DATABASE SIZE	b. PHYSICAL DATA FILES	c. DA	TABASE COMPLEX	ITY (X one)				
(1) Kilobytes	(1) Number of Files) Simple data, few fi	les, low capacit	ıy			
		(2	2) Simple, numerous	variables				
(2) Additional Comments	(2) Additional Comments	□ (3	B) Multiple files, fields	data interaction	ins			
		[(4) Complex file struct	ure				
		<u> </u>	5) Highly complex					
		(6) Ac	dditional Comments					
13. SOFTWARE REUSE (If applicable)								
a. LOGICAL COMPLEXITY OF CODE REUS	SED FROM OTHER	b. STRUCTURAL COMPLEXITY OF CODE REUSED FROM OTHER						
PROGRAMS (X one)		PROGRAMS (X one)						
(1) Simple algorithms and simple calcula		(1) Nonprocedural (Generated, query, spreadsheets, etc.)						
(2) Majority of simple algorithms and calc		(2) Well structured with usable modules						
(3) Algorithms and calculations of average		(3) Fair structure, some complex paths and modules (4) Poor structure, many complex paths and modules						
(4) Some difficult or complex calculations			dditional Comments	ny complex pai	ns and modules			
(5) Many difficult algorithms and complex calculations (6) Additional Comments		(3) AC	dulional Comments					
(o) Additional Comments								
c. COMPLEXITY OF DATABASE REUSED I	FROM OTHER PROGRAMS	d. IF F	PLANNING TO REU	SE THIS CSCI	IN ANOTHER PROGRAM,			
(If applicable)		SELECT INTENDED USE (X one)						
(1) Simple data, few variables, little complexity			(1) None					
(2) Several data elements, simple data relationships			(2) Reuse within element					
(3) Multiple files, switches, and data inter	ractions	(3) Reuse across element						
(4) Complex data elements, complex dat	ta interactions	(4) Reuse in another DoD program application						
(5) Very complex data elements and interactions			(5) Additional Comments					
(6) Additional Comments								
44 00573405 1441075340105								
14. SOFTWARE MAINTENANCE	anas will be required		(4) Indicate annual	ahanga rata fa	r ooftwara			
(1) Indicate number of years maintena	•	(E) A c	(4) Indicate annual	change rate to	r sonware			
(2) Indicate number of separate maintenance sites			(5) Additional Comments					
(3) Indicate estimated maintenance/software growth over life 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		mat cour	a ancoi inc cost and	yor size or the v	Joon, Joo Jonward			
	,							