A COMPUTERIZED MODEL FOR ESTIMATING SOFTWARE LIFE CYCLE COSTS (MODEL CONCEPT)

Gerard A. Bourdon, Captain, USAF
Joseph A. Duquette, Captain, USAF

April 1978

Approved for Public Release; Distribution Unlimited.

Prepared for
COMPTROLLER
ELECTRONIC SYSTEMS DIVISION
HANSCOM AIR FORCE BASE, MA 01731
LEGAL NOTICE

When U.S. Government drawings, specifications or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

OTHER NOTICES

Do not return this copy. Retain or destroy.

This technical report has been reviewed and is approved for publication.

ROBERT P. SCHIAZZA, GS-15
Chief, Cost Estimating and Analysis Branch
Cost Analysis Division

AUBREY J. LYNCH, Lt Col, USAF
Chief, Cost Analysis Division
Comptroller

FOR THE COMMANDER

DANIEL B. GERAN, Colonel, USAF
Comptroller
Electronic Systems Division
**A COMPUTERIZED MODEL FOR ESTIMATING SOFTWARE LIFE CYCLE COSTS**

**MODEL CONCEPT**

---

1. **REPORT NUMBER**: ESD-TR-77-253
2. **GOVT ACCESSION NO.**: Vol - 1
3. **RECIPIENT'S CATALOG NUMBER**: 
4. **TITLE (and Subtitle)**: A COMPUTERIZED MODEL FOR ESTIMATING SOFTWARE LIFE CYCLE COSTS (MODEL CONCEPT)
5. **TYPE OF REPORT & PERIOD COVERED**: Final
6. **PERFORMING ORG. REPORT NUMBER**: 
7. **AUTHOR(S)**: Gerard A. Bourdon, Captain, USAF
   Joseph A. Duquette, Captain, USAF
8. **CONTRACT OR GRANT NUMBER(S)**: IN-HOUSE
9. **PERFORMING ORGANIZATION NAME AND ADDRESS**: Comptroller
   Electronic Systems Division
   Hanscom AFB, MA 01731
10. **PROGRAM ELEMENT, PROJECT, TASK, AREA & WORK UNIT NUMBERS**: 
11. **CONTROLLING OFFICE NAME AND ADDRESS**: Electronic Systems Division
    Hanscom AFB, MA 01731
12. **REPORT DATE**: April 1978
13. **NUMBER OF PAGES**: 25
14. **MONITORING AGENCY NAME AND ADDRESS (IF DIFFERENT FROM CONTROLLING OFFICE)**: 
15. **SECURITY CLASS. (IF APPROPRIATE)**: UNCLASSIFIED
16. **DISTRIBUTION STATEMENT (OF THIS REPORT)**: Approved for Public Release; Distribution Unlimited.
17. **DISTRIBUTION STATEMENT (OF ABSTRACT ENTERED IN BLOCK 20, IF DIFFERENT FROM REPORT)**: 
18. **SUPPLEMENTARY NOTES**: 
19. **KEY WORDS** (Continue on reverse side if necessary and identify by block number)
   Software Life Cycle Cost
   Software Cost
   Computer Model for Software Cost
   Software Cost Estimating
   Software Cost Model
   Software
   Cost Modeling
20. **ABSTRACT** (Continue on reverse side if necessary and identify by block number)

   This report is the first volume of a series of reports on the development of a computerized model for estimating software life cycle costs. This volume deals with the basic concepts of the model. The report defines the basic stages of the model, the methodologies employed, and the desired features of the model. The report contains enough information to allow operation of the model by manual methods.
CONTENTS

LIST OF FIGURES ii

LIST OF TABLES ii

I. INTRODUCTION TO THE BASIC MODEL 1

   Software Cost Estimating Environment 1
   Reporting Sequence and Rationale 1
   Report Outline 2

II. MODEL OVERVIEW 2

III. STAGE DESCRIPTIONS 4

   Sizing 4
   Manpower 7
   Schedule 9
   Manloadding 11
   Cost 16

IV. MODEL FEATURES 19

V. UNCERTAINTY 21

VI. SUMMARY 22

REFERENCES 23
LIST OF FIGURES

Figure                                                                 Page
I.  BASIC MODEL                                                      3
II. NORDEN MANLOADING CURVE                                         11
III. NORMALIZED MANLOADING CURVE                                    15
IV. CONTRACTOR/GOVERNMENT MANLOADING MIX                             18

LIST OF TABLES

Table                                                                 Page
I.  MANPOWER ERs                                                   8
II. ENVIRONMENTAL FACTORS                                          10
III. DISTRIBUTION OF DEVELOPMENT EFFORT                             14
IV. DOTY VS PUTNAM EXPENDITURE/MANPOWER PROFILE                     14
SECTION I

INTRODUCTION TO THE BASIC MODEL

SOFTWARE COST ESTIMATING ENVIRONMENT

Estimating the costs for developing and operating large scale computer programs (software) is an endeavor subject to extensive error. Schwartz (Ref 8:56) notes that underestimation by a factor of 2.5 is common, while overestimations are unheard of. Devenny (Ref 3) presents data which indicates underestimation factors ranging from 1.1 to 2.6 at the Air Force Systems Command's Electronic Systems Division (ESD). Clapp (Ref 2:9), in her study of software cost estimating methods, notes, "There is only one real problem with software cost estimation: overruns."

The literature is replete with reasons why software cost estimating has historically been so difficult. However, four major obstacles seem to predominate:

1. Requirements are often ill-defined,

2. Available models/methodologies are overly simplistic,

3. Large, homogeneous data bases for model/methodology development do not exist, and

4. Past research efforts have not committed enough resources for successfully accomplishing their often ambitious objectives.

REPORTING SEQUENCE AND RATIONALE

The development, implementation, test, and subsequent modification(s) of this computer model will be accomplished by a series of ESD Technical Reports (TRs). This first TR
lays out the model concept and serves to introduce it to the software cost estimating community. The second TR will depict the model's detailed design, and will include program flowcharts, a listing of the source code, and examples of the model's use. The third TR will report on the model's performance against actual costs on selected software projects. Subsequent TRs will report on modifications to the basic model as reflected in subsequent version releases.

The rationale for issuing technical reports concurrent with model development is to encourage maximum participation on the part of the software cost estimating community. Only by doing so can we overcome the lack of a large homogeneous data base at ESD (and elsewhere) for testing and refining the model. Thus, this model will be tested and refined via widespread dissemination, use, and feedback.

REPORT OUTLINE

In Section III of this report, we discuss the basic model concept, and discuss each submodel, or stage. In Section IV, we discuss the computer implementation of the model, with emphasis on operation options which will be available to the user. Section V discusses how future versions of the model will be modified to handle uncertainty statistically.

SECTION II

MODEL OVERVIEW

The basic model, shown in Figure I, consists of five stages: SIZING, MANPOWER, SCHEDULE, MANLOADING, and COST. At a top level, SIZING serves to transform software functional characteristics into estimates of computer program size. MANPOWER combines a sizing estimate with
characteristics of the development environment to estimate required manpower for development and operation. SCHEDULE employs size to generate a phase-oriented development schedule, and MANLOADING serves to spread the estimated manpower over this schedule. Finally, COST serves to generate constant and then-year cost estimates based on the time-phased manloading profile.

The next section describes each stage in more detail.

SECTION III

STAGE DESCRIPTIONS

SIZING

Virtually every parametric software cost model is based in whole or in part on an estimate of computer program size, measured in terms of source or object instructions. Accurate estimates of program size are usually not available until detailed technical design analyses of the programs in question have been performed. Unfortunately, the cost analyst is often required to provide a cost estimate prior to the accomplishment of these design analyses, working only from a system concept. SIZING serves to estimate program size based upon system functional requirements and proposed hardware characteristics.

SIZING uses one or both of two estimating relationships (ERs) to estimate size: a core requirements ER and an analogy ER.

CORE_REQUIREMENTS ER

Regressing upon data collected by Johns Hopkins University and the MITRE Corporation, Doty Associates obtained the following ER for memory size requirements as a function of application, number of major functions to
be performed by the software, word size, and processor cycle time (Ref 4:173):

\[ M = e^{1.177 + k} \left( N_f \cdot 332 \cdot W_s \cdot 0.147 \cdot t_c^{-0.770} \right) \]

where

- \( \exp = 2.718 \)
- \( M = \) Memory size in thousands of words of object code
- \( N_f = \) Number of major functions (e.g., target tracking, target identification, system monitoring, display, steering, etc) to be performed by the software.
- \( W_s = \) Word size in bits
- \( t_c = \) Processor cycle time* in microseconds
- \( k = \) A constant dependent on application
  - 2.573 for signal processing
  - 2.727 for missile fire control
  - 2.781 for interfacing
  - 3.412 for communication
  - 3.565 for navigation
  - 4.046 for command and control
  - 4.451 for weapon fire control

The coefficient of determination \( (R^2) \) for this ER is .52, and the standard error is .086 ln units.

**ANALOGY ER**

In the same study, Doty derived the following ER for size by analogy to similar (existing) software (Ref 4:174):

\[ M = M_a \left( N_f \right) \]

*Time to either retrieve or store a word in processor memory.*
where

\[ M = \text{Size of the proposed program in thousands of object words} \]

\[ M_a = \text{Size of the analogous programs in thousands of object words} \]

\[ N_{fa} = \text{Number of major functions performed by the analogous software} \]

\[ N_f = \text{Projected number of major functions to be performed by the new software.} \]

Doty does not report either the coefficient of determination nor the standard error for this ER.

**WORDS TO OBJECT TO SOURCE CONVERSIONS**

The core requirements and analogy ERs estimate size in terms of object words. Since MANPOWER employs ERs based upon object and source instructions, the sizing estimate expressed in words must be converted to instructions.

To transform from words to object instructions, the analyst multiplies \( M \) by the blend factor (ratio of average number of instructions per core word required for a process).

To transform object instructions to source instructions, the analyst evaluates the following expression developed by Doty (Ref 5:D-5):

\[ I_s = \frac{I_o}{(1 + P_h (E_h - 1))} \]

where

\[ I_s = \text{Program size in source instructions} \]
\[ I_0 = \text{Program size in object instructions} \]
\[ P_h = \text{Fractional percentage of program to be implemented in Higher Order Language (HOL)} \]
\[ E_h = \text{HOL expansion ratio (average number of object instructions generated per source instruction after program compilation).} \]

If the code is to be written in Assembly language (a source language that includes symbolic machine language statements in which there is a one-to-one correspondence with the executable instruction and data formats of the computer), then \( E_h = 0 \) and \( I_s = I_0 \).

As shown in Figure I, the output of SIZING (Object and/or Source Instructions), serves as input to MANPOWER and SCHEDULE. (Specifically, MANPOWER uses \( I_o \) and \( I_s \), and SCHEDULE uses \( I_o \).)

**MANPOWER**

MANPOWER serves to estimate total manpower requirements through the use of ERs which employ estimated size and environmental factors as the independent variables. The ERs are shown in Table I, along with their coefficients of determination and standard errors.

For those situations where the sizing estimate is in source instructions, and is less than 10,000 instructions, the Table I ERs are of the form

\[ \text{MM} = \frac{b}{a} \prod_{i=1}^{14} f_i \]

where

\[ \text{MM} = \text{Total manmonths to develop the software} \]
<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>SIZING UNITS</th>
<th>ESTIMATED SIZE</th>
<th>DUTY ESTIMATING RELATIONSHIP</th>
<th>COEFFICIENT OF DETERMINATION</th>
<th>STANDARD ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCIENTIFIC</td>
<td>OBJECT</td>
<td>ALL</td>
<td>1.068</td>
<td>MM = 4.495I</td>
<td>.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≥ 10K</td>
<td>1.019</td>
<td>MM = 7.054I</td>
<td>.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt; 10K</td>
<td></td>
<td>MM = 2.011I ( f_i )</td>
<td>.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MM = 7.054I</td>
<td>.78</td>
</tr>
<tr>
<td>COMM &amp; CONTROL</td>
<td>OBJECT</td>
<td>ALL</td>
<td>1.228</td>
<td>MM = 4.573I</td>
<td>.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≥ 10K</td>
<td>1.263</td>
<td>MM = 4.089I</td>
<td>.80</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt; 10K</td>
<td></td>
<td>MM = .501I ( f_i )</td>
<td>.89</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MM = 4.089I</td>
<td>.80</td>
</tr>
<tr>
<td>BUSINESS</td>
<td>OBJECT</td>
<td>ALL</td>
<td>.784</td>
<td>MM = 2.895I</td>
<td>.48</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≥ 10K</td>
<td>.781</td>
<td>MM = 4.495I</td>
<td>.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt; 10K</td>
<td></td>
<td>MM = 3.742I ( f_i )</td>
<td>.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MM = 4.495I</td>
<td>.61</td>
</tr>
<tr>
<td>UTILITY</td>
<td>OBJECT</td>
<td>ALL</td>
<td>.719</td>
<td>MM = 12.039I</td>
<td>.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>≥ 10K</td>
<td>.811</td>
<td>MM = 10.078I</td>
<td>.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt; 10K</td>
<td></td>
<td>MM = 1.744I ( f_i )</td>
<td>.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MM = 10.078I</td>
<td>.45</td>
</tr>
</tbody>
</table>

**TABLE I**
I_s = Number of source instructions
a, b = Constants (derived via regression analysis)
\prod f_i = Product of environmental factors

In all other cases, \prod f_i = 1.

Table II shows the values assigned to the environmental factors if the appropriate environmental conditions are or are not present.

**SCHEDULE**

SCHEDULE serves to estimate the development schedule, and position it into phases

**DEVELOPMENT SCHEDULE**

Regressing upon data from 74 development programs, Doty derived an ER for development time given by

\[
\frac{.667}{t_D} = I_o / (99.25 + 2.33I_o)
\]

where

\[
t_D = "Reasonable" \text{ development time, in months}
\]

\[
I_o = \text{Number of delivered object instructions}
\]

Additionally, Doty's analyses yielded average distribution of schedule and expenditures as shown below

<table>
<thead>
<tr>
<th>Phase</th>
<th>Per Phase</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis</td>
<td>.1 (t_D)</td>
<td>.1 (t_D)</td>
</tr>
<tr>
<td>Design</td>
<td>.33 (t_D)</td>
<td>.43 (t_D)</td>
</tr>
</tbody>
</table>

9
<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>FACTOR</th>
<th>SCIENTIFIC</th>
<th>BUSINESS</th>
<th>COMMAND &amp; CONTROL</th>
<th>UTILITY</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special Display</td>
<td>Detailed Definition of Operational Requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Time Constraint</td>
<td>Concurrency of Hardware Development</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Memory Constraint</td>
<td>CPU Time Constraint</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>First S/M Developed on CPU</td>
<td>Time-Slice, VLSI-Match Processing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Operational Requirements</td>
<td>Operational Requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Real-Time Operations</td>
<td>Real-Time Operations</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**TABLE II**

**ENVIRONMENTAL FACTOR VALUES**

<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>FACTOR</th>
<th>SCIENTIFIC</th>
<th>BUSINESS</th>
<th>COMMAND &amp; CONTROL</th>
<th>UTILITY</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special Display</td>
<td>Detailed Definition of Operational Requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Time Constraint</td>
<td>Concurrency of Hardware Development</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Memory Constraint</td>
<td>CPU Time Constraint</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>First S/M Developed on CPU</td>
<td>Time-Slice, VLSI-Match Processing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Operational Requirements</td>
<td>Operational Requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Real-Time Operations</td>
<td>Real-Time Operations</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**ENVIRONMENTAL FACTOR VALUES**

<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>FACTOR</th>
<th>SCIENTIFIC</th>
<th>BUSINESS</th>
<th>COMMAND &amp; CONTROL</th>
<th>UTILITY</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special Display</td>
<td>Detailed Definition of Operational Requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Time Constraint</td>
<td>Concurrency of Hardware Development</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Memory Constraint</td>
<td>CPU Time Constraint</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>First S/M Developed on CPU</td>
<td>Time-Slice, VLSI-Match Processing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Operational Requirements</td>
<td>Operational Requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Real-Time Operations</td>
<td>Real-Time Operations</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**ENVIRONMENTAL FACTOR VALUES**

<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>FACTOR</th>
<th>SCIENTIFIC</th>
<th>BUSINESS</th>
<th>COMMAND &amp; CONTROL</th>
<th>UTILITY</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special Display</td>
<td>Detailed Definition of Operational Requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Time Constraint</td>
<td>Concurrency of Hardware Development</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Memory Constraint</td>
<td>CPU Time Constraint</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>First S/M Developed on CPU</td>
<td>Time-Slice, VLSI-Match Processing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Operational Requirements</td>
<td>Operational Requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Real-Time Operations</td>
<td>Real-Time Operations</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**ENVIRONMENTAL FACTOR VALUES**

<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>FACTOR</th>
<th>SCIENTIFIC</th>
<th>BUSINESS</th>
<th>COMMAND &amp; CONTROL</th>
<th>UTILITY</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special Display</td>
<td>Detailed Definition of Operational Requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Time Constraint</td>
<td>Concurrency of Hardware Development</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Memory Constraint</td>
<td>CPU Time Constraint</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>First S/M Developed on CPU</td>
<td>Time-Slice, VLSI-Match Processing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Operational Requirements</td>
<td>Operational Requirements</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Real-Time Operations</td>
<td>Real-Time Operations</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Phase</td>
<td>Number of Months</td>
<td>Per Phase</td>
<td>Cumulative</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------</td>
<td>-----------</td>
<td>------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>.09 t_D</td>
<td>.52 t_D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checkout</td>
<td>.085 t_D</td>
<td>.605 t_D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Package Test</td>
<td>.20 t_D</td>
<td>.805 t_D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System Test</td>
<td>.195 t_D</td>
<td>1.0 t_D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**MANLOADING**

The **MANLOADING** stage serves to spread the manpower provided by **MANPOWER** over the schedule(s) provided by **SCHEDULE**. To do so, it applies the software life cycle curve developed by Putnam (Ref 7).

Putnam's software life cycle curve is a tailored version of a generalized project life cycle manloading curve derived and observed by Norden of IBM (Ref 6). The curve, illustrated in Figure II, is given by

\[ Y = 2K a e^{-at^2} \]

\[ K \text{: Total MM/effort} \]

\[ a = \frac{1}{2} t_{max} \]

\[ t = Y_{max} \]

\[ Y_{max} \]

**Figure II**
\[-at^2\]
\[Y = 2 \text{ Kate}\]

where

\[Y = \text{ manmonths/month}\]
\[K = \text{ total life cycle manpower}\]
\[a = \text{ shape parameter}\]
\[e = 2.71828\]

Applying the Nordeh curve to U.S. Army Computer Systems Command software developments, Putnam shows that Norden's curve can be written as

\[Y = \left(\frac{k}{t_D}\right) e^{-t^2/2t_D^2}\]

where

\[t_D\] is approximately equal to the development duration.

In his paper (Ref 7), Putnam uses first principles from physics to prove that the parameters \(K\) (total required manpower), \(t_D\) (development time) and \(K/t_D^2\) (which he defines as "system difficulty"), are fundamental parameters of any software project. Furthermore, he states

"These parameters are natural attributes of a system. Systems are inherently stable and will seek (or be driven to) their natural parameters." (Ref 7:2)

He also corroborates his theoretical findings with empirical evidence on past Army Computer Systems Command software projects.
The hypothesis that development schedule is a "natural" parameter which systems will be driven to also corroborates well with experience outside of Army Computer Systems Command. For example, at ESD the validity of the hypothesis is evident in view of the fact that virtually all software developments have historically experienced schedule slips. (One might argue that these slips are the result of optimism in planning a schedule, acting in place of realism with respect to a natural schedule.)

In further support of this natural schedule phenomenon, Brooks, manager of the IBM 360 Operating System development, observes that "the urgency of the patron may govern the scheduled completion of the task, but it cannot govern the actual completion." (Ref 1:47). He then goes on to state "Brook's Law", which is consistent with Putnam's findings:

"Adding manpower to a late software project makes it later." (Ref 1:48)

To demonstrate this consistency, we note that Putnam's difficulty factor $D = \frac{K}{t_D}$ is also hypothesized and demonstrated empirically to be a natural system parameter. Such being the case, an increase in manpower ($K$) must be accompanied by an increase in development time ($t_D$) if difficulty ($K/t_D$) is to remain constant.

Table III depicts Doty's observation of expenditure profiles in the general case. In Table IV, we compare Doty's expenditure profile with the manpower expenditure profile obtained by evaluating the integral form of Putnam's curve at the appropriate points in the development schedule, and note a close correlation.

MANLOADING uses Putnam's curve to time phase the development manpower generated by MANPOWER against a development schedule generated by SCHEDULE. Additionally, if the user so desires, he can enter an externally provided schedule (such as one imposed by higher headquarters). In this case, MANLOADING generates two manloading profiles (against natural and imposed schedules).
### Distribution of Development Effort

#### Table III

<table>
<thead>
<tr>
<th>DEVELOPMENT PHASE</th>
<th>DEVELOPMENT MILESTONE</th>
<th>AVERAGE DISTRIBUTION OF EFFORT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SCHEDULE %</td>
</tr>
<tr>
<td>ANALYSIS</td>
<td>COMPLETE SYS. DESIGN (PDR)</td>
<td>10</td>
</tr>
<tr>
<td>DESIGN</td>
<td>COMPLETE PKG. DESIGN</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>COMPLETE UNIT DESIGN</td>
<td>8</td>
</tr>
<tr>
<td>CODING</td>
<td>COMPLETE UNIT CODING</td>
<td>9</td>
</tr>
<tr>
<td>CHECKOUT</td>
<td>COMPLETE UNIT DEBUGGING</td>
<td>8.5</td>
</tr>
<tr>
<td>TEST AND INTEGRATE</td>
<td>COMPLETE PKG. TEST</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>COMPLETE SYS. TEST</td>
<td>19.5</td>
</tr>
</tbody>
</table>

#### Distribution of Development Effort

<table>
<thead>
<tr>
<th>DEVELOPMENT MILESTONE</th>
<th>CUMULATIVE DEVELOPMENT</th>
<th>CUMULATIVE EXPENDITURES (DOTY)</th>
<th>CUMULATIVE MANPOWER EXPENDITURES (PUTNAM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPLETE SYS. DESIGN</td>
<td>0.1 $t_d$</td>
<td>3%</td>
<td>1.3%</td>
</tr>
<tr>
<td>COMPLETE PKG. DESIGN</td>
<td>0.35 $t_d$</td>
<td>20%</td>
<td>15.1%</td>
</tr>
<tr>
<td>COMPLETE UNIT DESIGN</td>
<td>0.43 $t_d$</td>
<td>27.5%</td>
<td>22.6%</td>
</tr>
<tr>
<td>COMPLETE UNIT CODING</td>
<td>0.52 $t_d$</td>
<td>38.5%</td>
<td>32.3%</td>
</tr>
<tr>
<td>COMPLETE UNIT DEBUGGING</td>
<td>0.605 $t_d$</td>
<td>48.5%</td>
<td>42.3%</td>
</tr>
<tr>
<td>COMPLETE PKG. TEST</td>
<td>0.805 $t_d$</td>
<td>75.5%</td>
<td>71.0%</td>
</tr>
<tr>
<td>COMPLETE SYS. TEST</td>
<td>1.0 $t_d$</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

CORRELATION R = 0.998

#### DOTY vs Putnam Expenditure/Manpower Profile

Table IV
LIFE-CYCLE MANLOADING

In generating the manloading profile, MANLOADING employs two observations made by Putnam. The first is that development manpower (area under the curve up to $t = t_d$) is approximately 39% of total life cycle manpower. The second is that, in reality, the manpower level for maintenance levels off at $t = 2.38 t_d$, to its value of $Y$ at that point. As shown in Figure III, where $t_d$ and $Y_{\text{max}}$ are normalized to 1, MANLOADING assumes a level manloading at $t = 2.38 t_d$ and simply increments time until the area under the curve equals $K$, at which time the system life cycle is assumed to be over.

As shown in Figure I, the dual schedule output of MANLOADING serves as input to COST.
COST

To this point, the model has generated time phased life cycle manloading requirements. COST serves to generate constant and then-year (escalated) cost estimates by cost category and budget appropriation.

DEVELOPMENT COST

To determine development costs, COST multiplies the time phased manloading profile by a user-supplied average labor rate, in dollars per manmonth (including overhead, and general and administrative costs, as appropriate). It then multiplies the estimated time-phased labor-related costs by cost factors derived by Doty for secondary resources such as computer time, documentation reproduction, and travel. COST then allocates estimated costs on a time phased basis to the following cost categories:

- Systems Engineering (Labor costs for analysis phase)
- Computer Programs (Labor costs from start of design phase to end of package test)
- Systems Test (Labor costs for systems test phase)
- Development Facilities (Computer time)
- Data (Documentation reproduction)
- Project Management (Travel)
- Overhead
- General and Administrative
- Fee

In the original model version, these factors will be single valued percentages. The user can choose to enter
his own percentages, or use default values pre-loaded in the model.

As an option, the user can specify the base year for his average labor rate, and expected starting month for the development. COST then presents the time-phased cost estimate for the above nine cost categories by applying approved Office of the Secretary of Defense (OSD) escalation factors for the 3600 (research and development) appropriation.

**TRANSITION COSTS**

For purposes of this model, the transition period is defined to be that portion of the life cycle between end of development and start of steady state maintenance. Activities during this period involve operational test and evaluation, enhancement, and correction of major design and coding errors. As shown in Figure IV, manpower during this period of time is assumed to consist of a mix of contractor and government personnel, with the government level building up linearly from zero to the steady state maintenance level during this period.

During this phase, COST first calculates the government level and subtracts it from the total level. It then costs the contractor segment (for all nine cost categories) in the same fashion as the development phase. It costs the government portion by applying user-supplied cost factors found in APM 173-10, **USAF Cost Planning Factors**.

**MAINTENANCE COSTS**

COST estimates steady-state maintenance costs by applying user-supplied cost factors found in AFM 173-10 to the steady-state maintenance level.

**COST_RISK**

COST calculates time-phase development, transition, and maintenance costs for manloading profiles based upon
both estimated development schedule and external, imposed
development schedule (if one is provided). In the case
where an external schedule is provided, the model takes
the cost differences between the two profiles on a month-
by-month basis and generates a time-phased schedule cost
risk profile.
SECTION IV
MODEL FEATURES

In Section III, we discussed the basic model structure. In this section, we discuss some of the features which will be implemented in the computer version.

ENTRY POINT SELECTION

Depending upon where the software project is in its life cycle, the analyst may wish to by-pass certain stages. For example, if the project has progressed beyond detailed design, the cost analyst may have contractor provided sizing estimates in which he has high confidence; in that case, he would prefer to by-pass SIZING.

The model will be implemented so as to allow the analyst to select his entry point.

BACK_CHECK_OPTION

This option will allow the analyst to exercise certain stages in reverse, as a check on data he may have received from other sources. For example, suppose he has externally-provided estimates of size, manpower, and development schedule and chooses to enter the model at MANLOADING. By exercising the bench-check option, the analyst can solve the inverse of the Doty ERs to obtain an estimate of size as a function of manpower and environmental factors. The resulting estimated size then serves as a check on the externally-provided sizing estimate.

MULTIPLE_RUNS_MODE

For purposes of sensitivity analysis, the analyst may wish to perform multiple runs, varying only a single variable from run to run. In the Multiple Runs Mode, the
program will ask the analyst to declare that variable which will vary from run to run, and will ask for the value of only that variable on subsequent runs.

INTERIM STAGE OUTPUT MODE

In normal operation, the model will provide only time-phased cost profiles as output. In the Interim Stage Output Mode, the analyst will be able to declare stages for which he desires to receive output. For example, by declaring Stage 4 (MANLOADING), the analyst will direct the model to output the time-phased manloading profiles generated by that stage.
SECTION V

UNCERTAINTY

In the initial version of the model, uncertainty will be handled solely by user-conducted analysis of variations in output as a function of variations in input. Although such sensitivity analyses provide a first cut at analyzing the effects of uncertainty in specifying the system, they do not capture uncertainty inherent to parameters and relationships internal to the model.

Subsequent versions of the model will handle uncertainty inherent to the model by employing a combination of statistical analyses on the regressions underlying the ERs, and Monte Carlo sampling techniques on those parameters and relationships derived by techniques other than regression.

Detailed discussion of the handling of uncertainty via statistical and Monte Carlo techniques will be contained in a subsequent Technical Report.
SECTION VI

SUMMARY

This report has outlined conceptually a software cost model incorporating and integrating recent and relevant research in various aspects of the software life cycle. The computer version of this model is currently (Winter 1978) being designed and implemented at ESD for use as a research and possible estimating tool.

The motivation for publicizing the model prior to its actual implementation is to motivate interest and discussion within the software cost estimating community. Hopefully, this interest and discussion will, in turn, result in widespread model usage and experimentation, with resulting successes and failures relayed back to ESD. Such feedback will be the primary means for testing and improving the model.

Individuals interested in discussing the model are encouraged to call Capt Joseph Duquette at (617)-861-2795, or AUTOVON 478-2795.
REFERENCES


END

DATE FIlMED

6-78
SUPPLEMENTARY

INFORMATION
ERRATA

AD-A053 937

No other volumes will be published per Ms. Dutra, ESD.

DTIC-/DDAC
2 Aug 84