N91-10612

2??

30 November 1988

Software Sizing, Cost Estimation and Scheduling

William G. Cheadle
Martin Marietta Astronautics Group
Mail Number L0330
Post Office Box 179
Denver, Colorado 80201

MI 411300

INTRODUCTION

The Technology Implementation and Support Section at Martin Marietta Astronautics Group Denver is tasked with software development analysis, data collection, software productivity improvement and developing and applying various computerized software tools and models. The computerized tools are parametric models that reflect actuals taken from our large data base of completed software development projects. Martin Marietta's data base consists of over 300 completed projects and hundreds of cost estimating relationships (CERs) that are used in sizing, costing, scheduling and productivity improvement equations, studies, models and computerized tools.

BACKGROUND

Martin Marietta resolved in 1975 to establish a study effort to investigate the software development process and the understanding of how to plan, schedule, size, and estimate software. The outcome of this analysis was that management decided to develop a company-peculiar parametric software estimating cost, schedule, and manloading model. This parametric model was generated by using actual software development data collected over a number of years. Cost estimating relationships (CERs) were created, project and mix complexity factors were established, and independent variables were quantified. The result was data base-derived software estimating equations for assembly and high-order language software. These equations and our resulting software parametric models have been validated by comparing project sizing, labor actuals, and schedules with PCEM outputs and documenting the results.

DEVELOPMENT APPROACH

During the early years of our data collection, analysis and model requirements generation activities it was decided that Martin Marietta's software parametric models would include the whole software development life cycle from systems requirements through systems test and provide budget and schedule outputs for the four software development organizations that contribute most to software development. These are:

Systems Engineering,
Software Engineering,
Test Engineering, and
Quality.

Our data base collection approach consists of breaking software actuals out by class, type and language.

Classes of software include:

- Manned flight
- Unmanned flight
- Avionics
- Shipboard/Submarine
- Ground
- Commercial

Types of software are:

Systems Software:	Operating systems and executives.
Support Software:	Simulation, emulation, math models and diagnostic software
Applications Software:	Software that solves the customer's problems.

We collected sizing data by programming language. Our software sizing data base library consists of over 5 million Martin Marietta (Denver) developed source lines of code and over 4 million source lines of code developed by other software development companies and organizations.

At Martin Marietta Denver, we are presently gathering detailed sizing information at the function level to provide additional inputs into our computerized sizing model.

An example of this detailed data is a program of 13,830 SLOC (less comments), of which 9,678 (70%) was programmed in FORTRAN IV and 4,152 SLOC was programmed in assembly language. There were also 1,434 data statements. The sizing summary by computer program component (CPC) consists of the following:

	Function Name	Assy	HOL	Total SLOC	Data State-ments
a)	Executive/Operating System				
	System Control	102	275	377	5
	Interrupt Handling	655	64	719	1
	Interprocessor communcations	75	139	214	0
	Initialization	13	35	48	1
b)	Operator Interface				
	Menu display and automatic generation	0	1,003	1,003	8
	Operator prompting and error checking	0	899	899	4
	Tabular displays	0	485	485	51
	Graphic displays	0	34	34	0
	CRT Formatter	0	22	22	0

c) Data Base Manipulation

Data base generation/regeneration	0	232	323	0
File management	203	94	297	1,116
Data storage and retrieval	0	248	248	9

d) Diagnostics, Fault Determination

Sensor diagnostics	104	3,312	3,416	144
Memory diagnostics	396	1,610	2,006	60
CPU diagnostics	2,510	381	2,891	20

e) Hardware Interface

Peripherals	54	0	54	0
Sensor Device	40	595	635	15
Format manipulation and information conversion	0	159	159	0
	4,152	9,678	13,830	1,434

The "interrupt handling" CPC function level breakout reflected these sizing numbers:

Function Name	Assy	HOL	Total SLOC	Data Statements
Real time interrupt handler (I)	52		52	
Enable/Disable subroutine	5		5	
Real time interrupt handler (II)	10		10	
Keyboard interrupt handler	53		53	
Keyboard handler subroutine	0	50	50	1
Put character	0	14	14	
Disable interrupts routine	8		8	
Enable interrupts routine	10		10	

MS Interrupt handler	79		79	
MSS Interrupt handler	63		63	
Real time interrupt handler	81		81	
STAR PIP interrupt handler	67		67	
ATOD data ready interrupt handler	51		51	
Deuce/STAR threshold data ready interrupt handler	80		80	
	___	___	___	___
	655	64	719	1

The above detailed sizing data along with the cost and schedule information by project provides the input for our detailed analysis and productivity improvement activities.

PARAMETRIC MODELS

The six models described in this paper are all PC-hosted models and trained users carry disks from job site to job site using available compatible PC computers located at the project facilities. These models provide a management capability that has not been available in the past, and there are no subscription costs or mainframe computer delays using these models.

1) <u>Software Parametric Cost Estimating Model (PCEM)</u>

This model provides a method for estimating the total budget, schedule and manloading for a software development activity. The model addresses all phases of software development from systems requirements through systems test. There are two versions of the PCEM model. Version 3.1 reflects MIL-STD-490/483/1679/1521A development. Version 4.0 reflects DOD-STD-2167 and Ada software development.

Description of the Parametric Model

The data based utilized in the Software Parametric Cost Estimating Model (PCEM) consists of "in-house" and "outside" historical software development actuals collected from over 300 completed software development projects.

The data based software projects were separated by "class" and "type" of software. Each class and type has a different complexity and different cost estimating relationships (CERs).

Class of Software

1) Manned space
2) Unmanned space
3) Avionics
4) Shipboard and submarine
5) Ground
6) Commercial

Type of Software

1) Systems Software
2) Applications Software
3) Support Software

Independent Variables

Several independent variables were investigated and the four which were selected and incorporated into the model are summarized below:

1. Lines of Code - The PCEM accepts either source lines of code or machine instructions (object instructions). The amount of functional decomposition performed prior to arriving at a sizing estimate is very important. A great deal of time and analysis is put into reviewing the decomposition so that a good determination of sizing accuracy can be resolved before we input sizing numbers into the PCEM.

2. Project Complexity - Project complexity consists of 14 factors which reflect how well the customer problem is understood and how prepared the contractor is to respond to solving his problem. The factors are weighted and all 14 must be addressed.

1) Requirements Definition
2) Documentation Requirements
3) Experience of Personnel
4) Experience with Equipment/System
5) Amount of Travel Required
6) Language Complexity
7) Interfaces
8) Man Interaction
9) Development Environment
10) Timing and Criticality
11) New or Existing Software
12) Reliability of Test Hardware
13) Testability of Software
14) Operational Hardware Constraints

3. Mix Complexity - The software mix complexity is applied after software sizing has been accomplished. A hundred percent of the identified software lines of code are distributed across the eight mix elements.

The eight elements of mix complexity describe fractions of the total number of source or object instructions, identified by the software engineer.

1) Mathematics
2) String Manipulation
3) Diagnostics, Support Software
4) Data Storage and Retrieval
5) On-line Communcations
6) Realtime Command and Control
7) Man-machine Interaction
8) Systems software

4. Schedule - PCEM determines the optimum schedule and establishes dates for software milestones. The optimum schedule is defined as that period of time when the software can be developed for the least amount of dollars. Costs will increase if the schedule is accelerated, or if it is stretched out beyond the optimum schedule.

With the four independent variables defined along with class and type information, the PCEM can arrive at a total software cost and schedule estimate.

Organizations Included in the PCEM Output:

The PCEM cost equations provide estimates of budget and schedule for the following three software development organizations:

1) Systems Engineering
2) Software Engineering
3) Software Test Engineering

With the information on source or object lines of code, project complexity, mix complexity and user-supplied schedule, the PCEM computerized model can now arrive at the number of manmonths and the schedule required for each of the three software development organizations.

The equations used in the computerized model are arrived at by a multiple regression methodology assessing and analyzing the collected data base information.

Assembly Language and High Order Language CERs

Development Costs

Equation: $Y = a\,(x_1^{\,b_1}) \cdot (x_2^{\,b_2}) \cdot (x_3^{\,b_3}) \cdot (x_4^{\,b_4})$

Where Y = Total Number of Manhours (165 hours = 1 M/M)

x_1 = Estimated Number of Source Lines Code

x_2 = Estimated Project Complexity

x_3 = Estimated Mix Complexity

x_4 = Schedule

a = Constant

b_1, b_2, b_3, b_4 = exponents

Budget and Schedule Information is provided by PCEM for both MIL-STD-490/483/1679/1521A and for DOD-STD-2167 Developments:

Version 3.1 (MIL-STD-490/483/1679/1521A)

SPR SRR SDR PDR CDR TRR TRR AR

REQUIREMENTS			DESIGN		CODE		TEST		
Systems Reqts	Reqts Alloc	Software Reqts	Prel Design	Detail Design	Code	Checkout	Unit Test	Integration PQT FQT	System Test

Version 4.0 (DOD-STD-2167)

SPR SRR SDR SSR PDR CDR TRR TRR FQR

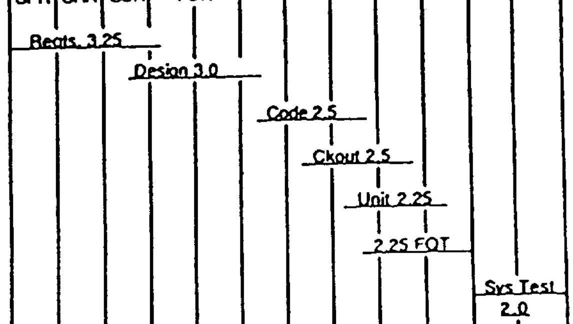
REQUIREMENTS			DESIGN		CODE		TEST		
Systems Concept	Sys S/W Reqts Anal	Software Reqts Anal	Prel Design	Detail Design	Code	Unit Test	CSC Informal Test	CSCI Formal Test	System Integration Test

The computerized PCEM model provides a labor estimate in manmonths, broken out by the phases and subphases of software development. The model identifies an optimum schedule and provides manloading information for each calendar month required for software development. The manmonth estimates are divided between the three organizations that have software development responsibility.

Example Version 3.1:

CALENDAR MONTHS

	1	2	3	4	5	6	7	8	9	10	11	12	
	SPR	SRR	SDR	PDR		CDR		TRR	TRR			AR	



Sys Engr	3.0	3.0	3.0	1.5	1.0	.5	.5	.5	.5	.5	.5	.5	15.0 M/M
S/W Engr	2.5	3.5	4.5	7.0	8.5	10.0	9.5	8.0	6.5	4.5	3.0	2.5	70.0 M/M
Test Engr	.5	.5	.5	.5	.5	.5	1.0	1.5	2.0	3.0	3.5	3.0	17.0 M/M
Total	6	7	8	9	10	11	11	10	9	8	7	6	102.0 M/M

2. Maintenance Model

The computerized "In Scope" maintenance model was recently validated, and became a Parametric Cost Estimating Model (PCEM) output during the first quarter of 1988. The parametric maintenance model is an historical data based derived tool designed to assist users in estimating the cost of "In Scope" maintenance efforts over a few calendar months or over several years. The software maintenance model output includes those efforts related to maintaining the baseline software configuration through error correction and fine tuning activities.

3. Performance Measurement Model

This state-of-the-art software development performance measurement tool was developed during 1988, and permits independent assessment of on-going software development project performance. The user establishes a performance structure which consists of a list of documentation, design reviews, and milestones that the model is going to use to track software development performance. The model provides a measurement of the performance level based on actuals with respect to budget and schedule and estimates a set of "to complete" budget numbers and calendar months for the identified project. During the course of the development the model identifies where the project is performing at either above or below a 100 percent capability.

4. Sizing Model

The software sizing model is a standalone model which is presently undergoing verification and validation testing, but in the very near future it will become a parametric cost estimating model (PCEM) output. The sizing model provides software development engineers with a new concept computerized functionality software sizing capability. The model gives the user a tool to create software development functional decompositions. Once the decomposition is established, the model helps the user create lower level functional decompositions based on whether the software functional element represents a processing task, an input task, or an output task. Software functionality menus containing generic lists allow the user to indicate functional elements that are components of the software systems to be developed. As the user identifies software elements, FORTRAN source lines of code estimates are provided by the sizing model. The model also includes an estimating algorithm for data statements sizing.

5. Risk Analysis Simulation Tool (RAST)

RAST is an interactive computer-based application model that provides a technique for performing quantitative software risk assessment. A major feature of the RAST model is the ability to apply statistics to assess cost risk of proposals and on-going projects. The RAST provides the capability to add, subtract, multiply, and divide Monte Carlo derived distributions and constants.

6. Software Architecture Sizing and Estimating Tool (SASET)

This is a new computerized software cost estimating, scheduling and functional sizing model developed for the naval Center for Cost Analysis in Washington, D.C. The SASET model is a forward-chainging rule-based expert system utilizing a hierarchically structured knowledge data base to provide sizing values, optimal development schedules and various associated manloading outputs depending on complexity and other factors. the model is divided in four separate tiers: Tier I, Project Emulation; Tier II, Sizing; Tier III, Complexity; and Tier IV, Maintenance. The model has recently gone through verification and validation testing and the Air Force, along with the Navy, has just recently (September 1988) provided additional dollars to add a calibration enhancement.

ADA

Martin Marietta Denver has been actively involved with the Ada language since its inception. We particpated in the public evaluation of the Red, Blue, Yellow and Green languages before the Green language was selected as Ada in 1979. Over 200 employees have attended our in-house software engineering Ada training course, and over 200,000 SLOC in Ada have been generated by Martin Marietta students and by engineers on projects using the Ada language. In 1981 Martin purchased the NYU Ada/Ed interpreter for the VAX computer and the demand for a higher performance implementation led to the purchase of a Telesoft/Ada compiler for the VAX/VMS in 1983. Martin Marietta also purchased a validated Rolm Ada Compiler and a Data General Eclipse MV 8000 II computer in 1983. C^3I software developed for a large system started in July 1984 and required rehosting Ada software from the Data General onto a VAX 11/780 computer. During 1987 and 1988 Martin Marietta Denver has won three large command and control projects requiring the use of Ada as the software development language.

CONCLUSIONS

Martin Marietta has one of the largest software development data bases in the country and has been involved in software development data collection, analysis and model building since 1975. Our analysis experts have conducted costing, sizing, scheduling and development management studies on the Ada language for the past several years and have provided new parametric models for Ada management costing and scheduling. Our models and techniques are project tested and geared to providing top management with the tools and resources needed for accurately sizing, costing and scheduling Ada projects and for doing performance measurement on these same projects as they move through the software development process.

THE VIEWGRAPH MATERIALS

FOR THE

W. CHEADLE PRESENTATION FOLLOW

SOFTWARE MANAGEMENT, ESTIMATING, SIZING AND SCHEDULING

PRESENTED BY: W. CHEADLE
MARTIN MARIETTA ASTRONAUTICS GROUP
DENVER DIVISION
MAIL NUMBER L0330
P.O. BOX 179
DENVER, COLORADO 80201

MARTIN MARIETTA

PARAMETRIC MODELS

MARTIN MARIETTA'S DATA BASE DERIVED PARAMETRIC MODELS

- PARAMETRIC COST ESTIMATING MODEL (PCEM) VERSION 3.1
- PARAMETRIC COST ESTIMATING MODEL (PCEM) VERSION 4.1
- MAINTENANCE MODEL
- PERFORMANCE MEASUREMENT MODEL
- SIZING MODEL
- CSCI/CPCI INTEGRATION MODEL
- RISK ANALYSIS SIMULATION TOOL (RAST)
- SOFTWARE ARCHITECTURE SIZING AND ESTIMATING TOOL (SASET)

DATA BASE MARTIN MARIETTA ASTRONAUTICS GROUP

MARTIN MARIETTA DENVER DATA BASE (OVER 300 PROGRAMS)

MARTIN MARIETTA: 29 projects plus 49 other programs
<u>29 projects = 143 programs</u>
192 total

Class of Software: Flight, ground, commercial.

Types of Software: Systems, applications, support.

Languages: HOL (Ada), assembly.

Development Schedule for each Program.

Development Manmonths for each Program.

Organizations Included in Software Development.

Percent of Development Life Cycle.

Source lines of code: 29 projects = 5,026,261 SLOC.

DATA BASE MARTIN MARIETTA ASTRONAUTICS GROUP

OTHER COMPANIES SOFTWARE DEVELOPMENT PROJECTS

Other Companies: 24 projects

24 projects = 110 programs.

Class of Software: Shipboard, ground.

Types of Software: Systems, applications, support.

Languages: HOL (Ada), assembly.

Development Schedule for each Program.

Development Manmonths for each Program.

Organizations Included in Software Development.

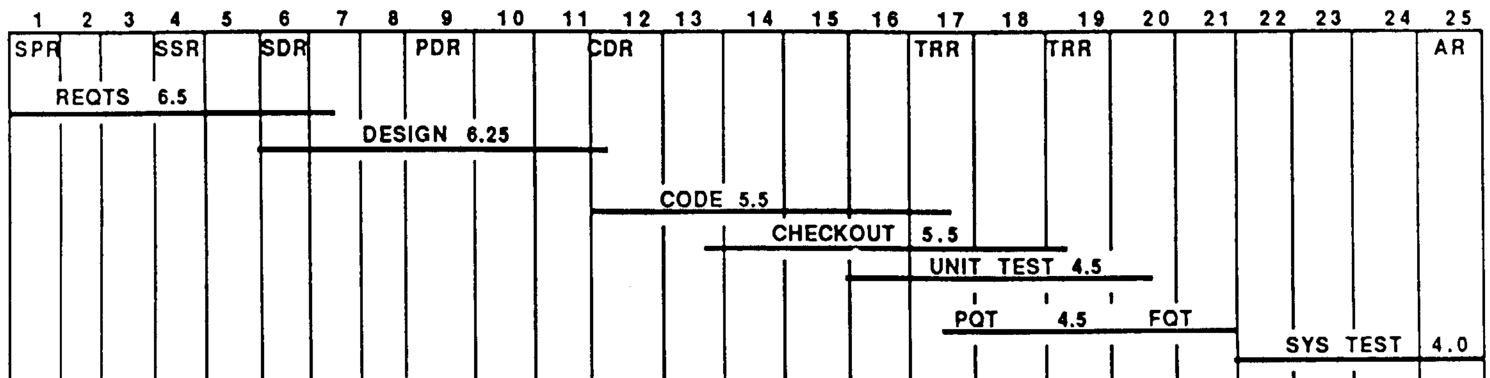
Percent of Development Life Cycle.

Source Lines of Code: 24 projects = 4,282,098 SLOC.

SOFTWARE MANAGEMENT

CLASS OF SOFTWARE : GROUND, NEAR REAL-TIME COMMAND AND CONTROL
CONTRACT TYPE : FPI
PROGRAMMING LANGUAGE : FORTRAN 23,800 NEW SOURCE LINES OF CODE
STANDARD : MIL-STD-1521A
SOFTWARE DEVELOPMENT SCHEDULE : 25 CALENDAR MONTHS
PARAMETRIC COST ESTIMATING MODEL (PCEM) COST AND SCHEDULE ESTIMATE

CALENDAR MONTHS



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	M/M
SYS ENGR	4	4	4	4	4	4	3	3	3	3	2.5	2	1	1	1	1	1	1	.5	.5	.5	.5	.5	.5	.5	50
S/W ENGR	3	4	5	6	7	8	10	11	11.5	11.5	13	14.5	14	13	12	10	10	9.5	9	8	6	5	4	4	4	213
TEST ENGR	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	.5	1	2	2	3	3	3	4	4	5	5	5	5	5	53
QUALITY	.5	.5	.5	.5	.5	.5	.5	.5	1	1	1	1	1	1	1	1	1	.5	.5	.5	.5	.5	.5	.5	.5	17
TOTAL	8	9	10	11	12	13	14	15	16	16	17	18	17	17	16	15	15	14	14	13	12	11	10	10	10	333.0

MARTIN MARIETTA

SOFTWARE DEVELOPMENT

Mil-Stds-490, 483, 1679, 1521A

SPR SRR SDR PDR CDR TRR TRR AR

REQUIREMENTS			DESIGN		CODE		TEST		
System Reqts.	Reqts. Allocation	Software Reqts.	Prelim Design	Detailed Design	Code	Checkout	Unit Test	PQT FQT Integrat.	System Test

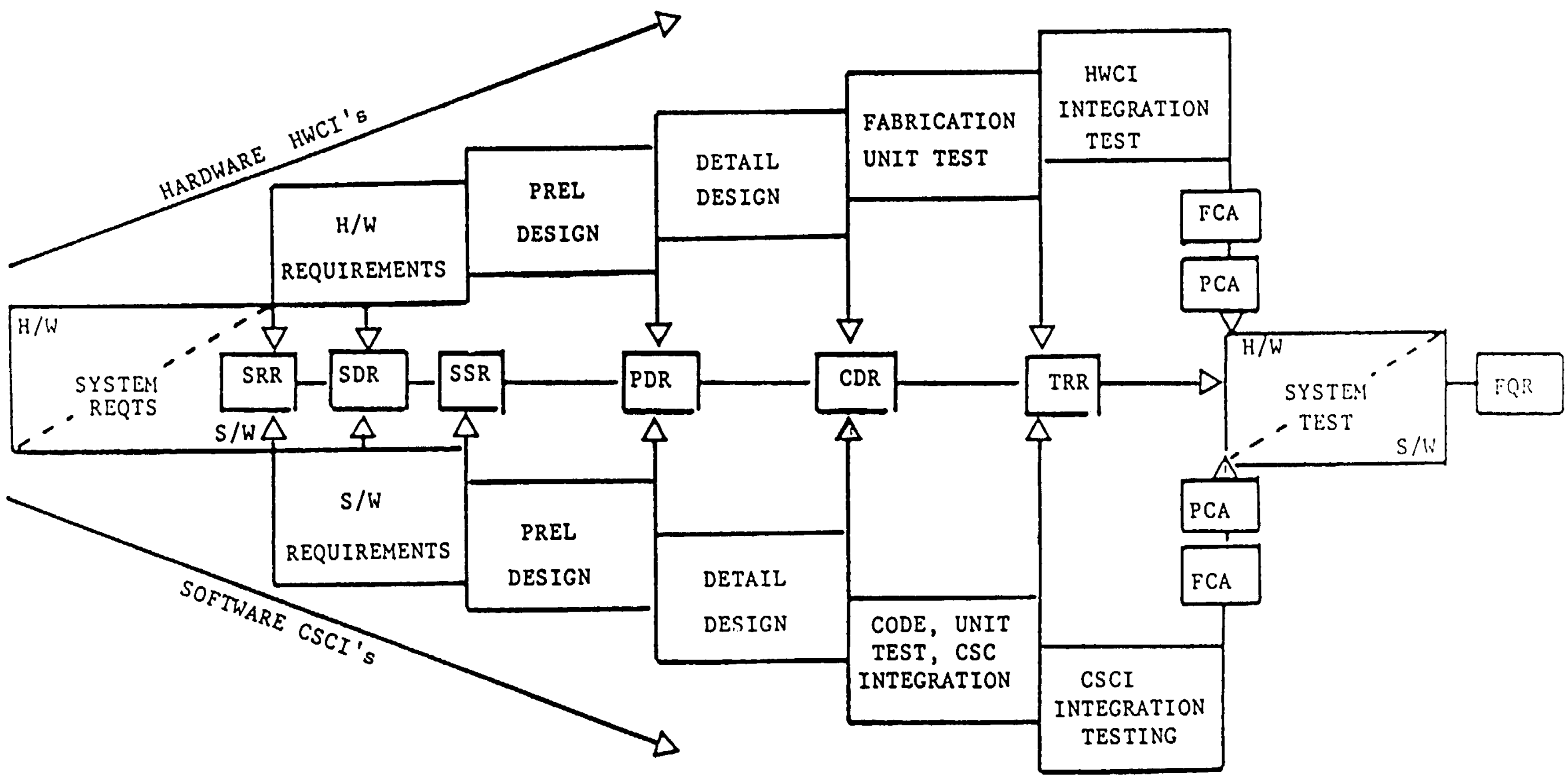
DoD-Std-2167

SPR SRR SDR SSR PDR CDR TRR FCA/PCA FQR

REQUIREMENTS			DESIGN		CODE		TEST		
System Concept	System Software Reqts. Analysis	Software Reqts. Analysis	Prelim Design	Detailed Design	Code	Unit Test	CSC Informal Integrat. Test	CSCI Formal Test	System Integrat. Test

MARTIN MARIETTA

DOD-STD-2167 CONCURRENT HARDWARE AND SOFTWARE DEVELOPMENT



IDEALIZED HARDWARE, SOFTWARE LIFE CYCLE PHASES

SRR	SYSTEM REQUIREMENTS REVIEW	CDR	CRITICAL DESIGN REVIEW	HWCI	HARDWARE CONFIGURATION ITEM
SDR	SYSTEM DESIGN REVIEW	TRR	TEST READINESS REVIEW	CSCI	COMPUTER SOFTWARE CONFIGURATION ITE
SSR	SOFTWARE SPECIFICATION REVIEW	FCA	FUNCTIONAL CONFIGURATION AUDIT	FQR	FORMAL QUALIFICATION REVIEW
PDR	PRELIMINARY DESIGN REVIEW	PCA	PHYSICAL CONFIGURATION AUDIT	CSC	COMPUTER SOFTWARE COMPONENTS

DATA BASE

LANGUAGE RISKS

SYSTEMS SOFTWARE	APPLICATIONS SOFTWARE		SUPPORT SOFTWARE
Machine Instruction	Assembly Language	SPECIAL APPLICATIONS LANGUAGE	Assembly Language
Assembly Language	Jovial	Prolog	Fortran
Pascal	CMS II	LISP	Basic
Fortran	HAL/S	5TH GENERATION USER LANGUAGE	COBOL
Basic	Fortran	M204	Pascal
NEW SYSTEM	COBOL	RAMAS II	C Language
ADA	Basic	IMS	NEW SUPPORT
	Pascal	Total	ADA
	PL/1	Ingres	TEST SEQUENCE LANGUAGE
	C Language		VTL
	NEW APPLICATIONS		STL
	ADA		Comet- H
			GOAL
			HELP
			ATLAS
			SGOS
			CTL

SOFTWARE DEVELOPMENT

SPAGHETTI CODE DEVELOPMENT

CDR

REQUIREMENTS	DESIGN	CODE	TEST
8%	17%	25%	50%
25%		75%	

TOP DOWN STRUCTURED APPROACH (REQTS. DEFINED, DOCUMENTATION EXAMINED AT DESIGN REVIEWS)

SPR SRR SDR PDR CDR TRR PQT FQT AR

REQUIREMENTS	DESIGN	CODE	TEST
23%	22%	20%	35%
45%		55%	

USE OF NEW LANGUAGES, TOOLS AND COMPUTERIZED SYSTEMS (ADA, RPS, AI, ...)

CDR

REQUIREMENTS	DESIGN	CODE	TEST
30%	25%	19%	26%
55%		45%	

MARTIN MARIETTA

HOURS PER SOURCE LINE OF CODE

- Effort encompasses more than coding.
- Each source line of code represents a unit of effort (work).
- Ground applications software development.

Requirements	Design	Code	Test

System Requirements

750-2,000 lines per man month

System Test

150 lines per man month

$$150 \overline{)165 \text{ hours}} = 1.1$$

$$2{,}000 \overline{)165 \text{ hours}} = .0825$$

		Percent
Systems Engr.	.217	14
S/W Engr.	1.1	70
Test Engr.	.25	16
	1.567	100%

$$.70 \overline{)1.567}$$

2.24 hours per SLOC

STUDY APPROACH FOR PREDICTIVE ~~SOFTWARE~~ HARDWARE COST MODELS

- DEFINE INFORMATION COLLECTION REQUIREMENTS AND COLLECT DATA.
- QUANTITATIVELY AND QUALITATIVELY ANALYZE DATA.
- BASED ON DATA ANALYSIS, DEVELOP A DATA BASE THAT WILL INTERFACE AUTOMATICALLY WITH A MODEL.
- DESIGN THE MODEL USING BOTH STATISTICAL AND QUANTITATIVE ANALYSIS TECHNIQUES.
- PROVE THE MODEL BY PERFORMANCE OF VALIDATION AND VERIFICATION TESTING.

MARTIN MARIETTA

SOFTWARE DEVELOPMENT

WHAT CONSTITUTES AN ADA SOURCE LINE OF CODE?

WE CALCULATE ADA SOURCE LINES OF CODE BY COUNTING SEMICOLONS USED AS DELIMITERS, EXCEPT THOSE IN PARENTHESES

NOTE: THIS EXCLUDES SEMICOLONS IN

- COMMENTS
- CHARACTER LITERALS
- STRING LITERALS

MARTIN MARIETTA

ADA SIZING, COSTING, AND SCHEDULING

PER COL. WILLIAM A. WHITAKER:

ADA SOURCE LINES OF CODE ARE CALCULATED BY COUNTING CERTAIN SEMICOLONS.
THERE ARE SEVEN (7) TIMES WHEN SEMICOLONS ARE USED IN ADA.
THREE (3) ARE COUNTED AS SOURCE LINES OF CODE, FOUR (4) ARE NOT.

THE 3 EXAMPLES WHERE SEMICOLONS ARE COUNTED:

1) SEMICOLONS THAT TERMINATE CLAUSES — `WITH TEXT _IO;`
2) SEMICOLONS THAT TERMINATE DECLARATIONS — `A : INTEGER;`
3) SEMICOLONS THAT TERMINATE STATEMENTS — `C := A + B;`

THE 4 EXAMPLES WHERE SEMICOLONS ARE NOT COUNTED:

4) SEMICOLONS THAT TERMINATE PARAMETERS IN A LIST ENCLOSED BY PARENTHESES. `( A : INTEGER ; B : FLOAT )`
5) SEMICOLONS IN COMMENTS `-- TEXT;`
6) SEMICOLONS USED IN SINGLE QUOTATION MARKS (CHARACTER LITERALS) `';'`
7) SEMICOLONS USED IN DOUBLE QUOTATION MARKS (STRING LITERALS) `" A ; B "`

MARTIN MARIETTA

ADA SIZING, COSTING, AND SCHEDULING

```
EXAMPLE ADA PROGRAM                                          LEGEND

  WITH TEXT _10;                                               E
  PROCEDURE EXAMPLE IS                                         A
                                                               B
--THIS IS A COMMENT; NOT A LINE OF CODE                        C
TYPE Z IS RANGE 4 .. 44;                                       D
   CHARACTER_LITERAL: CHARACTER  :=  ';' ;                     D
   STRING_LITERAL: STRING := " x ; y ";                        D
   PROCEDURE FIRST IS (R: IN Z; S: OUT Z) IS SEPARATE;         D
BEGIN                                                          A
  IF (A = 22) THEN                                             A
     B := 4;                                                   S
  END IF;                                                      S
                                                               B
END EXAMPLE ;                                                  D
```

THIS ADA EXAMPLE PROGRAM CONTAINS 14 CARRIAGE RETURNS

```
THERE IS   1  COMMENT STATEMENT                                C
THERE ARE  3  TEXT LINES                                       A
THERE ARE  2  BLANK LINES                                      B
THERE ARE  8  ADA SOURCE LINES OF CODE 5 DECLARATIONS          D
          --                           2 STATEMENTS            S
          14                           1 CLAUSE                E
```

MARTIN MARIETTA

ADA SIZING, COSTING, AND SCHEDULING

ADA LANGUAGE ATTRIBUTES

- STRONG DATA LINKAGE BETWEEN PARENT MODULE & SUBORDINATE MODULES.
- EXCEPTION HANDLING... IN THE EVENT OF AN ERRONEOUS CONDITION, ERRORS WILL BE IDENTIFIED.
- PACKAGES: USED TO GROUP RELATED ENTITIES THAT CAN BE CALLED FROM OUTSIDE THE PACKAGE.
- STRONG TYPING: ENSURES THAT ERRORS ARE DETECTED AT COMPILATION TIME.
- GENERICS: ENCOURAGES RE-USEABILITY, ALLOWS SOME LOGIC STRUCTURE TO BE USED OVER AND OVER.
- TASKING: ALLOWS EVENTS TO BE RUN IN PARALLEL.
- FAULT TOLERANCE: ABILITY OF EITHER H/W OR S/W TO DETECT AN ERROR AND TO RESPOND.

MARTIN MARIETTA

PANEL #3

STUDY OF SOFTWARE PRODUCTS

H. Sayani, Advanced System Technology Corporation
J. Hihn, Jet Propulsion Laboratory
R. LaBaugh, Martin Marietta