

N86 - 30366

**One Approach for Evaluating
The Distributed Computing Design System (DCDS) ***

- Extended Abstract -

Submitted to:

**Tenth Annual Software Engineering Workshop
Scheduled for December 4, 1985**

**Missile Operations and Data Systems Directorate
Flight Dynamics Division
Goddard Space Flight Center**

***This material was presented
by L. Baker of TRW**

John T. Ellis

**TRW Defense Systems Group
213 Wynn Drive
Huntsville, Alabama 35805
(205)830-3326**

SEPTEMBER 25, 1985

**J. Ellis
TRW
1 of 24**

One Approach For Evaluating the Distributed Computing Design System

DCDS provides an integrated environment to support the life cycle of developing real-time distributed computing systems. The primary focus of DCDS is to significantly increase system reliability and software development productivity, and to minimize schedule and cost risk. DCDS consists of integrated methodologies, languages, and tools to support the life cycle of developing distributed software and systems. Smooth and well-defined transitions from phase to phase, language to language, and tool to tool provide a unique and unified environment. An approach to evaluating DCDS highlights its benefits.

1. DCDS OVERVIEW

Distributed solutions to complex systems require sophisticated tools and techniques for the specification and development of distributed software. In response to this need, TRW has developed the Distributed Computing Design System (DCDS) to provide an integrated environment for the specification and life-cycle development of software and systems, with an emphasis on the development of real-time distributed software. The primary focus of DCDS is to significantly increase system reliability and software development productivity, through the use of disciplined techniques and automated tools. To minimize schedule and cost risk, DCDS offers management visibility into the development process. The development of DCDS is sponsored by the Ballistic Missile Defense Advanced Technology Center (BMDATC).

As illustrated in Figure 1, DCDS consists of integrated methodologies, integrated languages, and an integrated tool set. Following the five methodologies, the user can produce specifications for system requirements, software requirements, distributed architectural designs, detailed module designs, and tests. The five languages support the specific concepts for each of the methodologies, and provide the medium for expressing the requirements, designs, and tests. All five languages use the same constructs and syntax. DCDS formal languages, as opposed to natural languages such as English, can be used without ambiguity - all components of the language are explicitly defined.

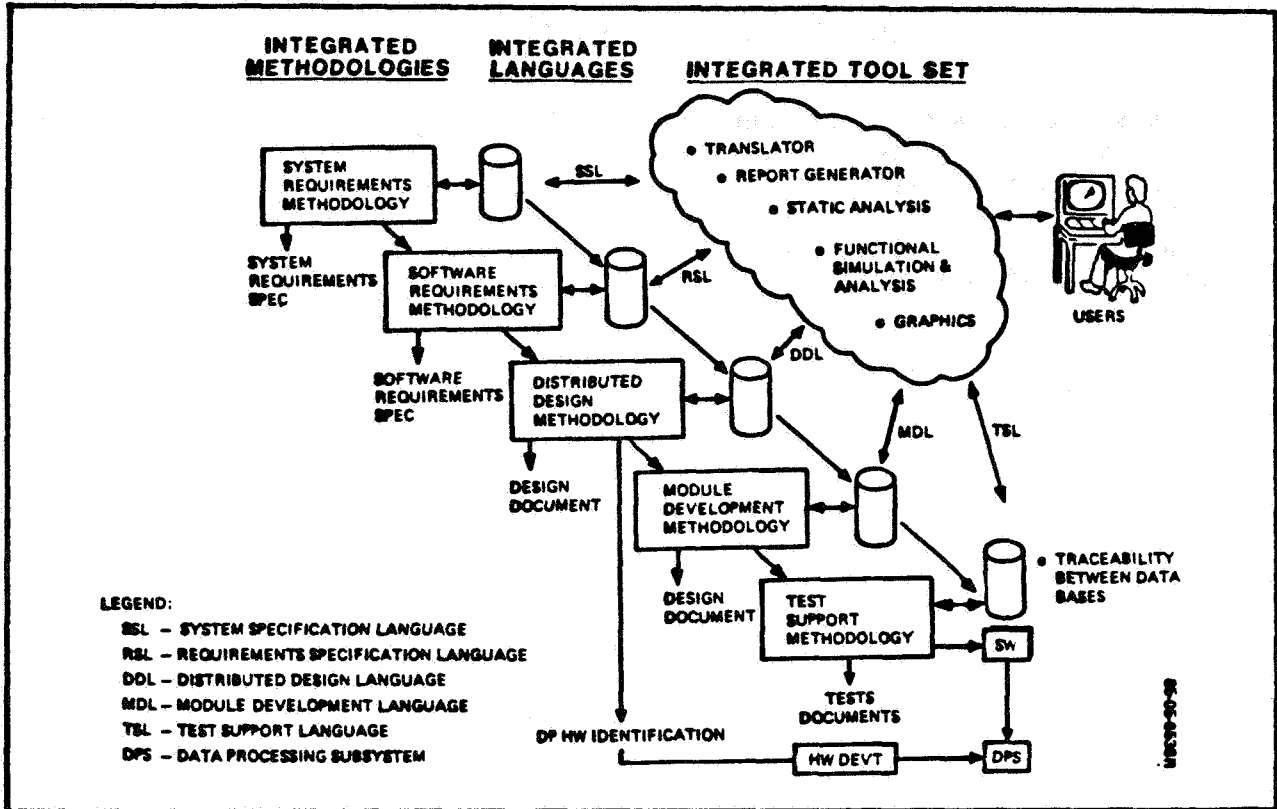


Figure 1. The DCDS Unified Environment

As shown in Figure 1, the user has access to a variety of tools to incrementally define the specification contents, and to check them for completeness and consistency. For each methodology, the tools maintain a data base to store the specification contents. The data base maintains the specification information in a support suitable for automated and thorough analysis. DCDS tools can also support simulation and various types of analyses.

Data extraction tools are used to generate readable listings according to user-defined formats. The listings can be used as working-level documentation, briefing charts, or incorporated into formal specifications. The data base from one methodology is used as a source to initialize the data bases in downstream methodologies, permitting automated traceability between specifications.

THE FIVE DCDS METHODOLOGIES

1. **System Requirements Engineering Methodology (SYSREM)** for defining and specifying system requirements, with an emphasis on the data processing subsystem.
2. **Software Requirements Engineering Methodology (SREM)** for defining system software requirements, with an emphasis on stimulus-response behavior.
3. **Distributed Design Methodology (DDM)** for developing a top-level architectural design for the system software, including distributed design, process design, and task design.
4. **Module Development Methodology (MDM)** for investigating and selecting algorithms, defining detailed design, and producing units of tested code.
5. **Test Support Methodology (TSM)** for defining test plans and procedures against requirements, producing an integrated tested system, and recording test results.

THE FIVE DCDS LANGUAGES

1. **System Specification Language (SSL)** for specifying structured sequences of functions to be performed by the system, inputs/outputs between functions, performance indices for functions, and allocations of functions to subsystems.
2. **Requirements Statement Language (RSL)** for describing a stimulus-response structure of inputs, outputs, processing, and performance of a DP subsystem in a form which assures unambiguous specifications of explicit, testable software requirements.
3. **Distributed Design Language (DDL)** for describing the distributed hardware architectures of processing nodes and interconnections, the software architecture, the allocation of processing and data to nodes, and the communication topology.
4. **Module Development Language (MDL)** for recording detailed designs and algorithms considered and selected for the design.
5. **Test Support Language (TSL)** for recording tests, their relationship to the requirements, test procedures, and test results.

Figure 2. DCDS Methodologies and Languages

DCDS is used to produce units of tested software, and to identify the data processing hardware. Tools are available to aid in the software process construction activities. The final output (Figure 1) from DCDS is the integrated and tested Data Processing Subsystem.

The DCDS methodologies and languages are defined in Figure 2. Within each methodology, individual steps are provided and are explicit and observable. Activities are defined and must be completed prior to each of the major reviews during the development life cycle. Well-defined interfaces between the life-cycle phases allow a unified approach for using DCDS. DCDS also provides measurable intermediate milestones for management visibility between the major review points.

DCDS provides a unique and proven capability. First, DCDS is the only integrated environment which addresses the entire life cycle of distributed software development. The techniques are independent of the implementation language, and can be applied effectively to development activities or used as a verification and validation tool. Second, DCDS concepts are based on proven technology - the early results, oriented for software requirements, have been validated, improved, and now extended to support the complete system development life cycle. DCDS is the result of 12 years of research and development, as discussed in IEEE COMPUTER magazine.*

2. DCDS EVALUATION

To gain a better perspective on DCDS and its characteristics, DCDS was compared against three other commercially available products. These three products provide methodologies and/or tools for developing specifications and software. To allow an objective and multi-factored comparative evaluation of the different methodologies and tools, TRW prepared a list of evaluation criteria partitioned into three classes: (1) factors lending credibility to the product, (2) costs of acquiring and using the product, and (3) benefits of the product.

*M. Alford, "SREM At the Age of Eight", IEEE COMPUTER, April 1985, pp. 36-46.

The individual criteria from each of the three classes was assigned a value weight of "high", "medium", and "low". A score of "better", "acceptable", or "deficient" was used to evaluate each product against each evaluation criteria. An explanation of each evaluation criteria and the rationale for each individual score against each product is available.

The results of the evaluation are summarized in Figure 3. Since the evaluation was not performed by an independent organization, the other three products shall remain nameless. However, they do represent well-known products. All the products support an overall acceptable rating, and have been used successfully in major applications. DCDS received an overall higher rating within this evaluation process due to the following discriminating factors:

- Automated traceability across life-cycle phases
- Automated analysis tools
- Documentation support capabilities
- Relatively low cost to acquire and use the product

It is anticipated that the evaluation approach and criteria as outlined in this report could be used by an independent agency for a more in-depth analysis and evaluation of various methodologies and tools. The author wishes to acknowledge Mack Alford and Bob Loshbough of TRW for their extensive technical contribution to the author's summation of DCDS and its evaluation.

ORIGINAL PAGE IS
 OF POOR QUALITY

GENERAL CREDIBILITY FACTORS	PROPERTIES OF		VALUE WEIGHT	DCDS	Pro-	Pro-	Pro-
	TOOL	PRODUCT			duct X	duct Y	duct Z
<ul style="list-style-type: none"> ● MATURITY/VENDOR SUPPORT ● USER COMMUNITY ● BASIC CONCEPT/ASSUMPTION 	X		HIGH				
	X		HIGH				
	X	X	MED				
COST FACTORS							
<ul style="list-style-type: none"> ● TOOL AND SUPPORTIVE HW/SW ACQUISITION 	X		MED				
<ul style="list-style-type: none"> ● LEARNING/TRAINING TIME AND COST 	X		MED				
<ul style="list-style-type: none"> ● OPERATING COST - BUILD, ANALYSIS DATA BASE 	X	X	HIGH				
<ul style="list-style-type: none"> ● OPERATING COST - DOCUMENTATION/PRODUCT 		X	HIGH				
BENEFIT FACTORS							
<ul style="list-style-type: none"> ● EXISTING METHODOLOGY 	X		MED				
<ul style="list-style-type: none"> ● ALTERNATE COMPUTER HOSTS AVAILABLE 	X		HIGH				
<ul style="list-style-type: none"> ● INTERPHASE TRACEABILITY (RQMTS--- TEST) 		X	HIGH				
<ul style="list-style-type: none"> ● DATA BASE CONCEPT 	X	X	LOW				
<ul style="list-style-type: none"> ● AUTOMATED ANALYSIS 	X		HIGH				
<ul style="list-style-type: none"> ● AUTOMATED DOCUMENTATION 	X		HIGH				
<ul style="list-style-type: none"> ● EASE OF MODIFYING DOCUMENTATION 	X	X	MED				
<ul style="list-style-type: none"> ● AVAILABILITY 		X	MED				
<ul style="list-style-type: none"> ● RESPONSIVENESS 	X		MED				
<ul style="list-style-type: none"> ● ERROR CLASSES IDENTIFIED BY TOOLS 	X	X	HIGH				
<ul style="list-style-type: none"> ● MANAGEMENT VISIBILITY 		X	HIGH				
<ul style="list-style-type: none"> ● OPERATING COST IMPACT 		X	HIGH				
<ul style="list-style-type: none"> ● LABOR REQUIRED 	X		MED				
<ul style="list-style-type: none"> ● COMPUTER TIME REQUIRED 	X		LOW				

PATTERN: BETTER (B) ACCEPTABLE (A) DEFICIENT (D)

Figure 3. Evaluation Results

THE VIEWGRAPH MATERIALS
for the
J. ELLIS PRESENTATION FOLLOW

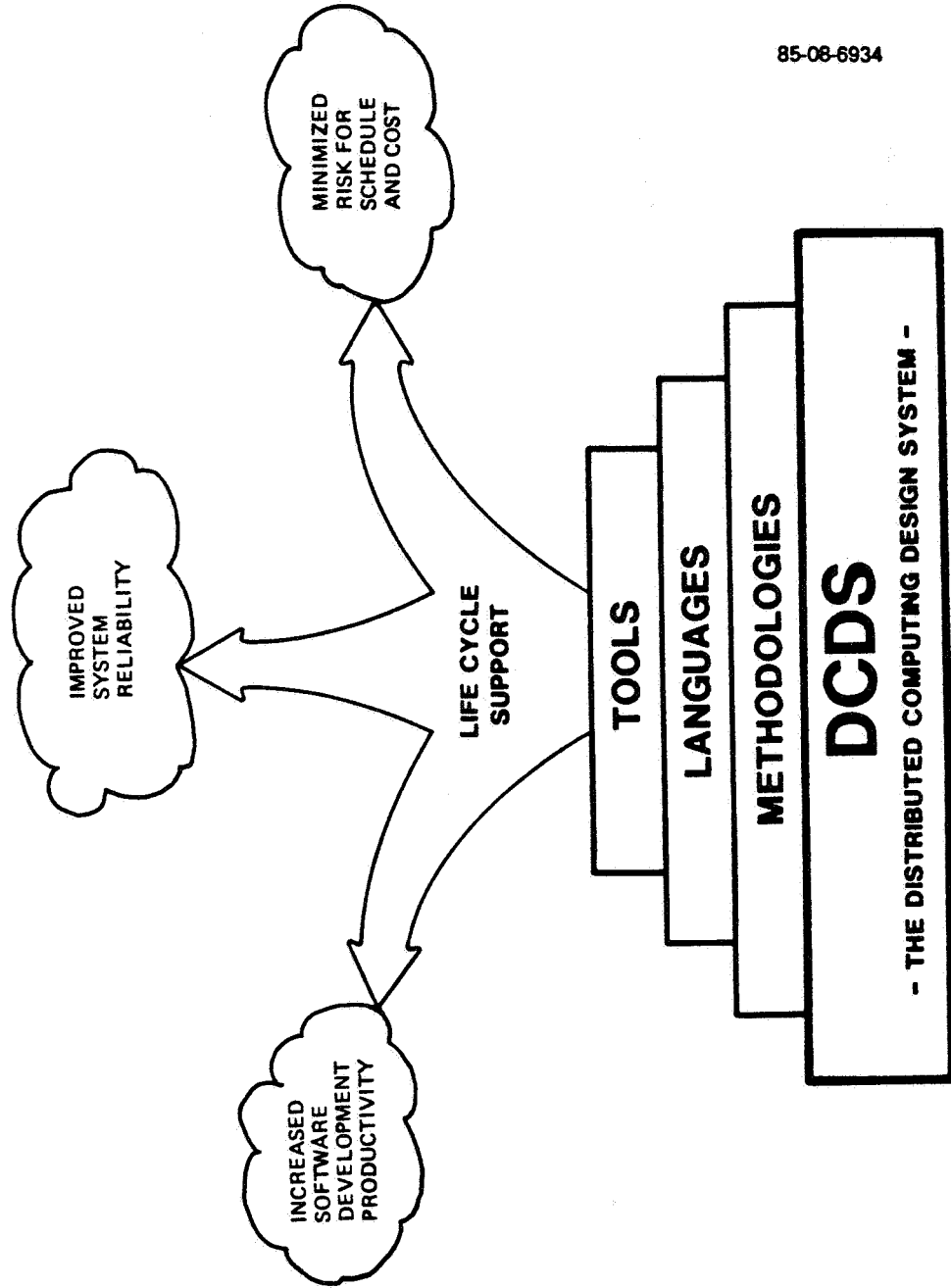


TRW85-169

**APPROACHES FOR EVALUATING
THE
DISTRIBUTED COMPUTING DESIGN SYSTEM
(DCDS)**

4 December 1985

THE DISTRIBUTED COMPUTING DESIGN SYSTEM



DCDS OBJECTIVES (1981)

TRW Defense
Systems Group
Huntsville Laboratory



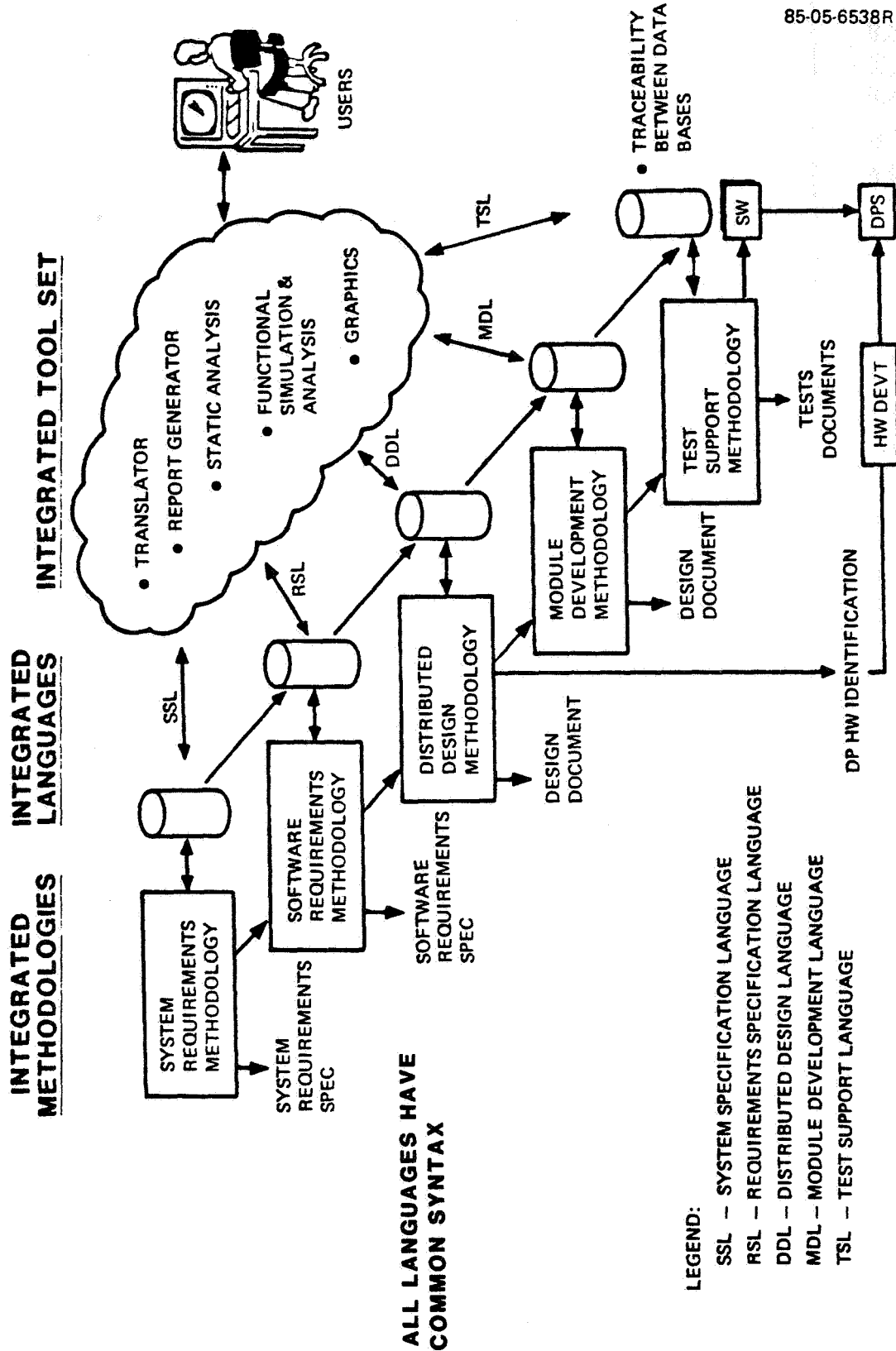
OBJECTIVES

- A SECOND GENERATION SOFTWARE DEVELOPMENT ENVIRONMENT WITH AN INTEGRATED METHODOLOGY
- BASED ON A REQUIREMENTS-DRIVEN METHODOLOGY
 - INTEGRATED MODEL ACROSS ALL DEVELOPMENT PHASES
 - INTEGRATED LANGUAGE FOR REQUIREMENTS, DESIGN, TEST, AND MANAGEMENT
 - AUTOMATED TOOLS EXPLOIT LANGUAGE
 - DESIGN ANALYSIS AND SYNTHESIS
 - EARLY AND MORE COMPREHENSIVE ERROR DETECTION
 - INTEGRATED METHODOLOGY FOR ALL PHASES
- PRODUCTIVITY INCREASE (200 PERCENT TO 400 PERCENT)
- QUALITY INCREASE (DEFECTS REDUCED TO 1 DEFECT PER 1000 LINES OF CODE)

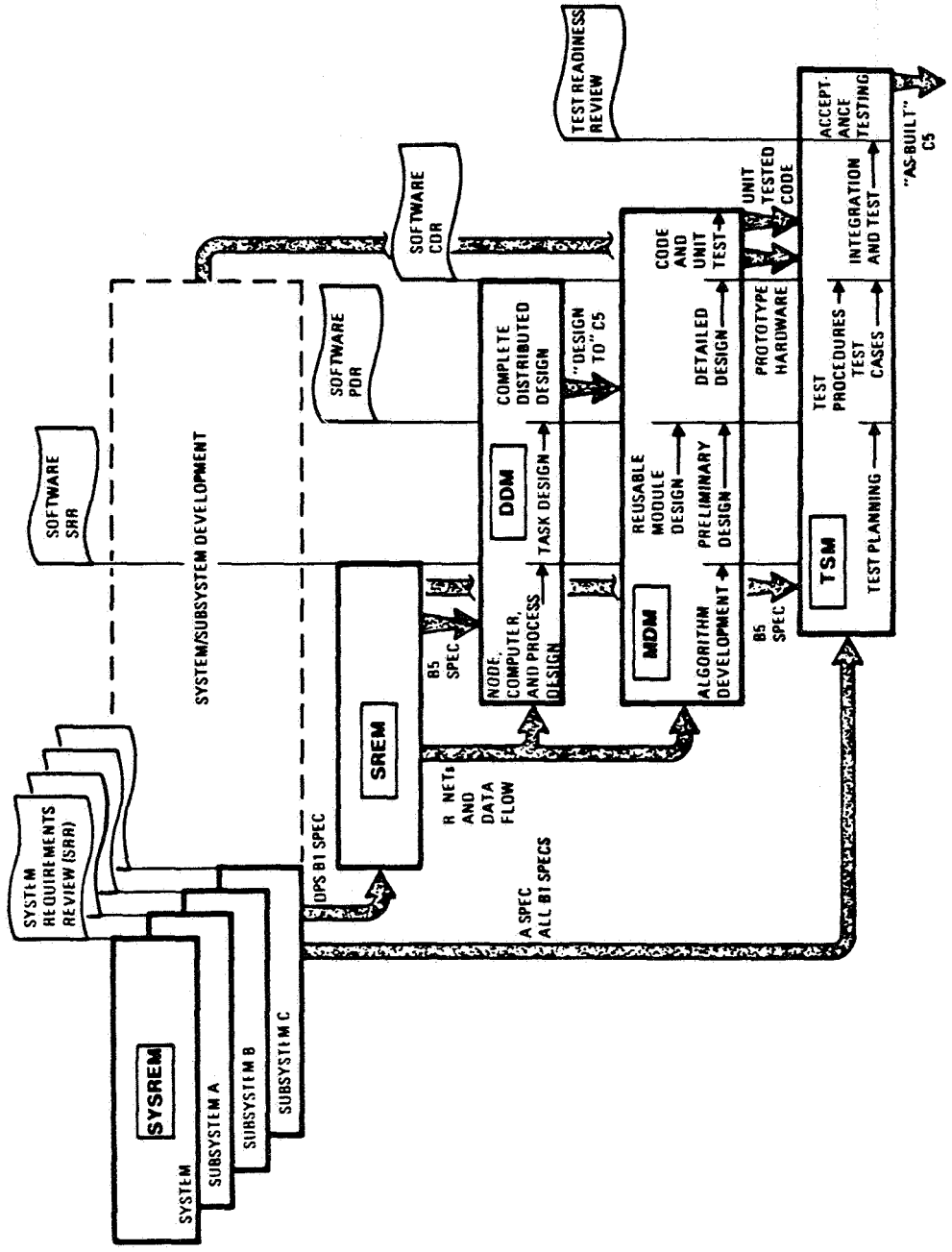
DCDS SUPPORTING GOALS



- INTEGRATION OF EFFORTS BETWEEN METHODOLOGIES
- BETTER UNDERSTANDING OF THE PROCESS AND PRODUCTS
 - SYSTEMATIC SEQUENCE OF STEPS AND DECISIONS
 - IMPROVED REPRESENTATION OF THE PRODUCTS IN DATA BASES
 - SEPARATION OF CONCERNS
 - ADDED DISCIPLINE
 - MANAGEMENT VISIBILITY
- EARLY REQUIREMENTS EMPHASIS
 - DISCIPLINED STRUCTURE APPROACH
 - PORTION OF EACH DATA BASE USED TO INITIATE DOWNSTREAM DATA BASES
 - ACHIEVES REAL PAYOFF IN REDUCTION OF DOWNSTREAM ERRORS
- AUTOMATED TOOL SUPPORT
 - MENU-DRIVEN SYSTEM
 - DATA BASE CONSISTENCY COMPLETENESS CHECKING
 - ELECTRONIC FORMS ENTRY
 - DATA FLOW ANALYZER
 - CLUSTERING ANALYZER
 - QUERY SYSTEM (ALSO SUPPORTS DOCUMENTATION)
 - INTERACTIVE GRAPHICS AND PLOTTING SUPPORTS STRUCTURES
 - AUTOMATED UNIT DEVELOPMENT FOLDER
 - PROCESS CONSTRUCTION SYSTEM
- STRONG TRACEABILITY
 - REQUIREMENTS DRIVEN
 - DIRECT TRACEABILITY ACHIEVABLE BETWEEN ELEMENTS IN EACH DATA BASE
 - EASES MODIFICATIONS



**THE INTEGRATED DCDS METHODOLOGIES SUPPORT
THE ENTIRE SOFTWARE DEVELOPMENT CYCLE**



ORIGINAL PAGE IS
OF POOR QUALITY

85-11-7574



HOW DO I EVALUATE

METHODOLOGY/ENVIRONMENT?

QUALITATIVE APPROACH



ORIGINAL PAGE IS
OF POOR QUALITY

85-12-7675

GENERAL CREDIBILITY FACTORS	PROPERTIES OF		VALUE WEIGHT	DCDS	SREM	X	Y	Z
	TOOL	PRODUCT						
<ul style="list-style-type: none"> MATURITY/VENDOR SUPPORT USER COMMUNITY BASIC CONCEPT/ASSUMPTION 	X		HIGH					
	X		HIGH					
	X	X	MED					
COST FACTORS								
<ul style="list-style-type: none"> TOOL AND SUPPORTIVE HW/SW 	X		MED					
<ul style="list-style-type: none"> ACQUISITION LEARNING/TRAINING TIME AND COST OPERATING COST - BUILD, ANALYSIS DATA BASE OPERATING COST - DOCUMENTATION/ PRODUCT 	X		MED					
	X		MED					
	X	X	HIGH					
	X	X	HIGH					
BENEFIT FACTORS								
<ul style="list-style-type: none"> EXISTING METHODOLOGY ALTERNATE COMPUTER HOSTS INTERPHASE TRACEABILITY (RQMTS--- TEST) DATA BASE CONCEPT AUTOMATED ANALYSIS AUTOMATED DOCUMENTATION EASE OF MODIFYING DOCUMENTATION AVAILABILITY RESPONSIVENESS ERROR CLASSES IDENTIFIED BY TOOLS MANAGEMENT VISIBILITY OPERATING COST IMPACT LABOR REQUIRED COMPUTER TIME REQUIRED 	X		MED					
	X		HIGH					
		X	HIGH					
		X	LOW					
	X		HIGH					
	X		HIGH					
	X		MED					
	X		MED					
	X		MED					
	X		HIGH					
	X	X	HIGH					
	X	X	HIGH					
	X		MED					
	X		MED					
	X		MED					
	X		HIGH					
	X	X	HIGH					
	X	X	HIGH					
	X		MED					
	X		LOW					

PATTERN: ACCEPTABLE (A) DEFICIENT (D)

BETTER (B)

QUANTITATIVE-MULTIPLE PROJECT TYPES



METHODOLOGY	SOFTWARE CATEGORY: SIMPLE																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	20	21	21	19	19	24	21	22	20	21	24	20	21	20	16	23	22	21
B	33	34	34	37	34	37	40	37	33	40	37	33	40	33	25	36	37	34
C	23	23	23	20	20	26	23	23	23	23	26	23	23	23	16	23	23	23
D	28	28	28	26	26	30	28	28	28	28	30	28	28	28	22	31	28	27
E	27	27	27	27	25	31	30	31	27	30	31	27	30	27	19	30	31	27
F*	40	39	39	40	40	40	41	39	40	41	40	40	41	40	29	42	39	38
G	22	19	19	20	20	24	23	24	22	23	24	22	23	22	13	26	24	22
H**	42	40	40	41	41	45	44	45	42	44	45	42	44	42	27	47	45	42
I	24	21	21	21	18	26	24	26	24	24	26	24	24	24	17	25	26	22
J	27	33	33	33	30	33	33	33	27	33	33	27	33	27	19	30	33	27
K	35	40	40	38	37	43	41	41	35	41	43	35	41	35	27	40	41	37
L	38	35	35	38	35	40	41	38	38	41	40	38	41	38	26	39	38	36

* SREM
** DCDS

TRW Defense
Systems Group
Huntsville Laboratory

QUANTITATIVE-MULTIPLE PROJECT TYPES (CONTINUED)



METHODOLOGY	SOFTWARE CATEGORY: MEDIUM																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	1	1	1	-1	0	2	-1	1	1	-1	2	1	-1	1	1	1	1	2
B	14	14	14	17	15	15	18	16	14	18	15	14	18	14	10	14	16	15
C	4	3	3	0	1	4	1	2	4	1	4	4	1	4	1	1	2	4
D	9	8	8	6	7	8	6	7	9	6	8	9	6	9	7	9	7	8
E	8	7	7	7	6	9	8	10	8	8	9	8	8	8	4	8	10	8
F*	21	19	19	20	21	18	19	18	21	19	18	21	19	21	14	20	18	19
G	3	-1	-1	0	1	2	1	3	3	1	2	3	1	3	-2	4	3	3
H**	23	20	20	21	22	23	22	24	23	22	23	23	22	23	12	25	24	23
I	5	1	1	1	1	4	2	5	5	2	4	5	2	5	2	3	5	3
J	8	13	13	13	11	11	11	12	8	11	11	8	11	8	4	8	12	8
K	16	20	20	18	18	21	19	20	16	19	21	16	19	16	12	18	20	18
L	19	15	15	18	16	18	19	17	19	19	18	19	19	19	11	17	17	17

* SREM
** DCDS

QUANTITATIVE-MULTIPLE PROJECT TYPES (CONTINUED)



85-12-7677

METHODOLOGY	SOFTWARE CATEGORY: REAL-TIME																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	-18	-19	-19	-21	-19	-20	-23	-20	-18	-23	-20	-18	-23	-18	-14	-21	-20	-17
B	-5	-6	-6	-3	-4	-7	-4	-5	-5	-4	-7	-5	-4	-5	-5	-8	-5	-4
C	-15	-17	-17	-20	-18	-18	-21	-19	-15	-21	-18	-15	-21	-15	-14	-21	-19	-15
D	-10	-12	-12	-14	-12	-14	-16	-14	-10	-16	-14	-10	-16	-10	-8	-13	-14	-11
E	-11	-13	-13	-13	-13	-13	-14	-11	-11	-14	-13	-11	-14	-11	-11	-14	-11	-11
F*	2	-1	-1	0	2	-4	-3	-3	2	-3	-4	2	-3	2	-1	-2	-3	0
G	-16	-21	-21	-20	-18	-20	-21	-18	-16	-21	-20	-16	-21	-16	-17	-18	-18	-16
H**	4	0	0	1	3	1	0	3	4	0	1	4	0	4	-3	3	3	4
I	-14	-19	-19	-19	-20	-18	-20	-16	-14	-20	-18	-14	-20	-14	-13	-19	-16	-16
J	-11	-7	-7	-7	-8	-11	-11	-9	-11	-11	-11	-11	-11	-11	-11	-14	-9	-11
K	-3	0	0	-2	-1	-1	-3	-1	-3	-3	-1	-3	-3	-3	-3	-4	-1	-1
L	0	-5	-5	-2	-3	-4	-3	-4	0	-3	-4	0	-3	0	-4	-5	-4	-2

* SREM
** DCDS

QUANTITATIVE-MULTIPLE PROJECT TYPES (CONTINUED)



85-12-7679

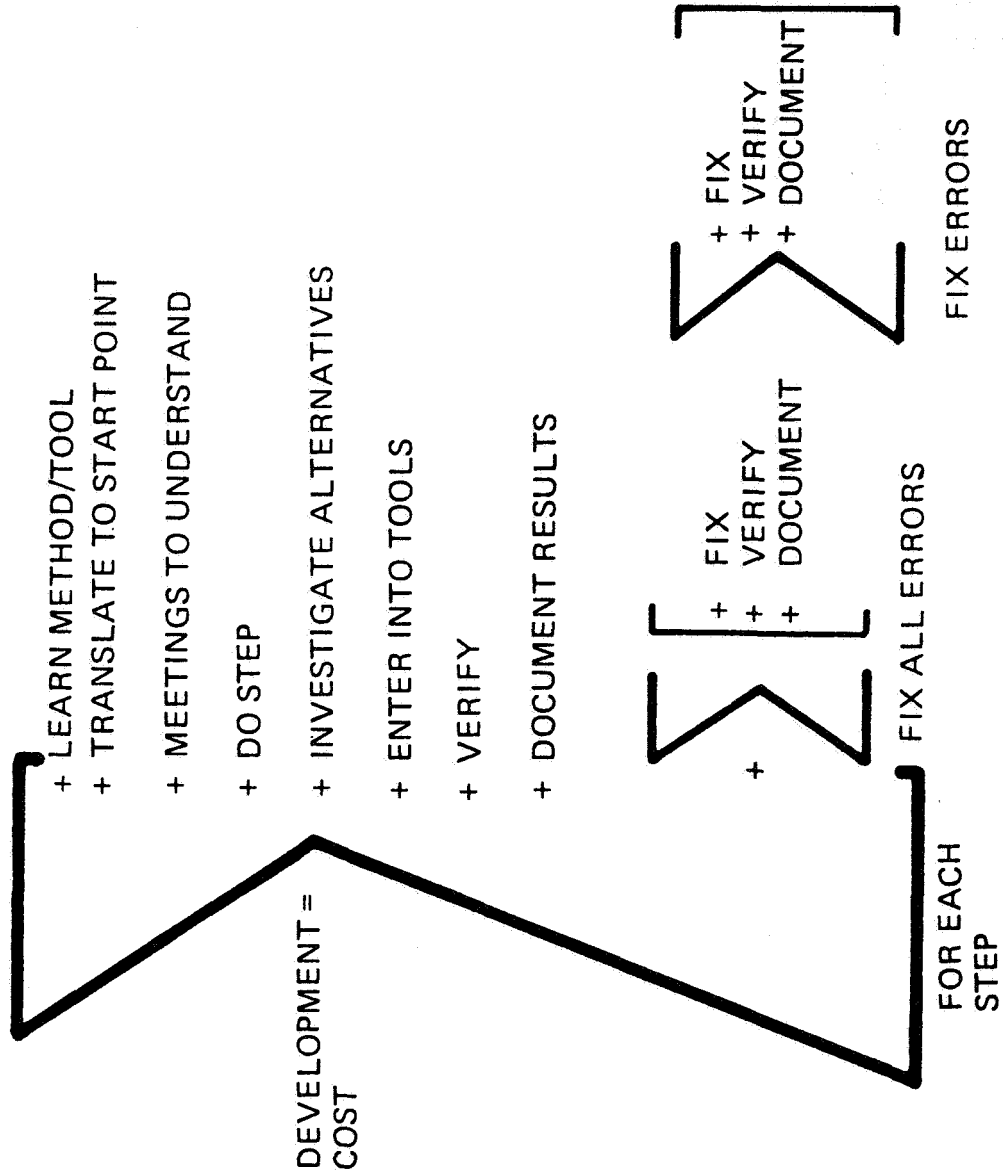
METHODOLOGY	SOFTWARE CATEGORY: MAN-RATED																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	-37	-30	-30	-41	-38	-42	-45	-41	-37	-45	-42	-37	-45	-37	-29	-43	-41	-36
B	-24	-26	-26	-23	-23	-29	-26	-26	-24	-26	-29	-24	-26	-24	-20	-30	-26	-23
C	-34	-37	-37	-40	-37	-40	-43	-40	-34	-43	-40	-34	-43	-34	-29	-43	-40	-34
D	-29	-32	-32	-34	-31	-36	-38	-35	-29	-38	-36	-29	-38	-29	-23	-35	-35	-30
E	-30	-33	-33	-33	-32	-35	-36	-32	-30	-36	-35	-30	-36	-30	-26	-36	-32	-30
F*	-17	-21	-21	-20	-17	-26	-25	-24	-17	-25	-26	-17	-25	-17	-16	-24	-24	-19
G	-35	-41	-41	-40	-37	-42	-43	-39	-35	-43	-42	-35	-43	-35	-32	-40	-39	-35
H**	-15	-20	-20	-19	-16	-21	-22	-18	-15	-22	-21	-15	-22	-15	-18	-19	-18	-15
I	-33	-39	-39	-39	-39	-40	-42	-37	-33	-42	-40	-33	-42	-33	-28	-41	-37	-35
J	-30	-27	-27	-27	-27	-33	-33	-30	-30	-33	-33	-30	-33	-30	-26	-36	-30	-30
K	-22	-20	-20	-22	-20	-23	-25	-22	-22	-25	-23	-22	-25	-22	-18	-26	-22	-20
L	-19	-25	-25	-22	-22	-26	-25	-25	-19	-25	-26	-19	-25	-19	-19	-27	-25	-21

* SREM
** DCDS

COST MODEL APPROACH FOR EVALUATING DCDS



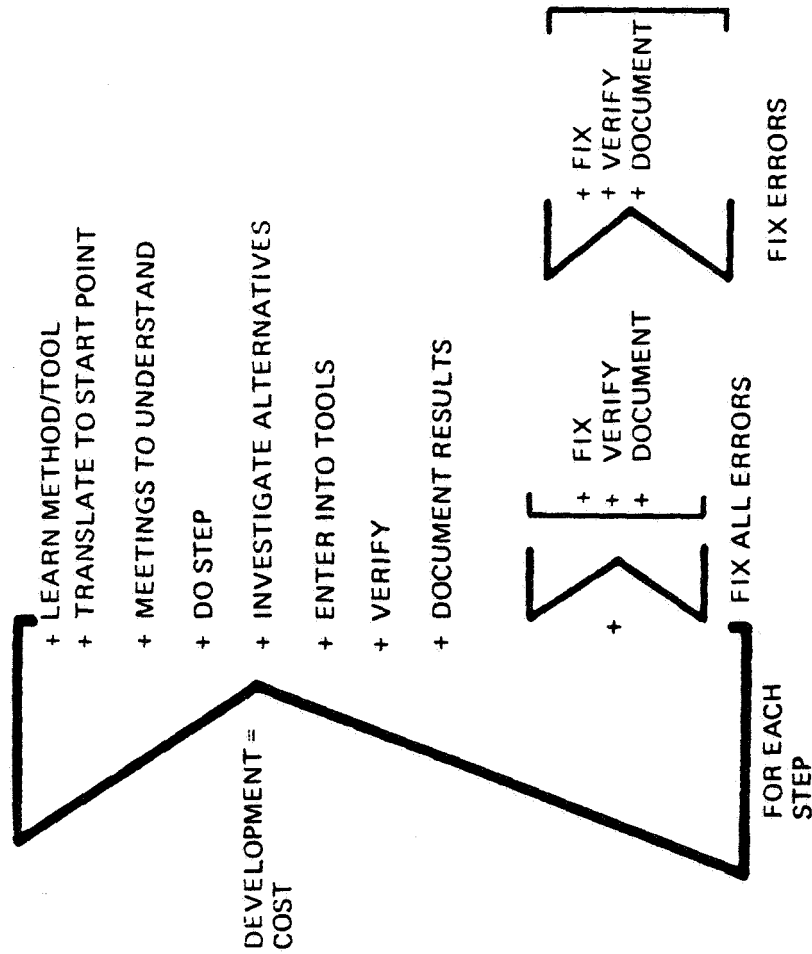
85-11-7589R



COST MODEL-SAMPLE CALCULATIONS



COST MODEL



EXAMPLE CALCULATIONS*

MANUAL METHOD	GOOD MANUAL VERIFICATION	METHODOLOGY AND AUTOMATED TOOLS (DCDS)
-	-	-
15	15	5
10	10	5
30	30	25 (15?)
10	10	10
0	0	5
5	25	5
30	30	5
<u>100%</u>	<u>120%</u>	<u>60% (50%)</u>

85-11-7589R

200%	100%	20%
300%	220%	80%

* Estimates

FULL COST VERSUS SIZE



SIZE	SMALL	MEDIUM	LARGE
COST	\$0.5M	\$5M	\$40M
NOMINAL REQUIREMENTS (3.6%)	\$18K	\$180K	\$1.8M
<u>REAL REQUIREMENTS</u>			
● MANUAL (300%)	\$54K	\$540K	\$5.4M
● GOOD MANUAL VERIFICATION (220%)	\$40K	\$400K	\$4.0M
● DCDS LABOR (80%)	\$14K	\$140K	\$1.4M
+			
(LEARN METHODOLOGY)	\$20K	\$50K	\$700K
+			
(TOOL COST (μVAX))	\$20K	\$25K	\$50K

CONCLUSIONS



- QUALITATIVE EVALUATION APPROACHES DON'T ADDRESS THE "REAL" BOTTOM LINE COST
- (DITTO QUANTITATIVE APPROACHES BASED ON THEM)
- DATA IS NOT BEING KEPT ON THE RIGHT COSTS TO SUPPORT EVALUATION
- SOME TRENDS ARE CLEAR
 - INTEGRATED METHODS/TOOLS REDUCE INTERPHASE COSTS/ ERRORS
 - VERIFICATION REDUCES ERRORS
 - AUTOMATED VERIFICATION REDUCES COSTS AND ERRORS
 - AUTOMATED DOCUMENTATION REDUCES COST
 - LEARNING AND TOOL COSTS ARE NON-LINEAR