SOFTWARE ENGINEERING LABORATORY SERIES

SEL-93-002

COST AND SCHEDULE ESTIMATION STUDY REPORT

NOVEMBER 1993

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

Foreword

The **Software Engineering Laboratory** (SEL) is an organization sponsored by the National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC) and created to investigate the effectiveness of software engineering technologies when applied to the development of applications software. The SEL was created in 1976 and has three primary organizational members:

NASA/GSFC, Software Engineering Branch
University of Maryland, Department of Computer Science
Computer Sciences Corporation, Software Engineering Operation

The goals of the SEL are (1) to understand the software development process in the GSFC environment; (2) to measure the effects of various methodologies, tools, and models on this process; and (3) to identify and then to apply successful development practices. The activities, findings, and recommendations of the SEL are recorded in the Software Engineering Laboratory Series, a continuing series of reports that includes this document.

The major contributors to this document are

Steve Condon (Computer Sciences Corporation)
Myrna Regardie (Computer Sciences Corporation)
Mike Stark (Goddard Space Flight Center)
Sharon Waligora (Computer Sciences Corporation)

Single copies of this document can be obtained by writing to

Software Engineering Branch
Code 552
Goddard Space Flight Center
Greenbelt, Maryland 20771

Abstract

This report describes the analysis performed and the findings of a study of the software development cost and schedule estimation models used by the Flight Dynamics Division (FDD), Goddard Space Flight Center. The study analyzes typical FDD projects, focusing primarily on those developed since 1982. The study reconfirms the standard SEL effort estimation model that is based on size adjusted for reuse; however, guidelines for the productivity and growth parameters in the baseline effort model have been updated. The study also produced a schedule prediction model based on empirical data that varies depending on application type. Models for the distribution of effort and schedule by life-cycle phase are also presented. Finally, this report explains how to use these models to plan SEL projects.

Keywords: *cost estimation, planning models, reuse, schedule prediction.*

Contents

Section 7. Applying the Planning Models

Appendix A. Summary of Cost and Schedule Models

Appendix B. Sample Subjective Evaluation Form

Appendix C. Effort and Schedule Detailed Distribution

Abbreviations and Acronyms

References

Standard Bibliography of SEL Literature

Figures

Tables

Executive Summary

Introduction

The Software Engineering Laboratory (SEL) has been collecting and interpreting data on software metrics for 16 years. Over the years it has repeatedly refined its models of the software development process as exhibited at the Flight Dynamics Division (FDD) of NASA's Goddard Space Flight Center (GSFC). This Cost and Schedule Estimation Study was undertaken to determine what changes, if any, have taken place in the software development process in recent years and to validate or refine current FDD models. The study analyzed both FORTRAN and Ada projects and focused on three main application types: Attitude Ground Support Systems (AGSSs), telemetry simulators, and dynamics simulators.

The current study sought to expand on the recent research performed for the *Ada Size Study Report* (Reference 1). The SEL introduced Ada in 1985 as a potentially beneficial technology that could improve the software development process. Most Ada systems that have been developed in the FDD are systems that simulate either spacecraft telemetry (telemetry simulators) or spacecraft dynamics (dynamics simulators).

Objective and Scope

The Cost and Schedule Estimation Study was undertaken to

- Review the relationships and models in the SEL literature and recommend a small set of equations to be used by project managers.
- Validate these size, cost, and schedule models against recent projects. Recommend revisions to the current estimation models.

This study sought to answer the following questions:

- Has the SEL effort estimation model changed and does it vary with language and type of application?
- How should the number of developed lines of code (DLOC) be computed to accurately represent total project effort?
- What are the typical productivities for FDD projects?
- Can the data in the SEL database provide any guidelines for enhancing the initial effort estimate, which is based only on size and typical productivity estimates, by including additional estimation factors such as team experience and problem complexity?
- What impact do increased levels of reused code have on a project's cost and schedule?
- What should the schedule estimation model be?

- What are the typical distributions of effort and schedule among life-cycle phases for projects? Are the distributions different from the standard SEL distribution models?
- What is the typical distribution of effort among software development activities for projects? Is it different from the standard SEL model?
- How do the effort and schedule models that are based on end-of-project actuals relate to the recommended SEL planning models for effort and schedule?

Approach

The study researched many preexisting FDD models relating to effort and schedule estimation and evaluated many of these models, using data from over 30 FDD projects, including AGSSs, telemetry simulators, and dynamics simulators, that are representative of the FDD environment. The study team searched for trends in language differences as well as differences in type of application. The recommended models emerged from an elimination process of considering many possible models using multiple combinations of project data.

Conclusions

The study indicates that

- The standard SEL effort estimation equation, based on a size estimate adjusted for reuse, is best for predicting effort in the FDD environment. Of the three effort model parameters—productivity, cost to reuse code, and growth factor—the productivity and reuse cost vary with language, whereas the growth factor varies with the level of reuse. The effort model parameters do not depend on the application type (that is, AGSS, telemetry simulator, or dynamics simulator).
- DLOC (total source lines of code (SLOC) adjusted for reuse) is an accurate basis for estimating total project effort. For FORTRAN projects, DLOC should continue to be computed with a 20-percent weight given to reused SLOC. (The 20-percent weighting is the reuse cost parameter.)
- For Ada projects, DLOC should continue to be computed with a 30-percent weight given to reused SLOC, but this figure may need to be reevaluated in the future. The 30-percent reuse cost for Ada projects was proposed by the *Ada Size Study Report*. At that time only a small number of completed Ada projects were available for analysis, and the Ada process had been evolving from project to project. Since that time only one additional Ada project (POWITS) has been completed and had its final project statistics verified. Today, therefore, the 30-percent Ada reuse cost represents the best model available for FDD Ada simulators, but as more Ada projects are completed, the Ada reuse cost may need to be reevaluated.
- The significant cost savings evidenced by SAMPEX AGSS and SAMPEXTS, two recent projects with very high reuse levels, suggest a divergence from the standard 30-percent and 20-percent reuse costs. For such high-reuse projects as these, a much lower reuse cost may be appropriate, perhaps as low as 10 percent. SAMPEXTS, however, piloted a streamlined development process, combining some documents and combining the preliminary design review (PDR) with the critical design review (CDR); the project's low reuse cost may result from these process changes as well as from the percentage of reused code. Data from more high-reuse projects are needed before certifying this as a trend.

- The productivity experienced on recent FORTRAN AGSSs varied from 3 to 5 DLOC per technical and management hour. For planning purposes, a conservative productivity value of 3.5 DLOC per technical staff/technical management hour is recommended. When support staff hours are included in the plan, an overall productivity rate of 3.2 DLOC per hour should be used.

- The productivity on recent Ada projects showed less variability than it did on the FORTRAN projects. For planning purposes, a productivity of 5.0 DLOC per technical staff/technical management hour is recommended. When support staff hours are included in the plan, an overall productivity rate of 4.5 DLOC per hour should be used.

- The Subjective Evaluation Form (SEF) data in the SEL database provide no demonstrable evidence that inclusion of estimates for such factors as problem complexity or team experience will significantly improve a manager's estimate of project effort. When making estimates for project effort, managers are still encouraged to include such factors as problem complexity or team experience *based on their own personal experience*, but the database of experience represented by the SEF data in the SEL database provides no guidelines.

- For projects with moderate to low code reuse (less than 70 percent), the post-CDR growth in DLOC due to requirement changes and TBDs is commensurate with past SEL experience: 40 percent. For projects with high code reuse (70 percent or more), the post-CDR growth in DLOC is only about half as much (20 percent).

- An exponential model like the Constructive Cost Model (COCOMO) can be used to predict the duration of projects from total project effort; the COCOMO multiplicative factor of 3.3 must be replaced with a factor of 5.0 for AGSSs (6.7 for simulators) when based on management and technical hours and 4.9 for AGSSs (6.5 for simulators) when based on management, technical, and support hours.

- For projects with moderate to low code reuse, the post-CDR growth in schedule is 35 percent. For projects with high reuse, the post-CDR growth in schedule is 5 percent.

- Based on the final project statistics for moderate to low-reuse projects (less than 70-percent code reuse), the distribution of the total effort and schedule among the life-cycle phases is as follows:

Phase	**Effort**	**Schedule**
Design:	24 ± 3%	30 ± 5%
Code:	45 ± 6%	34 ± 6%
Test:	31 ± 5%	36 ± 7%

- Based on the final project statistics for high-reuse projects (70 percent or more code reuse), the distribution of the total effort and schedule among the life-cycle phases is as shown below. The larger standard deviations for high-reuse projects demonstrate that the development process for high-reuse projects is still evolving, resulting in significant variability in the effort distribution. As more high-reuse projects are completed, it should become possible to more accurately model the high-reuse projects.

Phase	Effort	Schedule
Design:	26 ± 14%	37 ± 9%
Code:	38 ± 12%	26 ± 13%
Test:	36 ± 3%	37 ± 6%

- Based on the final project statistics for low-reuse projects, the distribution of the total effort among the software development activities is as follows:

Activity	Effort
Design:	21 ± 4%
Code:	26 ± 4%
Test:	25 ± 5%
Other:	28 ± 9%

- Based on the final project statistics for high-reuse projects, the distribution of the total effort among the software development activities is as follows:

Activity	Effort
Design:	17 ± 5%
Code:	17 ± 6%
Test:	32 ± 6%
Other:	34 ± 8%

- Requirements changes and system growth cause project effort and schedule to diverge from their predicted distributions in the manager's initial plan. In order to minimize the effects of requirements changes and system growth on project cost and schedule, a manager should usually *plan* for the following distributions of the total effort and schedule among the life-cycle phases:

Phase	Effort	Schedule
Design:	30%	35%
Code:	40%	30%
Test:	30%	35%

Recommendations

Recommendations for planning future projects to be developed within the FDD environment include the following:

- The initial effort estimate should be based on the standard SEL effort estimation model with an appropriate growth factor applied:

  Effort = (DLOC / Productivity) × Growth Factor

- DLOC should be computed as follows:

  DLOC = new SLOC + (reuse cost) × reused SLOC

Language	Reuse Cost
FORTRAN	0.2
Ada	0.3

- The total project effort should be computed using the following productivities:

	Productivity (DLOC per hour)	
Type of Effort	FORTRAN	Ada
Technical and Management Only	3.5	5.0
Technical, Management, and Support	3.2	4.5

- The initial effort estimate (DLOC/productivity) should be multiplied by an appropriate growth factor, which varies with the code reuse level. The recommended post-CDR growth factors are as follows:

Code Reuse Level	Growth Factor
Less than 70%	1.4
70% or more	1.2

- The schedule duration should be computed in calendar months, using the total project effort estimate, in staff-months (155 hours per staff month). The effort estimate should include the growth factor. The coefficient, COEFF, of the schedule duration formula varies with the project type and is not dependent on the development language.

$$\text{Schedule Duration} = \text{COEFF} \times (\text{Effort})^{0.3}$$

Type of Effort	**COEFF**	
	AGSS	**Simulator**
Technical and Management Only	5.0	6.7
Technical, Management, and Support	4.9	6.5

- The following percentages are still valid for planning the effort and schedule within various life-cycle phases:

Phase	**Effort**	**Schedule**
Design:	30%	35%
Code:	40%	30%
Test:	30%	35%

Section 1. Introduction

The Software Engineering Laboratory (SEL) is an organization sponsored by the National Aeronautics and Space Administration/Goddard Space Flight Center (NASA/GSFC). It was created in 1977 to investigate the effectiveness of software engineering technologies applied to the development of applications software. The SEL has three primary organizational members: NASA/GSFC, Software Engineering Branch; University of Maryland, Department of Computer Science; and Computer Sciences Corporation, Software Engineering Operation.

Applications developed in the NASA Flight Dynamics Division (FDD) environment are used primarily to determine and predict the orbit and attitude of Earth-orbiting satellites. All of the operational Attitude Ground Support Systems (AGSSs) developed by the FDD have been written in FORTRAN. Until the late 1980s the systems developed in the FDD to simulate either spacecraft telemetry (telemetry simulators) or spacecraft dynamics (dynamics simulators) were also developed in FORTRAN. Beginning in 1987, however, these simulators began to be developed in Ada.

1.1 Motivation for Study

The SEL has been collecting and interpreting data on software metrics for 16 years. Over the years it has repeatedly refined its models of the software development process as exhibited at the FDD. The Cost and Schedule Estimation Study was undertaken to determine what changes, if any, have taken place in the software development process in recent years and to validate or refine current FDD models. The study analyzed both FORTRAN and Ada projects and focused on three main application types: AGSSs, telemetry simulators, and dynamics simulators.

1.2 Document Organization

Section 1 describes the motivation for the study and the document's organization. Section 2 discusses the data used in the study. Section 3 presents and validates models used to estimate total project effort. These models are followed by other models depicting the distribution of project effort by life-cycle phase and by software development activity. Section 4 analyzes the benefit of adjusting initial effort or productivity estimates to take into account such factors as problem complexity or team experience. Section 5 presents and examines the models used to estimate total project duration and life-cycle phase duration. Section 6 gives the study's conclusions and recommendations. Section 7 describes how to apply the planning models produced by this study.

Appendix A contains a matrix of costing and scheduling formulas recommended in the FDD over the last 14 years. Appendix B contains a sample of the Subjective Evaluation Form (SEF) that is completed at the end of each FDD software development project. Appendix C contains project-by-project data on the distribution of effort and schedule by life-cycle phase and also the distribution of effort by software development activity.

Section 2. Data Used in Study

The Cost and Schedule Estimation Study analyzed both objective and subjective data for the projects studied. Objective data, taken primarily from the SEL database but with occasional reference to the software development history reports, included such data as the hours of effort expended, the number of lines of new and reused code, and the beginning and end dates of life-cycle phases in the final project schedules. These objective data are presented in the tables in this section and are described in the accompanying text. These data were used to support the effort model analysis presented in Section 3 and the schedule model analysis presented in Section 5. For some of the projects, supporting subjective data were obtained from the software development history reports and from discussions with developers. Additional extensive subjective data were taken from the Subjective Evaluation Form (SEF) data in the SEL database in order to support the analysis of subjective factors, discussed in Section 4.

Table 2-1 lists the projects studied along with their application type, language, development period, duration, and the total effort charged by technical staff and managers (but excluding support staff).

In the SEL, source lines of code (SLOC) are defined to include source lines, comment lines, and blank lines. Table 2-2 presents a detailed picture of SLOC for each project, classifying the total SLOC into four categories:

- Newly written code (i.e., code for entirely new units)
- Extensively modified code (i.e., code for reused units in which 25 percent or more of the lines were modified)
- Slightly modified code (i.e., code for reused units in which less than 25 percent of the lines were modified)
- Verbatim code (i.e., code for units that were reused verbatim)

For estimation purposes, SLOC figures are often classified into two overall categories that combine newly written code and extensively modified code under the title *new code* and slightly modified code and verbatim code under the title *reused code*. Table 2-3 presents the figures for new code, reused code, total SLOC, and the percentage of reused code. This reuse percentage is defined simply as the number of lines of reused code divided by the total number of SLOC. For PAS, for example, this would be 27,139/111,868, or 24 percent.

The number for new code is combined with a weighted value for the reused code to yield the number of DLOC as shown in Equation 2-1. Table 2-4 presents the project totals for SLOC and DLOC side by side for comparison. This study used 20 percent for the FORTRAN reuse cost and 30 percent for the Ada reuse cost. It also includes the total project effort charged by technical staff, technical management, and support staff (upper management, librarians, Technical Publications, and secretarial).

$$DLOC = (New\ SLOC) + (Reuse\ Cost) \times (Reused\ SLOC) \quad (2\text{-}1)$$

In order to effectively staff a project, a manager needs to know how much effort will be required in each development phase. Table 2-5 presents the effort in each of the three major life-cycle phases; system test and acceptance test are considered as one overall test phase. The effort hours shown for each major phase, as well as the total hours for all three phases, reflect the hours charged by technical staff and technical management, i.e., those personnel submitting Personnel Resource Forms (PRFs) to the SEL database (see Reference 2). Note that the additional effort total shown in Tables 2-1 and 2-4 also include hours charged during preproject and cleanup phases. In addition, Table 2-4 lists the support staff hours from preproject through cleanup phases. The numbers in Table 2-5 were used to test the accuracy of various models for predicting effort by phase (see Section 3.3).

In addition to data on each life-cycle phase, the SEL database collects and maintains data on the number of hours spent by technical personnel in each of the identified software development *activities* regardless of the life-cycle phase in which the activity occurs. These activities are slightly different in the Cleanroom software development process than in the standard software development process (see Reference 3). To analyze these data more easily, this study grouped these activities into four overall categories named for the life-cycle phase in which its activities were felt to predominate (Table 2-6). The activity hours in each category are presented in Table 2-7. The numbers in each column reflect the hours charged by technical personnel to that overall activity from design phase through test phase.

Another focus of this study was the analysis of the projects' schedules. The number of weeks spent on each project in each of the four main life-cycle phases is depicted in Table 2-8. In this table the test phase is broken out into system test and acceptance test phases just for information. Elsewhere in this study these two formerly separate test phases are treated as one combined test phase.

Table 2-1. Projects Studied

Project	Type	Lang.	Devel. Period [1]	Duration (Weeks)	Tech. & Mgmt.[6] Hours
PAS	AGSS	F	05/76 - 09/77	69	15760
ISEEB	AGSS	F	10/76 - 09/77	50	15262
AEM	AGSS	F	02/77 - 03/78	57	12588
SEASAT	AGSS	F	04/77 - 04/78	54	14508
ISEEC	AGSS	F	08/77 - 05/78	38	5792
SMM	AGSS	F	04/78 - 10/79	76	14371
MAGSAT	AGSS	F	06/78 - 08/79	62	15122
FOXPRO	AGSS	F	02/79 - 10/79	36	2521
DEA	AGSS	F	09/79 - 06/81	89	19475
DEB	AGSS	F	09/79 - 05/81	83	17997
DESIM	TS	F	09/79 - 10/80	56	4466
ERBS	AGSS	F	05/82 - 04/84	97	49476
DERBY	DS	F	07/82 - 11/83	72	18352
GROSS	DS	F	12/84 - 10/87	145	15334
GRODY	DS	A	09/85 - 10/88	160	23244
COBEDS	DS	F	12/84 - 01/87	105	12005
ASP	AGSS	F	01/85 - 09/86	87	17057
GROAGSS	AGSS	F	08/85 - 03/89	188	54755
GROSIM	TS	F	08/85 - 08/87	100	11463
COBSIM	TS	F	01/86 - 08/87	82	6106
COBEAGSS	AGSS	F	06/86 - 09/88	116	49931
GOADA	DS	A	06/87 - 04/90	149	28056
GOFOR	DS	F	06/87 - 09/89	119	12804
GOESAGSS	AGSS	F	08/87 - 11/89	115	37806
GOESIM	TS	A	09/87 - 07/89	99	13658
UARSAGSS	AGSS[2]	F	11/87 - 09/90	147	89514
ACME	AGSS[2]	F	01/88 - 09/90	137	7965
UARS_2	AGSS[2]	F	N/A	N/A	97479
UARSDSIM	DS	F	01/88 - 06/90	128	17976
UARSTELS	TS	A	02/88 - 12/89	94	11526
EUVEAGSS	AGSS	F	10/88 - 09/90	102	21658
EUVE_2[3]	AGSS	F	N/A	N/A	21658
EUVETELS	TS	A	10/88 - 05/90	83	4727
EUVEDSIM	DS	A	10/88 - 09/90	121[4]	20775[4]
SAMPEXTS	TS	A	03/90 - 03/91	48	2516
SAMPEX	AGSS[5]	F	03/90 - 11/91	85	4598
SAMPEXTP	AGSS[5]	F	03/90 - 11/91	87	6772
SAMPEX_2	AGSS[5]	F	N/A	N/A	11370
POWITS	TS	A	03/90 - 05/92	111	11695

[1] Design phase through acceptance test phase.

[2] The AGSS for the UARS satellite was developed as two projects. One project, containing the majority of the AGSS code and functionality, was called simply UARSAGSS and was developed by CSC. The other project, containing two utilities (CFADS and STARID), was called ACME and was developed inhouse by GSFC. When referring to the total size or effort of the two combined projects, this study uses the name UARS_2.

[3] The EUVE AGSS was developed as a single project, and the EUVEAGSS account in the SEL database includes all hours spent on this AGSS. In recording the lines of code in the EUVEAGSS account, however, the SEL database did not include the ACME lines of code, all of which were borrowed from the ACME project and reused verbatim in the EUVE AGSS. When referring to the size or productivity of the total EUVE AGSS, this study uses the name EUVE_2. The values for effort and schedule duration do not vary between EUVE AGSS and EUVE_2.

[4] Duration adjusted by +15% and Effort adjusted by +10% because EUVEDSIM did not have an acceptance test phase. These values are consistent with those of the *Ada Size Study Report*.

[5] The AGSS for the SAMPEX satellite was developed as two projects. The telemetry processor part, called SAMPEXTP, was developed inhouse by GSFC. The other project, containing the majority of the AGSS code and functionality, was called simply SAMPEX and was developed by CSC. When referring to the total size or effort of the two combined projects this study uses the name SAMPEX_2.

[6] Includes technical staff and technical management hours for preproject through cleanup phases. Does not include support staff hours (project management, librarians, secretaries, technical publications).

A	Ada
AGSS	Attitude Ground Support System
DS	dynamics simulator
F	FORTRAN
TS	telemetry simulator

Table 2-2. Detailed Line-of-Code Data

Project Name	Newly Written	Extensively Modified	Slightly Modified	Verbatim
PAS	84729	0	20041	7098
ISEEB	43955	0	3506	7776
AEM	45345	0	4673	893
SEASAT	49316	0	4252	21825
ISEEC	20075	0	6727	48618
SMM	76883	0	5652	2834
MAGSAT	61950	0	14297	13266
FOXPRO	5354	0	1323	2449
DEA	45004	0	9705	12616
DEB	44644	0	8606	13016
DESIM	14873	0	0	385
ERBS	137739	0	5767	15635
DERBY	37137	0	3901	4549
GROSS	33196	3493	8574	6441
GRODY	123935	1143	3037	146
COBEDS	26986	0	7363	2556
ASP	70951	0	0	10483
GROAGSS	194169	9982	18133	14109
GROSIM	31775	0	4294	2881
COBSIM	45825	1342	1156	4494
COBEAGSS	141084	16017	13647	7934
GOADA	109807	12496	41750	7049
GOFOR	22175	2867	6671	5330
GOESAGSS	106834	6377	9779	5869
GOESIM	59783	5784	15078	11450
UARSAGSS	260382	9340	21536	11868
ACME	34902	0	0	0
UARS_2	295284	9340	21536	11868
UARSDSIM	63861	17476	20710	4399
UARSTELS	38327	6114	12163	11544
EUVEAGSS	41552	13597	14844	179016
EUVE_2	41552	13597	14844	213918
EUVETELS	2161	371	5573	58591
EUVEDSIM	20859	36248	87415	39495
SAMPEXTS	0	3301	6120	52026
SAMPEX	10590	1631	1282	141006
SAMPEXTP	15899	1920	1777	36
SAMPEX_2	26489	3551	3059	141042
POWITS	12974	7980	20878	26275

Table 2-3. Line-of-Code Summary Data

Project Name	New Code[1]	Reused Code[2]	Total[3]	Reuse Percentage[4]
PAS	84729	27139	111868	24%
ISEEB	43955	11282	55237	20%
AEM	45345	5566	50911	11%
SEASAT	49316	26077	75393	35%
ISEEC	20075	55345	75420	73%
SMM	76883	8486	85369	10%
MAGSAT	61950	27563	89513	31%
FOXPRO	5354	3772	9126	41%
DEA	45004	22321	67325	33%
DEB	44644	21622	66266	33%
DESIM	14873	385	15258	3%
ERBS	137739	21402	159141	13%
DERBY	37137	8450	45587	19%
GROSS	36689	15015	51704	29%
GRODY	125078	3183	128261	2%
COBEDS	26986	9919	36905	27%
ASP	70951	10483	81434	13%
GROAGSS	204151	32242	236393	14%
GROSIM	31775	7175	38950	18%
COBSIM	47167	5650	52817	11%
COBEAGSS	157101	21581	178682	12%
GOADA	122303	48799	171102	29%
GOFOR	25042	12001	37043	32%
GOESAGSS	113211	15648	128859	12%
GOESIM	65567	26528	92095	29%
UARSAGSS	269722	33404	303126	11%
ACME	34902	0	34902	0%
UARS_2	304624	33404	338028	10%
UARSDSIM	81337	25109	106446	24%
UARSTELS	44441	23707	68148	35%
EUVEAGSS	55149	193860	249009	78%
EUVE_2	55149	228762	283911	81%
EUVETELS	2532	64164	66696	96%
EUVEDSIM	57107	126910	184017	69%
SAMPEXTS	3301	58146	61447	95%
SAMPEX	12221	142288	154509	92%
SAMPEXTP	17819	1813	19632	9%
SAMPEX_2	30040	144101	174141	83%
POWITS	20954	47153	68107	69%

[1] Includes newly written code and extensively modified code.

[2] Includes slightly modified code and verbatim code.

[3] New code plus reused code.

[4] Reused code divided by total SLOC.

Table 2-4. SLOC, DLOC, and Effort

Project Name	SLOC	DLOC[1]	Tech. & MGMT[2] Hours	Support[3] Hours
PAS	111868	90157	15760	4316
ISEEB	55237	46211	15262	1378
AEM	50911	46458	12588	1109
SEASAT	75393	54531	14508	1231
ISEEC	75420	31144	5792	1079
SMM	85369	78580	14371	2744
MAGSAT	89513	67463	15122	1926
FOXPRO	9126	6108	2521	528
DEA	67325	49468	19475	2846
DEB	66266	48968	17997	3267
DESIM	15258	14950	4466	1194
ERBS	159141	142019	49476	5620
DERBY	45587	38827	18352	1870
GROSS	51704	39692	15334	2207
GRODY	128261	126033	23244	2560
COBEDS	36905	28970	12005	1524
ASP	81434	73048	17057	1875
GROAGSS	236393	210599	54755	4718
GROSIM	38950	33210	11463	796
COBSIM	52817	48297	6106	0
COBEAGSS	178682	161417	49931	4313
GOADA	171102	136943	28056	2125
GOFOR	37043	27442	12804	894
GOESAGSS	128859	116341	37806	2876
GOESIM	92095	73525	13658	1290
UARSAGSS	303126	276403	89514	7854
ACME	34902	34902	7965	0
UARS_2	338028	311305	97479	7854
UARSDSIM	106446	86359	17976	1987
UARSTELS	68148	51553	11526	1034
EUVEAGSS	249009	93921	21658	2538
EUVE_2	283911	100901	21658	2538
EUVETELS	66696	21781	4727	855
EUVEDSIM	184017	95180	20775	2362
SAMPEXTS	61447	20745	2516	756
SAMPEX	154509	40679	4598	685
SAMPEXTP	19632	18182	6772	0
SAMPEX_2	174141	58861	11370	685
POWITS	68107	35100	11695	308

[1] Based on 20% reuse cost for FORTRAN projects and 30% reuse cost for Ada projects.

[2] Includes technical staff and technical management hours for preproject through cleanup phases.

[3] Includes upper management, librarians, Tech Pubs, and secretarial hours for preproject through cleanup phases.

Table 2-5. Technical Staff Hours[1] Distributed by Life-Cycle Phase

Project Name	Design Phase	Code Phase	Test Phase	3-Phase Total
PAS	2761	8775	3840	15376
ISEEB	2871	7485	2750	13106
AEM	2347	6102	3670	12119
SEASAT	3516	6817	3470	13802
ISEEC	1806	2433	1850	6090
SMM	4533	6373	4394	15300
MAGSAT	3315	5858	5955	15128
FOXPRO	439	653	1210	2301
DEA	3187	9682	6551	19421
DEB	3565	8846	5388	17798
DESIM	1427	1766	822	4015
ERBS	10548	24467	13040	48055
DERBY	5001	7872	4340	17213
GROSS	3679	5397	6089	15165
GRODY	2987	11174	4972	19133
COBEDS	4008	3559	4639	12206
ASP	3854	7271	5854	16979
GROAGSS	11416	28132	14329	53877
GROSIM	2240	4751	3942	10933
COBSIM	1434	2388	1822	5644
COBEAGSS	11012	18173	18410	47595
GOADA	7170	10815	7901	25886
GOFOR	1898	3853	6482	12233
GOESAGSS	6844	19892	9808	36543
GOESIM	3712	5763	3565	13039
UARSAGSS	16592	42473	26612	85676
ACME	2870	3723	985	7577
UARS_2	19462	46196	27597	93253
UARSDSIM	3100	7914	6182	17195
UARSTELS	2751	4402	4014	11167
EUVEAGSS	2881	9926	7732	20539
EUVETELS	1107	1718	1411	4235
EUVEDSIM	4258	8846	4701	17805
EUVEDSIM(rev)	4258	8846	6679	19783
SAMPEXTS	981	368	690	2038
SAMPEX	1189	732	2578	4498
SAMPEXTP	1709	3330	1600	6639
SAMPEX_2	2898	4062	4178	11137
POWITS	1588	5493	4597	11677

[1] Includes technical staff and technical management hours for the phases listed; does not include preproject hours or cleanup phase hours; does not include support staff (upper management, librarians, secretaries, Tech Pubs) hours.

Table 2-6. Groupings of Software Development Activities

Overall Category	SOFTWARE DEVELOPMENT ACTIVITIES Standard Development Process	Cleanroom Process
Design	Predesign Create Design Read/Review Design	Predesign Create Design Verify/Review Design
Coding	Write Code Read/Review Code Unit Test Code	Write Code Read/Review Code
Testing	Debugging Integration Test Acceptance Test	Pretest Independent Test Response to SFR Acceptance Test
Other	Other	Other

Table 2-7. Technical Staff Hours Distributed by Development Activity

Project Name	Design Activity	Coding Activity	Test Activity	Other Activity	Tech. Staff Hours
PAS	1028	3873	2092	8383	15376
ISEEB	2125	2972	1313	6696	13106
AEM	2383	3144	1928	4664	12119
SEASAT	1959	3687	1935	6222	13802
ISEEC	1764	1730	395	2201	6090
SMM	4038	4153	2188	4920	15300
MAGSAT	3849	3828	2760	4691	15128
FOXPRO	741	623	393	544	2301
DEA	2940	3655	4826	8001	19421
DEB	3557	3872	2899	7471	17798
DESIM	1160	938	574	1344	4015
ERBS	8798	14024	8019	17213	48055
DERBY	4562	2254	2558	7839	17213
GROSS	3534	4253	2615	4762	15165
GRODY	4909	6467	2925	4832	19133
COBEDS	2982	2538	1966	4721	12206
ASP	2487	3599	4032	6861	16979
GROAGSS	10829	15642	11124	16283	53877
GROSIM	2408	3560	1681	3285	10933
COBSIM	1269	1759	813	1802	5644
COBEAGSS	11465	10545	13166	12419	47595
GOADA	4967	7209	6131	7579	25886
GOFOR	1427	2260	4792	3754	12233
GOESAGSS	9256	11610	8976	6702	36543
GOESIM	2503	2973	3081	4483	13039
UARSAGSS	20561	24940	24710	15465	85676
ACME	2195	1320	2370	1693	7577
UARS_2	22756	26259	27080	17158	93254
UARSDSIM	3117	5831	4707	3542	17195
UARSTELS	2160	3067	3715	2226	11167
EUVEAGSS	4419	5133	6437	4551	20539
EUVETELS	644	711	1111	1771	4235
EUVEDSIM	3732	5348	3807	4918	17805
SAMPEXTS	341	338	546	814	2038
SAMPEX	654	290	1371	2185	4498
SAMPEXTP	1802	697	2620	1521	6639
SAMPEX_2	2455	986	3991	3705	11138
POWITS	1072	2209	4760	3636	11677

Table 2-8. Schedule Distribution (Calendar Weeks)

Project Name	Design	Code	Systest	Acctest	4-Phase[1] Total
PAS	19	32	9	9	69
ISEEB	21	21	4	4	50
AEM	16	26	9	6	57
SEASAT	17	24	5	8	54
ISEEC	16	14	4	4	38
SMM	24	24	9	19	76
MAGSAT	19	24	9	10	62
FOXPRO	16	10	4	6	36
DEA	32	42	4	11	89
DEB	32	31	10	10	83
DESIM	28	20	4	4	56
ERBS	42	33	12	10	97
DERBY	26	23	8	15	72
GROSS	23	29	18	75	145
GRODY	27	67	56	10	160
COBEDS	36	24	33	12	105
ASP	26	27	13	21	87
GROAGSS	44	75	31	38	188
GROSIM	35	39	17	9	100
COBSIM	23	33	15	11	82
COBEAGSS	31	31	24	30	116
GOADA	41	43	46	19	149
GOFOR	30	33	38	18	119
GOESAGSS	31	44	19	21	115
GOESIM	34	29	8	28	99
UARSAGSS	45	53	24	25	147
ACME	42	54	15	26	137
UARSDSIM	33	58	9	28	128
UARSTELS	30	28	10	26	94
EUVEAGSS	38	34	15	15	102
EUVETELS	22	35	10	16	83
EUVEDSIM[2]	33	43	27	18	121
SAMPEXTS	23	4	8	13	48
SAMPEX	39	12	19	15	85
SAMPEXTP	27	33	14	13	87
POWITS	29	35	9	38	111

[1] System test and acceptance test phase durations are broken out separately in this table just for information. Elsewhere in this study, these formerly separate phases are treated as one combined test phase.

[2] Includes 18 weeks added to the schedule to create an artificial acceptance test phase (equal to 15% of the project duration).

Section 3. Effort Analysis

This section derives and validates models for estimating total effort, life-cycle phase effort, and software development activity effort.

Section 3.1 introduces and discusses the basic effort model for total effort. This model includes parameters for reuse cost and productivity but does not model post-CDR growth.

Section 3.2 then validates two slightly different versions of the effort model, the original model without a growth factor and the same model with a growth factor added. First it validates the original model using end-of-project values for both SLOC and effort. Following this it adds a post-CDR growth factor to the model, inserts the *CDR SLOC estimates* into the model, and validates the model against the actual end-of-project effort.

Section 3.3 discusses the models for the distribution of technical staff effort by life-cycle phase.

Section 3.4 presents the models for distribution of technical staff effort by software development activity.

3.1 Reuse Cost Analysis, Productivity, and Total Project Effort

One of the primary concerns in planning and managing a software project is determining the total effort (measured in staff hours or staff months) required to complete the project. The effort depends primarily upon the extent of the work, and the simplest and most reliable measure yet found for describing the size of a software project in the SEL is the number of SLOC that it contains. In the SEL, SLOC are defined to include source lines, comment lines, and blank lines.

Borrowing code written for an earlier software project and adapting it for the current project often requires less effort than writing entirely new code. Testing reused code also typically requires less effort, because most of the software errors in the reused code have already been eliminated. Therefore, if a software project makes significant use of reused code, the project will usually require less overall effort than if it had written all of its code from scratch.

When planning a project, FDD managers multiply the reused SLOC by a *reuse cost factor*, in order to reflect the reduced cost of using old code. Adding the resulting weighted value for the reused SLOC to the number of new SLOC yields what the SEL calls the DLOC, as shown in Equation 3-1. The DLOC number is the standard measure for the size of an FDD software project.

$$\text{DLOC} = (\text{New SLOC}) + (\text{Reuse Cost}) \times (\text{Reused SLOC}) \qquad (3\text{-}1)$$

The traditional reuse cost for FDD projects is 20 percent, and this remains the recommended standard for FORTRAN projects. The recently developed SEL model for Ada projects, however, recommends using a reuse cost of 30 percent (see Equations 3-1A and 3-1B).

$$\text{FORTRAN DLOC} = \text{new SLOC} + 0.2 \times \text{reused SLOC} \tag{3-1A}$$

$$\text{Ada DLOC} = \text{new SLOC} + 0.3 \times \text{reused SLOC} \tag{3-1B}$$

The 30-percent reuse cost for Ada projects was proposed by the *Ada Size Study Report*. At the time that study was conducted, the Ada process was still evolving from project to project, and only a small number of completed Ada projects were available for analysis. Since then only one additional Ada project, POWITS, has been completed and had its final project statistics verified. (The WINDPOLR final project statistics were verified too recently to be included in this report.) Today, therefore, the 30-percent Ada reuse cost still represents the best available model for FDD Ada simulators. As more Ada simulators are completed, however, a clearer picture of the standard Ada development process may become discernible. At that time the Ada reuse cost should be reevaluated.

This reevaluation is particularly advisable in light of changing development practices on high-reuse projects. These practices sometimes include combining the PDR with the CDR and also combining or structuring related documents in such a way as to reuse large portions of documents. As the process for developing projects with high software reuse becomes more consistent, and as more high-reuse projects are finalized in the database, it should be possible to modify the SEL effort model to better reflect these projects. This may include revising the recommended parameters for reuse cost and productivity.

The SEL has collected statistics on over 100 software projects during the past 2 decades. These statistics include the number of new and reused SLOC in each project and the number of staff hours expended on each project. From these data SEL researchers can compute the average productivity, expressed in DLOC per hour, on any project. As can be seen in Equation 3-2, the productivity calculation for a past project depends both on the effort for that project and also on the value that is assigned as the reuse cost (embedded in the definition of DLOC).

$$\text{Productivity} = \text{DLOC} / \text{Effort} \tag{3-2}$$

To arrive at a first-order estimate for the effort of an upcoming project, one divides the estimated DLOC by the anticipated productivity (DLOC per hour), as shown in Equation 3-3.

$$\text{Effort} = \text{DLOC} / \text{Productivity} \tag{3-3}$$

Figure 3-1 graphs the project productivities for 33 AGSS and simulator projects found in the SEL database. The effort used to calculate these productivities is the total technical staff and technical management effort; it does not include the support hours, such as project management, Technical Publications, secretarial, and librarian support. (Project support hours are tracked for CSC-developed projects, but are usually not tracked for GSFC in-house projects.) In the remainder of this report, all productivities are based on technical management and technical management effort only, unless specified otherwise.

Figure 3-1 contains three data points representing the overall productivities of combined projects. The project labeled as UARS_2 represents the total UARS AGSS, which was developed as two separate efforts, a large CSC project (identified simply as UARSAGSS in the SEL database) and a smaller GSFC inhouse project (identified as ACME). The name SAMPEX_2 similarly denotes the total SAMPEX AGSS, which was composed of a large CSC project (identified simply as SAMPEX) and a smaller GSFC inhouse project (identified as SAMPEXTP). The EUVE AGSS was developed as a *single* project, and the EUVEAGSS account in the SEL database includes all hours spent on this AGSS. In recording the SLOC number in the EUVEAGSS account, however, the SEL database did not include the ACME SLOC, all of which was borrowed from the ACME project and reused verbatim in the EUVE AGSS. The overall productivity for the EUVE AGSS is given by the EUVE_2 data point and represents the sum of the ACME DLOC and the EUVEAGSS DLOC, both divided by the EUVEAGSS effort.

Figure 3-1 shows significant variability in the productivities for the projects. In particular, two projects, SAMPEXTS and COBSIM, stand out with significantly higher productivities than similar projects.

The SAMPEX telemetry simulator project (SAMPEXTS) had a productivity of over 8 DLOC per hour, much higher than EUVETELS, the preceding Ada telemetry simulator. Both SAMPEXTS and EUVETELS benefited from a very high level of verbatim code reuse, but the stability of the libraries from which they borrowed was not equivalent. EUVETELS borrowed much of its code from UARSTELS, but the development cycles of these two projects largely overlapped. Thus, EUVETELS was sometimes adversely impacted by design and coding changes made by the UARSTELS project. On the other hand, the development cycles of SAMPEXTS and EUVETELS overlapped very little. As a result, SAMPEXTS was able to efficiently borrow code from a more stable code library. In addition SAMPEXTS piloted a streamlined development process, combining some documents and combining the PDR with the CDR. SAMPEXTS also used a lower staffing level and followed a shorter delivery schedule than EUVETELS.

It is likely that as a result of all these advantages, the reuse cost on SAMPEXTS was actually less than the 30-percent standard attributed to Ada projects. Using a lower reuse cost to compute the DLOC for SAMPEXTS would result in a lower productivity value. For example, a 20-percent reuse cost would lead to a productivity of 5.9 DLOC per hour; a 10-percent reuse cost would result in a productivity of 3.6 DLOC per hour. These productivity numbers are presented only as suggestions. More data are needed before revising the Ada reuse cost. In all subsequent analysis, the 30-percent reuse cost is assumed for Ada projects.

The next Ada telemetry simulator completed, POWITS, had a lower productivity than both EUVETELS and SAMPEXTS. POWITS also had a much lower reuse percentage than SAMPEXTS, 69 percent versus 95 percent. In particular, the percentage of verbatim reuse was much lower, 39 percent versus 85 percent. Part of the difficulty with POWITS was that this project was trying to model a spin-stabilized satellite by reusing the generic telemetry simulator architecture that was designed for three-axis-stabilized satellites.

The COBSIM project, the other project in Figure 3-1 with a very high productivity, was the last FORTRAN telemetry simulator developed before the switch to Ada. It was an inhouse GSFC project. In addition to having an unusually high productivity, the software also grew significantly relative to both of the two preceding FORTRAN telemetry simulators and relative to COBSIM's own CDR estimate. Measured in DLOC, COBSIM was 145 percent the size of GROSIM and 320 percent the size of DESIM. The final COBSIM size was 330 percent of its CDR DLOC estimate. The reasons for the significant growth and high productivity remain unresolved.

3.2 Accuracy of Models for Total Effort

This section derives recommended productivity values and then validates the accuracy of Equation 3-3 for estimating the technical and management effort on an FDD software development project. Adjustments are then made to the recommended productivities to take into account the addition of support staff effort. Section 3.2.1 computes the estimated effort from the end-of-project DLOC value. Section 3.2.2 computes the estimated effort from the CDR estimate for DLOC and then applies a standard growth factor to this effort estimate.

As stated above, Equation 3-3 gives a first-order estimate for the effort of a software development project. Software cost estimation methods currently used in the FDD advocate the use of additional multipliers to adjust such effort estimates or the productivities on which they are based. The multipliers advocated reflect estimates for such contributing factors as team experience or problem complexity. The current study examined data from the SEFs that are completed at the end of each FDD project. The SEF data provide estimates for many factors such as problem complexity and team experience. The resulting analysis showed that the SEF data in the SEL database provide no demonstrable evidence that inclusion of estimates for such factors as problem complexity or team experience will significantly improve a manager's estimate of project effort. When making estimates for project effort, managers are still encouraged to include such factors as problem complexity or team experience *based on their own personal experience*, but the database of experience represented by the SEF data in the SEL database provides no guidelines. Section 4 includes a complete discussion of this topic.

3.2.1 Model Predictions Based on Final Project Statistics

In recent years, FDD AGSSs have continued to be written in FORTRAN. FDD simulators, however, are now written in Ada rather than FORTRAN. In order to determine the optimum productivities for modeling FORTRAN and Ada FDD projects, therefore, this study has concentrated on the recent FORTRAN AGSSs and most of the Ada simulators, disregarding the earlier FORTRAN simulators. The SAMPEXTS project was excluded from the Ada productivity analysis because it piloted a streamlined development process for which too few data are available at this time. The POWITS project was also deemed an outlier and was excluded. Its productivity was significantly lower than the other Ada projects, mainly because of the problems encountered in modeling a spinning spacecraft.

To determine the best FORTRAN productivity to use in Equation 3-3, the study focused on the eight most recent AGSSs, ERBS through SAMPEX_2. As can be seen in Figure 3-1, the productivities of these eight projects (numbers 11 through 18) varied from approximately 3 to 5 DLOC per technical staff and technical management hour. Given this wide variation, it is best to choose a model productivity that is closer to the lower bound than to the mean productivity. This choice reduces the likelihood of significantly underestimating the effort of a future project. For planning purposes, therefore, a productivity value of 3.5 DLOC per technical and management hour is recommended (see Equation 3-3A).

$$\text{FORTRAN Effort} = \text{DLOC}/(3.5\ \text{DLOC/hour}) \qquad (3\text{-}3A)$$

Project effort estimates were computed for the eight projects, using 3.5 DLOC per hour and the end-of-project DLOC value. Figure 3-2 plots the percent deviations from actual effort for these effort estimates. The RMS percent deviation is 24 percent. As can be seen, the estimates are particularly good for the middle four AGSSs, GROAGSS through UARS_2. The two recent high-reuse AGSSs, EUVE_2 and SAMPEX_2, do not fit the model nearly as well.

The Ada productivities (excluding outliers SAMPEXTS and POWITS) were more uniform than the FORTRAN productivities. Consequently, the model productivity can be chosen closer to the mean without increasing the risk of significantly underestimating the effort of a future project. A productivity value of 5.0 DLOC per technical and management hour is recommended (see Equation 3-3B). Figure 3-3 plots the percent deviations for these effort estimates. The RMS percent deviation is 7 percent.

$$\text{Ada Effort} = \text{DLOC} / (5.0\ \text{DLOC/hour}) \qquad (3\text{-}3B)$$

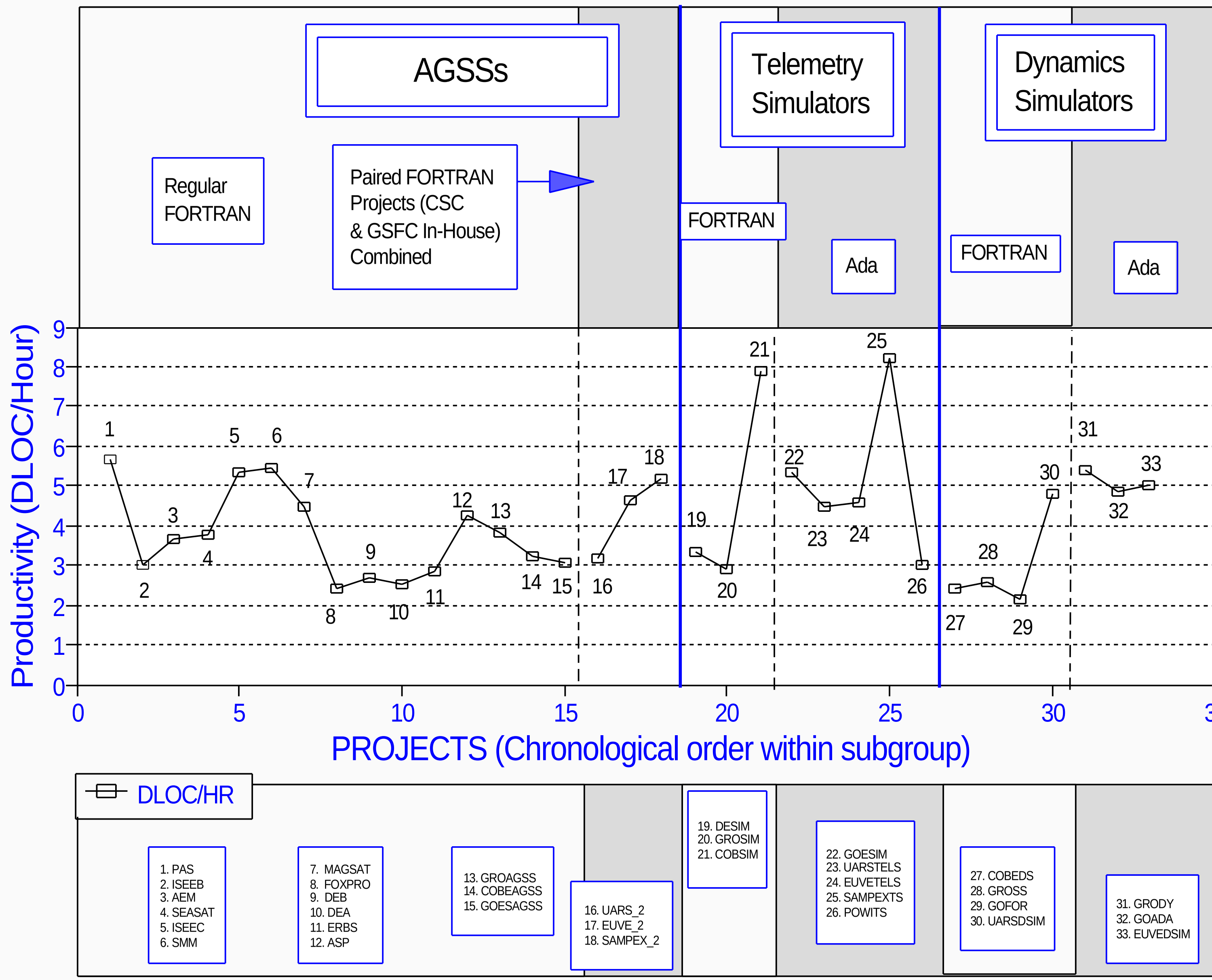


Figure 3-1. Productivity for AGSSs and Simulators (Based on Technical & Mgmt Hours)

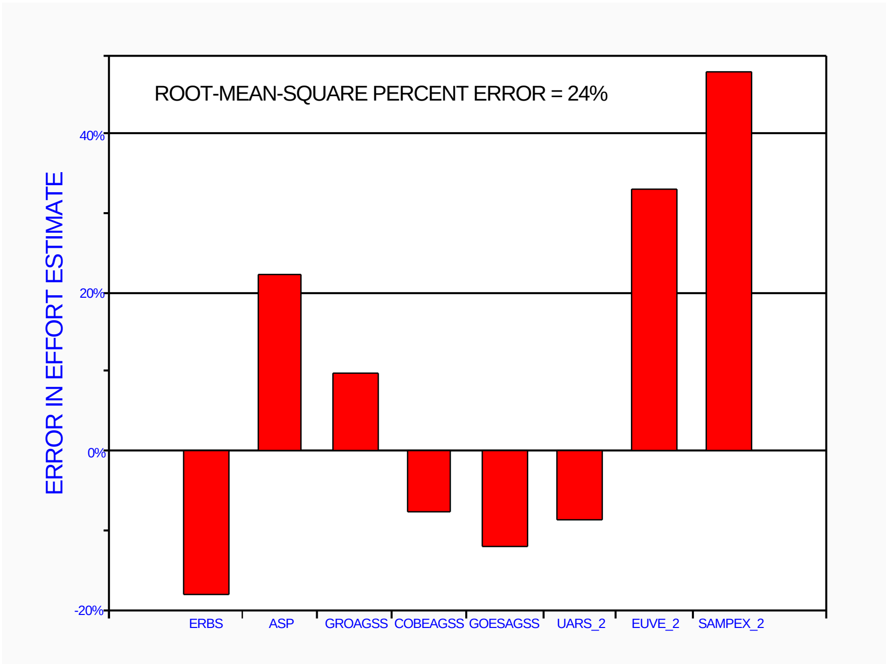


Figure 3-2. Accuracy of FORTRAN Effort Estimation Model

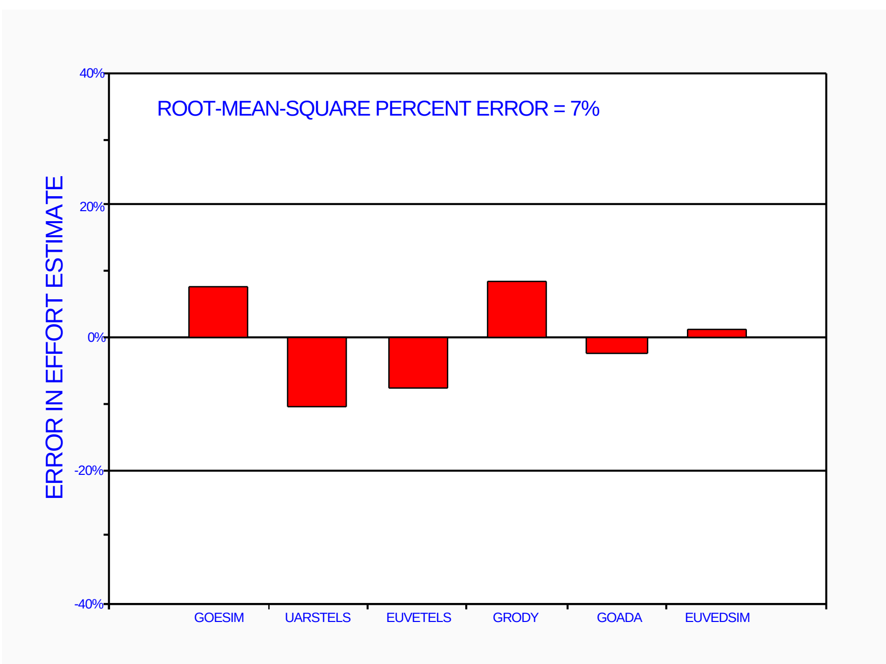


Figure 3-3. Accuracy of Ada Effort Estimation Model

Both GSFC in-house projects and CSC-developed projects track technical staff and technical management hours, but only CSC-developed projects track support hours (project management, librarians, Technical Publications personnel, and secretaries). In order to compare GSFC inhouse projects with CSC-developed projects, therefore, it is necessary to have a model based on technical effort.

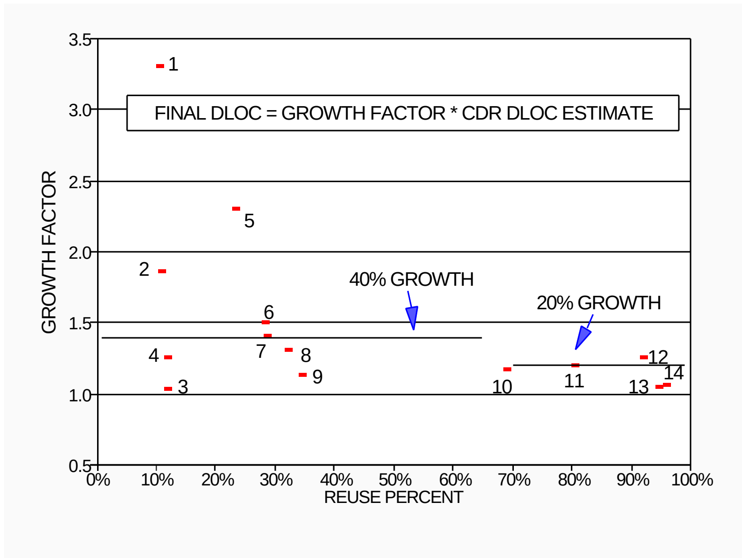
Since CSC-developed projects are planned with total cost in mind, however, it is also necessary to have a model based on total effort, including support hours. For the 21 CSC-developed projects from ERBS through SAMPEX the support hours add approximately 10 percent on top of the hours computed from technical effort alone. (For these 21 projects the mean value of support hours divided by technical hours is 11.5 percent, with a standard deviation of 5 percent.) The appropriate model productivities are shown below.

	Productivity (DLOC per hour)	
Type of Effort	**FORTRAN**	**Ada**
Technical and Management	3.5	5.0
Technical, Management, and Support	3.2	4.5

3.2.2 Model Predictions Based on CDR Estimates

The effort model presented in Section 3.2.1 describes how the end-of-project DLOC value is related to the end-of-project effort value. During the development of a project, however, project managers must rely on estimates of DLOC to predict total project effort. Requirements changes, TBDs, and in some cases the impossibility of reusing code as planned, typically cause these DLOC estimates to grow during the life of a project. Because of this well-known tendency, project managers usually apply a growth factor to their DLOC estimates to determine the effort that will be required for the complete project. This section proposes two values for average post-CDR growth factor, based on a project's amount of code reuse. It then validates the effort estimation model using CDR DLOC estimates along with these growth factors. Section 6 presents a more complete discussion of planning models and their relationship to models that are based on end-of-project data.

Figure 3-4 presents a project-by-project scatter plot of DLOC growth versus code reuse. The projects represented are the post-ERBS projects for which CDR DLOC estimates were available in the database. The *y*-axis plots the DLOC growth factor, which is the end-of-project DLOC divided by the CDR estimate. The *x*-axis plots the percent of code reuse attained by each project. As can be seen, the high-reuse projects (70 percent or more reuse) tended to show lower DLOC growth than did the low-reuse projects. Based on these data, this study recommends planning for 20-percent growth in DLOC on high-reuse projects and 40-percent growth on lower reuse projects. Equation 3-3C presents the revised effort estimation equation based on the DLOC estimated at CDR plus the DLOC due to growth.



1. COBSIM
2. UARSAGSS
3. COBEAGSS
4. GOESAGSS
5. UARSDSIM
6. GOADA
7. GOESIM
8. GOFOR
9. UARSTELS
10. POWITS
11. EUVEAGSS
12. SAMPEX
13. SAMPEXTS
14. EUVETELS

Figure 3-4. DLOC Growth Factors: Actual DLOC Divided by CDR Estimate

$$\text{Effort} = (\text{DLOC} / \text{Productivity}) * \text{Growth Factor} \quad (3\text{-}3C)$$

Code Reuse Level	Growth Factor
Less than 70%	1.4
70% or more	1.2

3.3 Distribution of Effort by Life-Cycle Phase

To staff a software project properly and to plan milestones accurately, a manager needs to know how much effort will be required in each of the life-cycle phases. This study examined three of these phases (design phase, code phase, and combined system test and acceptance test phase).

Historically, the SEL has relied on a predictive model that assumes that each project will spend the same fixed percentage of the total project effort in a given life-cycle phase, regardless of how much code is reused. Table 3-1 lists the phase effort distributions for eight recent FORTRAN AGSSs and eight recent Ada simulators. FORTRAN simulators were excluded, since all FDD simulators are now written in Ada.

Table 3-1. Effort Distributed by Life-Cycle Phase

Project	Design	Code	Test
AGSSs			
ERBS	22%	51%	27%
ASP	23%	43%	34%
GROAGSS [1]	21%	52%	27%
COBEAGSS	23%	38%	39%
GOESAGSS	19%	54%	27%
UARS_2	21%	50%	30%
EUVE_2 [2]	14%	48%	38%
SAMPEX_2 [2]	26%	36%	38%
SIMULATORS			
GRODY[1`]	16%	58%	26%
GOADA	28%	42%	31%
GOESIM	28%	44%	27%
UARSTELS	25%	39%	36%
EUVEDSIM [1, 2]	24%	50%	26%
EUVETELS [2]	26%	41%	33%
POWITS [2]	14%	47%	39%
SAMPEXTS [2]	48%	18%	34%

[1] Excluded from analysis of phase effort.
[2] High reuse.

Several projects from this list were excluded from further analysis of phase effort. The GROAGSS project was excluded because its development life-cycle was significantly distorted due to the lapse in Space Shuttle flights following the Challenger disaster. The EUVEDSIM project was excluded because this dynamics simulator project was terminated early and had no acceptance test phase. The GRODY project, another dynamics simulator, was the first Ada development project in the FDD. Due to its experimental purpose, GRODY's phase effort is much different from the Ada projects that followed it. Consequently GRODY was also excluded from further calculations of phase effort.

Among the remaining 13 projects there are 5 projects with much higher reuse than the other 8 projects. These high-reuse projects—EUVE_2, SAMPEX_2, EUVETELS, POWITS, and SAMPEXTS—all have about 70-percent reuse or more; the other eight projects have only 10 percent to 35 percent reuse. As can be seen from Table 3-1, there is much more variability in the phase effort distributions among the five high-reuse projects than among the eight low-reuse projects.

Among the eight moderate to low-reuse projects, there are five FORTRAN AGSSs and three Ada simulators. Table 3-2 presents three phase effort models for moderate to low-reuse projects: one model for the FORTRAN projects, one model for the Ada projects, and one overall model for all eight projects. For each phase the effort percentage was arrived at by computing the mean percentages for the projects in the subset. The standard deviations are also shown. As can be seen, the AGSSs spend relatively less effort on design and more effort on coding than do the Ada simulators. The moderate standard deviations for the eight-project model, however, show that there is still a good deal of agreement between the two types of projects.

Table 3-2. Effort-by-Phase Models for Moderate to Low-Reuse Projects

	5 FORTRAN AGSSs		3 Ada Simulators		All 8 Projects	
Phase	**Effort Percentage**	**Std. Dev.**	**Effort Percentage**	**Std. Dev.**	**Effort Percentage**	**Std. Dev.**
Design	21%	(2%)	27%	(2%)	24%	(3%)
Code	47%	(7%)	42%	(2%)	45%	(6%)
Test	31%	(5%)	31%	(4%)	31%	(5%)

Table 3-3 presents a preliminary phase effort model for high-reuse projects. It is based on the five high-reuse projects mentioned above, two FORTRAN AGSSs and three Ada simulators. The larger standard deviations for the high-reuse model reflect the greater variability in effort distributions for high-reuse projects to date. This will be revisited when there are more data.

Table 3-3. Preliminary Effort-by-Phase Model for High-Reuse Projects

	5 High-Reuse Projects	
Phase	**Effort Percentage**	**Std. Dev.**
Design	26%	(14%)
Code	38%	(12%)
Test	36	(3%)

3.4 Distribution of Effort by Software Development Activity

Table 3-4 lists the effort distributions by software development activity for the same eight recent FORTRAN AGSSs and eight recent Ada simulators. The activities are grouped as shown in Table 2-5. Again the outliers GROAGSS, GRODY, and EUVEDSIM were excluded when developing an effort distribution model for moderate to low-reuse and high-reuse projects. Table 3-5 presents three activity effort models for moderate to low-reuse projects: one model based on only FORTRAN AGSSs, one model based on only Ada simulators, and one overall model based on both FORTRAN AGSSs and Ada simulators. Table 3-6 presents a preliminary activity effort model for high-reuse projects. It is based on the same five high-reuse projects as used in the phase effort model in the preceding section, two FORTRAN AGSSs and three Ada simulators.

Table 3-4. Effort Distributed by Software Development Activity

Project	Design	Code	Test	Other
AGSSs				
ERBS	18%	29%	17%	36%
ASP	15%	21%	24%	40%
GROAGSS[1]	20%	29%	21%	30%
COBEAGSS	24%	22%	28%	26%
GOESAGSS	25%	32%	25%	18%
UARS_2	24%	28%	29%	18%
EUVE_2 [2]	22%	25%	31%	22%
SAMPEX_2 [2]	22%	9%	36%	33%
SIMULATORS				
GRODY [1]	26%	34%	15%	25%
GOADA	19%	28%	24%	29%
GOESIM	19%	23%	24%	34%
UARSTELS	19%	28%	33%	20%
EUVEDSIM [1,2]	21%	30%	21%	28%
EUVETELS [2]	15%	17%	26%	42%
POWITS [2]	9%	19%	41%	31%
SAMPEXTS [2]	17%	17%	27%	40%

[1] Excluded from analysis of activity effort.

[2] High reuse.

Table 3-5. Effort-by-Activity Models for Moderate to Low Reuse Projects

	5 FORTRAN AGSSs		3 Ada Simulators		All 8 Projects	
Activity	**Effort Percentage**	**Std. Dev.**	**Effort Percentage**	**Std. Dev.**	**Effort Percentage**	**Std. Dev.**
Design	21%	(5%)	19%	(0%)	21%	(4%)
Code	26%	(5%)	26%	(3%)	26%	(4%)
Test	24%	(5%)	27%	(6%)	25%	(5%)
Other	28%	(10%)	28%	(7%)	28%	(9%)

Table 3-6. Preliminary Effort-by-Activity Models for High Reuse Projects

	5 High-Reuse Projects	
Activity	Effort Percentage	Std. Dev.
Design	17%	(5%)
Code	17%	(6%)
Test	32%	(6%)
Other	34%	(8%)

Section 4. Methods for Adjusting Total Effort

4.1 Overview

Software cost estimation methods frequently attempt to improve project effort estimates by factoring in the effects of project-dependent influences such as problem complexity and development team experience. Some estimation models attempt to build linear equations that include as independent variables the estimates for factors such as these. Other estimation models attempt to adjust an initial effort (or productivity) estimate by applying several multiplicative factors, each of which is a function of a software development influence, such as problem complexity or team experience. In the FDD, two models of the latter type have been advocated over the years. The estimation model currently used in the FDD advocates applying productivity multipliers to adjust the productivity estimate shown in Equation 3-3. A previous estimation model recommended by the FDD advocated applying similar multiplicative factors directly to the effort estimate itself. Appendix A, which presents a matrix of costing and scheduling formulas that have been recommended in the FDD over the last 14 years, displays both these FDD models.

The current study sought to use empirical data in the SEL database to validate the usefulness of including such software development influences in effort estimates. The study sought to determine the following:

- Does the inclusion of such factors improve the accuracy of estimates for project effort or productivity?
- Which factors consistently provide the greatest improvement in estimation accuracy?

Two different approaches were followed, both using project-specific data from FDD SEFs to evaluate these effects. The first approach sought to derive a relationship between one or more SEF parameters and the final project productivity. By iterative optimization methods, the weights of the SEF parameters were adjusted until the estimated productivity came closest to the end-of-project productivity. Several different subsets of projects were evaluated, including both FORTRAN and Ada projects.

The second approach focused directly on project effort and relied on traditional linear regression methods. This approach derived linear equations for effort, in which DLOC and the SEF parameters served as the independent variables. Two subsets of projects were evaluated, one containing 24 older FORTRAN projects and one containing 15 recent FORTRAN projects.

Of the 35 SEF parameters tested, a handful seemed to improve the accuracy of the final predictions for either productivity or effort. Between different subsets of projects, however, there was no consistency with regard to which SEF parameters were helpful.

As a further test, the project-specific SEF data were replaced with random numbers and the equations for productivity and effort were rederived. The new equations (and the random SEF data on which they were based) also resulted in improved predictions for some SEF parameters. The number and degree of improvements resulting from random data were comparable to that achieved with the actual SEF data.

This study concludes that the SEF data provide no evidence of a causal relationship between SEF-type parameters and either effort or productivity. This conclusion follows from two observations. First, the phenomenon of interest lacks continuity from one project subset to another and from one timeframe to another. Second, the 35 sets of random integers demonstrate a degree of improvement that is comparable to that observed with the 35 sets of actual SEF parameter measurements.

One should not infer from the preceding statements that there is *no* connection between software development effort and the influences that the SEF attempts to measure. On the contrary, it is very likely that there are some cases in which influences such as team experience and problem complexity will have a measurable effect on project effort. For example, on a small project with only one or two programmers, team experience could be a crucial factor in determining project effort.

The SEF data in the SEL database, however, provide no demonstrable evidence that inclusion of estimates for factors such as problem complexity, team experience, schedule constraints, or requirements stability will significantly improve a manager's estimate of project effort. The absence of such a measurable effect may be due to the fact that these typical FDD projects are fairly homogeneous with regard to these influences . The effect on effort of the slight variations in these influences may be overwhelmed by other influences not measured by the SEF. Alternatively, the influences of these parameters may be invisible because the SEF does not consistently measure them. It should be noted that it was not the purpose of this study to determine whether or not to continue collecting SEF data, but rather to make a recommendation as to whether or not to include such parameters in the equation for estimating software development effort.

When making estimates for project effort, managers are still encouraged to include such factors as problem complexity or team experience *based on their own personal experience*, but the database of experience represented by the SEF data in the SEL database provides no guidelines.

The remainder of this chapter is organized as follows. Section 4.2 describes the scope of the analysis and lists the projects studied. Section 4.3 describes the criteria used to evaluate the success of the models. Section 4.4 uses iterative optimization techniques to analyze the usefulness of productivity multipliers. Section 4.5 uses traditional linear regression methods to analyze the usefulness of linear effort models that include subjective factors. Section 4.6 presents conclusions.

4.2 Scope of the Analysis

In the past 14 years various models have been proposed and used in the FDD to enhance predictions of the effort required to develop software. These models fall into two formula types. Although these formulas have different appearances, they are functionally equivalent, and both are consistent with the Constructive Cost Model (COCOMO) (see

Reference 4). Page A-3 and pages A-5 through A-7 of Appendix A give examples of the two formulas mentioned above, which are described more fully in the following paragraphs.

The first formula starts with an equation for the estimated effort, expressed as a function of software size. A multiplicative factor preceding the software size implicitly contains a first-order estimate for the average software development productivity. Following this first factor, one can then insert additional multiplicative factors that reflect the effect on productivity of other influences. This method was advocated in *An Approach to Software Cost Estimation* (Reference 5) and *Manager's Handbook for Software Development, Revision 1* (Reference 6).

The second formula is exemplified by the SEAS Basic Estimation Method (BEMS), used in the *SEAS System Development Methodology (SSDM)* Standard and Procedure No. 1102: "Software Development Estimation" (Reference 7). The BEMS formula begins with an explicit *base development productivity*. It then multiplies this average productivity by several optional factors, each modeling the effect of one influence on productivity, until it arrives at the *adjusted productivity* estimate. Dividing the software size estimate by this adjusted productivity yields the BEMS *adjusted estimated effort*.

For the current study, the assessments of various software development factors such as schedule constraints and requirements stability were taken from the SEF data found in the SEL database. At the completion of each software development project in the FDD, an SEF is completed. (There are, however, no firm guidelines as to which personnel take part in completing the SEF.) This form rates the project on 35 characteristics of the development task, the personnel, the technical management, the process, the development environment, and the final product. Each characteristic is rated on a 1-to-5 scale. Table 4-1 lists these 35 SEF parameters. A sample SEF questionnaire is included as Appendix B.

To test the validity of productivity multipliers, the study focused on 33 projects: 18 AGSS projects (all written in FORTRAN), 8 telemetry simulator projects (three in FORTRAN and five in Ada), and 7 dynamics simulator projects (four in FORTRAN and three in Ada). These 33 projects, listed in Table 4-2, include all the AGSS projects and simulator projects whose data have been completed and verified and for which SEF data were available.

To evaluate the utility of a linear equation composed of software development parameters, two project sets were used. One set consisted of 24 older FORTRAN projects. The other set consisted of 15 recent FORTRAN projects. Table 4-3 lists these two sets of projects.

As mentioned previously, the UARS_2 and SAMPEX_2 projects each comprised *two* development projects. For the analysis in this study, the SEF values of the related subprojects were weighted by the relative efforts of the subprojects and then averaged to obtain the SEF value for the project as a whole. This process resulted in noninteger SEF values for UARS_2 and SAMPEX_2. This step was not necessary for the EUVE AGSS, which was conducted as a single development project. The EUVE_2 data differs from EUVEAGSS only in that EUVE_2 includes the ACME SLOC, all of which was reused verbatim in the EUVE AGSS.

Table 4-1. SEF Parameters

Problem Characteristics:	
PM01	Problem Complexity
PM02	Schedule Constraints
PM03	Requirements Stability
PM04	Requirements Specifications Quality
PM05	Documentation Extensiveness
PM06	Rigor of Review Requirements
Technical Staff Characteristics:	
ST07	Development Team Quality
ST08	Team Experience With Application
ST09	Team Experience With Environment
ST10	Team Stability
Technical Management Characteristics:	
TM11	Management Performance
TM12	Management Experience With Application
TM13	Management Stability
TM14	Degree of Disciplined Project Planning
TM15	Fidelity to Project Plan
Process Characteristics:	
PC16	Degree Modern Programming Practices Used
PC17	Disciplined Procedures for Spec. Mods., Req't Specs, and Interface Agreements
PC18	Used Well-Defined Req't Analysis Methodology
PC19	Used Well-Defined Design Methodology
PC20	Used Well-Defined Testing Methodology
PC21	(not applicable)
PC22	Fidelity to Test Plans
PC23	Used Well-Defined & Disciplined QA Procedures
PC24	Used Well-Defined & Disciplined CM Procedures
Environment Characteristics:	
EN25	Team Access to Development System
EN26	Ratio of Programmers to Terminals
EN27	Constrained by Main Memory or DA Storage
EN28	System Response Time
EN29	Stability of Hardware & System Support SW
EN30	Effectiveness of Software Tools
Product Characteristics:	
PT31	Software Meets Specified Requirements
PT32	Quality of Delivered Software
PT33	Quality of Design in Delivered Software
PT34	Quality of Delivered System Documentation
PT35	Software Delivered On Time
PT36	Relative Ease of Acceptance Testing

Table 4-2. Projects Used To Test Productivity Multipliers

AGSSs	Telemetry Simulators
PAS	DESIM
ISEEB	GROSIM
AEM	COBSIM
SEASAT	GOESIM
ISEEC	UARSTELS
SMM	EUVETELS
MAGSAT	POWITS
FOXPRO	SAMPEXTS
DEB	
DEA	**Dynamics Simulators**
ERBS	COBEDS
ASP	GROSS
GROAGSS	GOFOR
COBEAGSS	UARSDSIM
GOESAGSS	GRODY
UARS_2	GOADA
EUVE_2	EUVEDSIM
SAMPEX_2	

Table 4-3. Projects Used To Test Correlations Between Effort and SEF Parameters

24 Older FORTRAN Projects

AEM	FOXPRO	ASP
ISEEB	DEA	DSPLBLDR
PAS	DEB	COBSIM
ISEEC	DESIM	GROSIM
SEASAT	GSOC	GOFOR
SMM	DEDET	UARSDSIM
MAGSAT	GROSS	ACME
FOXPP	COBEDS	BBXRT

15 Recent FORTRAN Projects

AGSSs:	Simulators
ERBS	DESIM
ASP	GROSIM
GROAGSS	COBSIM
COBEAGSS	COBEDS
GOESAGSS	GROSS
UARS_2	GOFOR
EUVE_2	UARSDSIM
SAMPEX_2	

4.3 Methods Used to Evaluate Success

Adding SEF parameters to the estimation equation either as productivity multipliers or as additional linear parameters alongside DLOC may result in improved estimates. To evaluate the utility of including the SEF parameters requires determining how much improvement results in the accuracy of the estimate when the SEF parameters are included. This involves measuring the closeness of fit between the derived equation (based on one or more SEF parameters) and the final project productivities or efforts given by the data points. This closeness of fit could be measured in terms of the RMS percent deviation or, in the case of linear regression results, the R-squared value. One must be careful in such comparisons. The R-squared values, for example, should not be used for comparison between equations with differing degrees of freedom.

In assessing the apparent improvement in fit, one should also consider what portion of the improvement is due to the purely mathematical effect of adding extra dimensions to the equation. Each extra dimension adds another independent parameter to the equation. As the number of independent parameters rises, it becomes easier to make the equation more closely fit the fixed number of data points. In the extreme case where there are more independent parameters than the number of projects in one's dataset, it is usually possible to make the derived equation precisely fit the data points. Such results are meaningless, however, because of the scarcity of data points relative to the number of dimensions. For example, one would hesitate to estimate with a *two*-dimensional linear equation that was derived from only *two* experimental data points.

It should also be noted that estimating equations derived from end-of-project SEF data will be less accurate when applied early in the life cycle. This is because such estimates will be based on *early* assessments of the SEF parameters, which are inherently less accurate than the end-of-project SEF assessments used in this study.

4.4 Deriving Productivity Multipliers with Optimization Procedures

The first approach, deriving productivity multipliers with optimization procedures, began with a base productivity estimate and then attempted to improve it by including project-specific knowledge about one or more software development influences, until the estimated productivity came as close as possible to the final project productivity. In order to achieve the best fit between the initial model and the given dataset of 33 projects, the base productivity for FORTRAN was chosen to be 3.83 DLOC per hour, the mean productivity for the 25 FORTRAN projects in the sample. Likewise, the Ada base productivity was chosen to be 5.13 DLOC per hour, the mean productivity for the 8 Ada projects in the sample. (These numbers vary slightly from the moderately conservative productivity planning numbers recommended in Section 3.) Because all values for effort and lines of code were taken directly from the final project statistics, growth factors were not considered.

4.4.1 Effect of Individual SEF Parameters on Productivity Estimate

To investigate the usefulness of individual SEF parameters for predicting the productivity, the following equation was used:

$$\text{(Predicted productivity)} = \text{(Base productivity)}$$

$$\times \{ 1.0 - [(\text{SEF Parameter} - 3.0) \times (\text{Parameter Scale Factor})] \} \quad (4\text{-}1)$$

The part of Equation 4-1 enclosed in braces ({ }) is a function of the SEF parameter. If the SEF parameter is 3 (the middle of the 1-to-5 scale on the form), then the value of this function is just 1.0, and the predicted productivity is the same as the base productivity. If the SEF parameter differs from 3, then the predicted productivity differs from the base productivity. How much the predicted productivity differs depends both on the value of the SEF parameter (which varies from project to project) and on the *parameter scale factor* (which varies from one SEF parameter to another but is independent of project). For each SEF parameter, the parameter scale factor was set to an initial value and then optimized to the best value. For example, an SEF parameter of 4 combined with an initial parameter scale factor of 10 percent would result in a 10-percent *reduction* in the predicted productivity. An SEF parameter of 1 would result in a 20-percent *increase* in the predicted productivity.

Equation 4-1 produces a value for the predicted productivity of a project based on evaluating one SEF parameter. The percent deviation of this prediction from the actual project productivity gives a measure of how effective this productivity model is for *that project*. The RMS percent deviation of the predicted productivities for a subset of projects gives a measure of how effective this model is for *that subset*. The goal of this approach was to find a model that was effective for a subset of projects. Table 4-4 shows the seven subsets of projects that were analyzed. The way to demonstrate the effectiveness of a model for a subset of projects is to find one value of the parameter scale factor—positive or negative—that produces a significant reduction in the RMS percent deviation of the subset's productivities.

Table 4-4. Subsets of Projects Tested

33 AGSSs and Simulators
All 18 AGSSs
23 AGSSs and Simulators (8 recent AGSSs and all 15 simulators)
8 recent AGSSs (ERBS through SAMPEX)
All 15 Simulators
All 8 Telemetry Simulators
All 7 Dynamics Simulators

The tool used to perform these optimizations was part of the personal computer software package known as Quattro Pro, Version 4.0 (see Reference 8). This package contains a tool called the Optimizer, which solves linear and nonlinear problems involving multiple variables and multiple constraints. For the present work the Optimizer was required to adjust the value of the given parameter scale factor until an RMS percent deviation within the specified precision of the minimum value was achieved. The default precision of .0005 was generally used. The Optimizer contains several switches that allow the user to specify various mathematical options, some of which are specified below. A variety of option settings were used at different times.

- Specify the approach used to obtain initial estimates of the basic variables in each iteration. Linear extrapolation from a tangent vector or quadratic extrapolation are the two choices available.

- Specify either forward or central differencing for estimates of partial derivatives.

- Specify the method for computing the search direction, either a quasi-Newton method or conjugate method.

In most cases, the optimization process began with a parameter scale factor of +10 percent, and the Optimizer then varied it until achieving the minimum RMS percent deviation. In order to remove any doubt that the final result of the optimization process might depend on the choice of the initial value, a significant number of trials were repeated beginning with a variety of initial values. In each such case, the choice of the initial value had no effect on the final optimized value.

Table 4-5 lists, for each subset of projects and for each SEF parameter, the parameter scale factor that yields the lowest RMS percent deviation of the predicted productivities versus the actual project productivities. The RMS percent deviation (expressed as a decimal number) is listed in parentheses beneath the value of each parameter scale factor. At the top of the table are listed the RMS percent deviations that result when *no* SEF parameter is used to adjust the mean FORTRAN and Ada productivities. These base RMS values vary from 21 percent to 42 percent, depending on the subset of projects considered. For each subset of projects, a few SEF parameters that provide the most significant improvements in the RMS percent deviation are denoted by boldfaced RMS values. Each row of Table 4-5 represents a productivity model that is based on the influence of a *single* SEF parameter. Models displaying the simultaneous influence of *more than one* SEF parameter are described later.

Often the use of one SEF parameter improves the base RMS percent deviation by only one or two percentage points. Improvements of this magnitude—and sometimes larger—can easily be achieved using random values for SEF parameters and are only valid for the given set of SEF values and productivities; they represent no generally valid relationship.

This conclusion was verified by substituting *random values* for the 35 SEF parameters for each project and then repeating the calculations that led to Table 4-5. This again resulted in 35 parameter scale factors and 35 RMS percent deviations for each of the seven subsets of projects. Since the SEF values used this time were random, none of the 35 resulting rows represents the effect of PM01 or any other real SEF value on the productivity of the subsets. The value of any one of the new 35 RMS percent deviations in a column is thus not important. What is important is the range and distribution of the 35 RMS percent deviations and how these compare to the RMS percent deviations resulting when *no* SEF data are used (the row labeled "None" in Table 4-5) and also to the range and distribution of the RMS percent deviations resulting from real SEF data in Table 4-5.

Table 4-5. Effect of Individual SEF Parameters on Productivity Estimate (1 of 4)

	Parameter Scale Factor (Root-mean-square percent deviation in predicted productivity)[1]						
SEF Parameter	**33 AGSSs & Simulators**	**All 18 AGSSs**	**23 AGSSs & Simulators**	**8 Recent AGSSs**	**15 Simulators**	**8 Telemetry Simulators**	**7 Dynamics Simulators**
None	(0.339)	(0.287)	(0.340)	(0.210)	(0.392)	(0.367)	(0.420)
PM01	0.042 (0.335)	-0.077 (0.273)	0.140 **(0.296)**	0.066 (0.197)	0.173 **(0.329)**	0.094 (0.349)	0.250 **(0.270)**
PM02	0.054 (0.329)	-0.023 (0.285)	0.116 **(0.300)**	0.027 (0.209)	0.128 **(0.334)**	0.100 (0.337)	0.149 (0.325)
PM03	-0.057 (0.329)	-0.033 (0.283)	-0.102 (0.316)	-0.178 **(0.143)**	-0.087 (0.373)	-0.098 (0.327)	-0.050 (0.417)
PM04	-0.034 (0.335)	-0.029 (0.283)	-0.040 (0.338)	-0.013 (0.210)	-0.052 (0.389)	-0.172 **(0.305)**	0.282 (0.361)
PM05	0.061 (0.335)	0.062 (0.279)	0.093 (0.334)	0.104 (0.181)	0.051 (0.392)	0.125 (0.363)	-0.056 (0.419)
PM06	0.099 (0.323)	0.071 (0.272)	0.182 (0.312)	0.131 **(0.167)**	0.256 (0.362)	0.235 (0.350)	0.266 (0.375)

[1]Percent deviation expressed as decimal number, i.e., 0.339 means 33.9%.

Table 4-5. Effect of Individual SEF Parameters on Productivity Estimate (2 of 4)

	Parameter Scale Factor (Root-mean-square percent deviation in predicted productivity)[1]						
SEF Parameter	**33 AGSSs & Simulators**	**All 18 AGSSs**	**23 AGSSs & Simulators**	**8 Recent AGSSs**	**15 Simulators**	**8 Telemetry Simulators**	**7 Dynamics Simulators**
None	(0.339)	(0.287)	(0.340)	(0.210)	(0.392)	(0.367)	(0.420)
ST07	0.098 (0.316)	0.070 (0.279)	0.096 (0.313)	0.035 (0.207)	0.111 (0.353)	0.104 (0.316)	0.129 (0.390)
ST08	-0.0032 (0.339)	0.072 (0.274)	-0.073 (0.328)	-0.087 (0.197)	-0.070 (0.380)	-0.030 (0.364)	-0.168 (0.375)
ST09	0.071 (0.326)	0.127 **(0.246)**	0.032 (0.338)	0.073 (0.203)	0.026 (0.390)	-0.049 (0.360)	0.109 (0.389)
ST10	0.094 (0.318)	0.120 **(0.248)**	0.049 (0.335)	-0.0085 (0.210)	0.066 (0.383)	0.036 (0.364)	0.096 (0.400)
TM11	0.078 (0.331)	0.0033 (0.287)	0.131 (0.314)	0.098 (0.191)	0.144 (0.362)	0.140 (0.343)	0.147 (0.384)
TM12	0.097 (0.326)	0.029 (0.286)	0.131 (0.308)	0.086 (0.181)	0.168 (0.354)	0.101 (0.351)	0.256 (0.338)
TM13	0.027 (0.337)	0.0057 (0.287)	0.046 (0.335)	0.022 (0.209)	0.053 (0.386)	0.013 (0.366)	0.110 (0.395)
TM14	0.047 (0.334)	-0.017 (0.286)	0.131 (0.307)	0.047 (0.198)	0.280 **(0.315)**	0.225 **(0.313)**	0.348 (0.308)
TM15	0.057 (0.332)	0.011 (0.287)	0.131 (0.312)	0.070 (0.194)	0.181 (0.353)	0.178 **(0.312)**	0.187 (0.395)

[1]Percent deviation expressed as decimal number, i.e., 0.339 means 33.9%.

Table 4-5. Effect of Individual SEF Parameters on Productivity Estimate (3 of 4)

	Parameter Scale Factor (Root-mean-square percent deviation in predicted productivity)[1]						
SEF Parameter	**33 AGSSs & Simulators**	**All 18 AGSSs**	**23 AGSSs & Simulators**	**8 Recent AGSSs**	**15 Simulators**	**8 Telemetry Simulators**	**7 Dynamics Simulators**
None	(0.339)	(0.287)	(0.340)	(0.210)	(0.392)	(0.367)	(0.420)
PC16	0.047 (0.333)	-0.055 (0.279)	0.106 (0.308)	0.0063 (0.210)	0.130 (0.339)	0.068 (0.358)	0.160 (0.302)
PC17	-0.028 (0.334)	-0.039 (0.268)	0.034 (0.337)	0.062 (0.196)	0.022 (0.391)	-0.014 (0.366)	0.084 (0.408)
PC18	-0.039 (0.334)	-0.059 (0.267)	0.039 (0.338)	0.045 (0.206)	0.036 (0.391)	-0.098 (0.354)	0.168 (0.380)
PC19	0.090 (0.322)	0.059 (0.278)	0.124 (0.314)	0.089 (0.190)	0.140 (0.362)	0.083 (0.355)	0.214 (0.355)
PC20	0.058 (0.330)	-0.018 (0.286)	0.107 (0.307)	0.056 (0.199)	0.127 (0.347)	0.058 (0.360)	0.166 (0.314)
PC22	0.015 (0.338)	-0.086 (0.266)	0.101 (0.314)	0.049 (0.202)	0.125 (0.355)	0.122 (0.340)	0.127 (0.372)
PC23	0.065 (0.332)	0.0 (0.287)	0.125 (0.319)	0.0084 (0.210)	0.212 (0.345)	0.189 **(0.313)**	0.266 (0.375)
PC24	0.032 (0.337)	-0.047 (0.281)	0.092 (0.318)	0.045 (0.203)	0.114 (0.362)	0.109 (0.334)	0.120 (0.392)
EN25	0.130 **(0.310)**	-0.151 (0.279)	0.162 **(0.279)**	2.021 (0.198)	0.161 **(0.310)**	0.107 (0.334)	0.212 **(0.256)**
EN26	0.058 (0.333)	-0.140 (0.258)	0.167 **(0.297)**	-0.107 (0.201)	0.205 **(0.313)**	0.099 (0.350)	0.310 **(0.197)**
EN27	-0.072 (0.335)	-0.080 (0.282)	-0.016 (0.340)	0.118 (0.198)	-0.063 (0.390)	0.180 (0.360)	-0.101 (0.407)
EN28	0.015 (0.339)	-0.057 (0.284)	0.037 (0.338)	-0.026 (0.210)	0.042 (0.388)	-0.129 (0.335)	0.156 (0.346)
EN29	-0.018 (0.338)	-0.062 (0.278)	0.033 (0.338)	0.031 (0.208)	0.034 (0.390)	-0.230 **(0.267)**	0.273 **(0.254)**
EN30	-0.026 (0.335)	-0.040 (0.268)	0.065 (0.332)	0.167 **(0.160)**	0.042 (0.389)	-0.014 (0.366)	0.183 (0.378)

[1]Percent deviation expressed as decimal number, i.e., 0.339 means 33.9%.

Table 4-5. Effect of Individual SEF Parameters on Productivity Estimate (4 of 4)

SEF Parameter	Parameter Scale Factor (Root-mean-square percent deviation in predicted productivity)[1] 33 AGSSs & Simulators	All 18 AGSSs	23 AGSSs & Simulators	8 Recent AGSSs	15 Simulators	8 Telemetry Simulators	7 Dynamics Simulators
None	(0.339)	(0.287)	(0.340)	(0.210)	(0.392)	(0.367)	(0.420)
PT31	0.093 **(0.298)**	0.058 (0.269)	0.097 **(0.295)**	0.029 (0.204)	0.139 **(0.312)**	0.089 (0.334)	0.194 **(0.255)**
PT32	0.073 (0.320)	0.024 (0.285)	0.086 (0.311)	0.015 (0.209)	0.127 (0.337)	0.041 (0.361)	0.216 **(0.232)**
PT33	0.023 (0.338)	-0.069 (0.277)	0.089 (0.326)	-0.013 (0.210)	0.117 (0.367)	-0.046 (0.362)	0.306 **(0.213)**
PT34	0.078 (0.320)	0.066 (0.271)	0.073 (0.326)	0.017 (0.210)	0.092 (0.370)	0.068 (0.357)	0.107 (0.382)
PT35	0.0030 (0.339)	-0.053 (0.281)	0.084 (0.328)	0.118 **(0.174)**	0.070 (0.384)	0.041 (0.363)	0.140 (0.400)
PT36	-0.036 (0.336)	0.0028 (0.287)	-0.063 (0.332)	0.025 (0.209)	-0.086 (0.376)	-0.121 **(0.312)**	0.036 (0.418)

[1]Percent deviation expressed as decimal number, i.e., 0.339 means 33.9%.

Therefore, rather than print all 35 rows of parameter scale factors and RMS percent deviations resulting from random SEF values, only three rows summarizing the 35 RMS percent deviations are presented. These contain the maximum, mean, and minimum RMS percent deviations for each set of 35 RMS percent deviations and are presented in Table 4-6. The mean improvement in the RMS percent deviation was one percentage point each for the first four subsets (the 33 AGSSs and simulators, the 18 AGSSs, the 23 AGSSs and simulators, and the 8 recent AGSSs). For the subset of 15 simulators, the mean improvement was 2 percentage points, for the subset of 8 telemetry simulators, 3 percentage points, and for the subset of 7 dynamics simulators, 7 percentage points.

Table 4-6. Reduction in RMS Percent Deviation Using Random SEF Values

	Root-mean-square percent deviation in predicted productivity[1]						
SEF Parameter	**33 AGSSs & Simulators**	**All 18 AGSSs**	**23 AGSSs & Simulators**	**8 Recent AGSSs**	**15 Simulators**	**8 Telemetry Simulators**	**7 Dynamics Simulators**
Base RMS value (no SEF used)	0.339	0.287	0.340	0.210	0.392	0.367	0.420
Maximum RMS	0.339	0.287	0.340	0.210	0.392	0.367	0.419
Mean RMS	0.331	0.275	0.329	0.196	0.372	0.338	0.350
Minimum RMS	0.293	0.233	0.287	0.138	0.300	0.257	0.123

[1] Percent deviation expressed as a decimal number, i.e., 0.339 means 33.9%.

4.4.2 Effect of Multiple SEF Parameters on Productivity Estimate

Equation 4-1 can be extended to test for the simultaneous effect of multiple SEF parameters. This is done by repeating—once for each additional desired SEF parameter—the portion of the equation within the braces, as shown in Equation 4-5. Table 4-7 lists the subsets of SEF parameters whose effects on productivity were tested for in this way. The same subsets of projects were again tested to find the change in the RMS percent deviation. The results are shown in Table 4-8.

$$\begin{aligned}(\text{Predicted productivity}) = {} & (\text{Base productivity}) \\ & \times \{\, 1.0 - [(\text{SEF Parameter}_1 - 3.0) \times (\text{Parameter Scale Factor}_1)]\, \} \\ & \times \{\, 1.0 - [(\text{SEF Parameter}_2 - 3.0) \times (\text{Parameter Scale Factor}_2)]\, \} \\ & \times \{...\} \times \{...\} \times ... \end{aligned} \qquad (4\text{-}5)$$

Table 4-7. Subsets of SEF Parameters Tested

Problem Characteristics:	PM01 - PM06
Technical Staff Characteristics:	ST07 - ST10
Technical Management Characteristics:	TM11 - TM15
Process Characteristics:	PC16 - PC24
Environment Characteristics:	EN25 - EN30
Product Characteristics:	PT31 - PT36

Table 4-8. Effect of Multiple SEF Parameters on Productivity Estimate (1 of 2)

Parameter Scale Factors Based on Actual SEF Values
(RMS percent deviation based on actual SEF data)[1]
[RMS percent deviation based on random SEF values]

SEF Parameter	33 AGSSs & Simulators	All 18 AGSSs	23 AGSSs & Simulators	8 Recent AGSSs	15 Simulators	8 Telemetry Simulators	7 Dynamics Simulators
None	(0.339)	(0.287)	(0.340)	(0.210)	(0.392)	(0.367)	(0.420)
PM01	0.030	-0.121	0.069	-0.014	0.095	-0.199	0.100
PM02	0.101	-0.096	0.092	0.202	0.136	0.247	0.166
PM03	0.012	0.111	-0.004	-0.014	0.046	0.124	0.043
PM04	-0.123	-0.095	-0.064	0.629	-0.062	-0.253	0.504
PM05	-0.074	-0.180	-0.010	0.430	0.027	-8.755	0.414
PM06	0.233	0.300	0.171	-0.607	0.250	0.732	-0.329
	(0.262)	(0.177)	(0.256)	(0.129)[2]	(0.263)	(0.243)[2]	(0.115)[2]
	[0.259]	[0.240]	[0.238]	[0.116]	[0.227]	[0.163]	[0.067]
ST07	0.064	-0.057	0.105	0.073	0.122	0.168	0.112
ST08	-0.127	-0.092	-0.155	-0.143	-0.162	0.092	-0.312
ST09	0.113	0.169	0.059	0.184	0.060	-0.334	0.167
ST10	0.092	0.078	0.030	0.205	0.053	-0.064	-0.0078
	(0.279)	(0.226)	(0.271)	(0.261)	(0.299)	(0.215)	(0.244)[2]
	[0.321]	[0.253]	[0.307]	[0.169]	[0.305]	[0.215]	[0.219]
TM11	0.063	0.036	0.024	-0.180	-0.058	-0.037	-0.127
TM12	0.131	0.162	0.080	0.064	0.051	0.030	-0.213
TM13	-0.030	-0.113	-0.036	-0.028	-0.066	-0.103	-0.147
TM14	-0.186	-0.274	0.067	0.421	0.304	0.192	0.541
TM15	0.153	0.241	0.033	0.011	0.040	0.132	0.163
	(0.310)	(0.197)	(0.297)	(0.309)[2]	(0.302)	(0.280)[2]	(0.297)[2]
	[0.306]	[0.252]	[0.286]	[0.139]	[0.294]	[0.252]	[0.044]

[1] Percent deviation expressed as a decimal number, i.e., 0.339 means 33.9%.

[2] These results are based on too few projects to have confidence in their validity.

Table 4-8. Effect of Multiple SEF Parameters on Productivity Estimate (2 of 2)

Parameter Scale Factors Based on Actual SEF Values
(RMS percent deviation based on actual SEF data)[1]
[RMS percent deviation based on random SEF values]

SEF Parameter	33 AGSSs & Simulators	All 18 AGSSs	23 AGSSs & Simulators	8 Recent AGSSs	15 Simulators	8 Telemetry Simulators	7 Dynamics Simulators
None	(0.339)	(0.287)	(0.340)	(0.210)	(0.392)	(0.367)	(0.420)
PC16	0.048	-0.188	0.081	0.154	0.163	0.144	0.147
PC17	-0.050	0.126	-0.062	-0.075	-0.017	-0.290	-0.059
PC18	0.017	-0.117	0.040	0.337	0.047	0.479	0.343
PC19	0.059	0.175	-0.032	-5.633	-0.50	-2.781	-4.153
PC20	0.055	-0.042	0.137	0.405	0.114	-0.582	0.379
PC22	-0.153	-0.097	-0.164	0.298	-0.117	-3.629	0.296
PC23	0.014	0.049	0.016	0.528	0.244	0.488	0.526
PC24	0.084	0.028	0.093	-0.788	0.044	0.479	-0.784
	(0.303)	(0.176)	(0.291)	(0.0000)[2]	(0.294)	(0.154)[2]	(0.000)[2]
	[0.296]	[0.119]	[0.277]	[0.0000]	[0.280]	[0.0000]	[0.0000]
EN25	0.181	-0.119	0.141	-0.710	0.161	0.112	-0.589
EN26	-0.025	-0.210	0.084	0.509	0.079	-0.120	0.505
EN27	-0.017	-0.044	0.022	-0.071	-0.026	0.098	-0.063
EN28	0.025	0.198	-0.064	-0.024	-0.029	0.082	-0.083
EN29	-0.0068	-0.079	0.061	0.349	0.030	-0.228	0.292
EN30	-0.059	0.011	-0.021	-0.074	-0.079	-0.089	0.051
	(0.284)	(0.230)	(0.270)	(0.123)[2]	(0.285)	(0.252)[2]	(0.112)[2]
	[0.292]	[0.145]	[0.283]	[0.082]	[0.260]	[0.117]	[0.073]
PT31	0.014	-0.089	0.063	0.100	0.0043	-0.517	0.134
PT32	0.155	0.119	0.078	-0.222	0.224	0.278	-0.211
PT33	-0.154	-0.282	-0.031	0.383	-0.091	-0.317	0.367
PT34	0.023	0.128	-0.053	-0.126	-0.078	0.197	-0.054
PT35	0.00093	0.037	0.050	0.332	0.0286	0.214	0.223
PT36	-0.080	0.019	-0.084	0.325	-0.153	-0.259	0.182
	(0.276)	(0.202)	(0.281)	(0.132)[2]	(0.274)	(0.148)[2]	(0.118)[2]
	[0.288]	[0.196]	[0.276]	[0.070]	[0.266]	[0.221]	[0.154]

[1] Percent deviation expressed as a decimal number, i.e., 0.339 means 33.9%.

[2] These results are based on too few projects to have confidence in their validity.

Although Table 4-8 seems to show greater reductions in the base RMS percent deviations than were shown in Table 4-5, these results must be interpreted carefully. Equation 4-4 only solves for one parameter scale factor at a time. Equation 4-5 simultaneously solves for between four and eight parameter scale factors, depending on the subset of SEF parameters. This greater flexibility makes it easier to find a final equation that more closely fits the data represented by the subset of projects. But this greater flexibility can lead to statistically unreliable results when the number of datasets (that is, the number of projects in the project subset) is only slightly greater than the number of simultaneous SEF parameters. These unreliable RMS percent deviations are pointed out in Table 4-8.

An example to consider is the subset of six "Problem Characteristics" parameters (PM01 through PM06) taken with the subset of seven dynamics simulators. Here Equation 4-5 has one dependent variable (predicted productivity) and six independent variables (the six PM factors). Thus it is a seven-dimensional equation. The dataset consists of *only* seven data points (the seven dynamics simulator projects). With experimental data containing experimental noise, many more data points are needed than the number of dimensions in the equation. So although Equation 4-5 results here in an RMS percent deviation of 11.5 percent, one cannot have confidence that the six resulting scale factors represent a true picture of the effects of PM01 through PM06 on the productivity of dynamics simulator projects.

The RMS percent deviations resulting from the actual SEF data are shown in parentheses in Table 4-8. For comparison, the RMS percent deviation that results from using a random set of SEF values is shown in brackets in Table 4-8. As can be readily seen, the improvements resulting from the use of random SEF values are of the same order as the improvements resulting from the actual SEL data SEF values. Because the parameter scale factors in Table 4-8 provide no more consistency in predicting productivity than is provided by the models based on random SEF data, one must be very skeptical of these models. One cannot say with confidence that the parameter scale factors in Table 4-8 represent valid models of the influence of multiple SEF parameters on productivity.

Perhaps by more carefully selecting the SEF parameters to include in Equation 4-5 one might develop a more successful model. This motivation guided the next stage of the investigation. It is useful here to consider again the models based on Equation 4-4 and displayed in Table 4-5. The values of the parameter scale factors in Table 4-5 show little consistency from one project subset to another. This could mean that the influences of the parameters cannot be observed by the methods of this study. It could alternatively mean that the influences have evolved over time and that they exhibit qualitatively different effects in different application areas. With this second possibility in mind, the study sought to focus further on one broad subset of projects that would reflect the way the FDD currently develops software.

The study chose the subset consisting of all AGSSs and simulators developed since DESIM, the first simulator. This is a large subset—23 projects—so one can evaluate the simultaneous effect of several SEF parameters on this subset without undue concern about statistically invalid results. The subset is broadly based—including both AGSSs and simulators—so a model derived from it could be widely used within the FDD. The subset closely represents the way the FDD develops software *today* because it contains all of the 8 recent AGSSs (ERBS through SAMPEX_2), and because 14 of the 15 simulators cover the same period. (The first simulator, DESIM, preceded ERBS by 3 years).

Table 4-5 shows that for this subset of projects, five parameters (PM01, PM02, EN25, EN26, and PT31) when applied individually to Equation 4-4, had moderate success at reducing the RMS percent deviation. The first four of these parameters are influences that the project manager might have some ability to estimate during a project, so attempts were made to model the simultaneous effect of these four parameters following the format of Equation 4-5. SEF parameters PM03 and PM06 had only slightly less success than the five parameters listed above at reducing the RMS percent deviation. In addition, the parameter scale factors for PM03 and PM06 (-10.2 percent and 18.2 percent, respectively) suggest a fairly strong link to productivity. These two parameters were therefore also included in the model.

To include the effects of other SEF parameters without overwhelming the equation with too many parameter scale factors, additional SEF parameters that were functionally similar and that behaved similarly in Table 4-5 were averaged to produce additional SEF parameters. The SEF parameters TM11, TM12, TM14, and TM15 all had nearly the same parameter scale factor (0.131) and nearly the same RMS percent deviation. These four SEF parameters were averaged to produce one new parameter. The SEF parameters PC16, PC19, PC20, PC22, PC23, and PC24 all had approximately the same parameter scale factor (0.092 to 0.125) and nearly the same RMS percent deviation. These six SEF parameters were averaged to produce another new parameter.

The resulting equation followed the format of Equation 4-5 and had eight parameters (six individual SEF parameters plus two SEF parameter averages). When the eight parameter scale factors were optimized, the resulting equation produced an RMS percent deviation of 23.1 percent for the 23 predicted productivities. This is an improvement of 11 percentage points over the prediction using no parameter scale factors to predict productivity for this subset of projects.

Next, the real SEF data for these 23 projects were replaced with random data, the parameter scale factors were again optimized, and the RMS percent deviation for the predicted productivities was computed. This 3-step randomization process was repeated 35 times. For the 35 RMS percent deviations computed, the maximum, mean, and minimum values were 31.4 percent, 25.2 percent, and 19.2 percent, respectively. The results of these tests with real and random SEF data are summarized in Table 4-9.

The tests with random SEF data show that most of the reduction in the RMS percent deviation is due to the mathematical ease of fitting the 23 final productivities in the dataset to any SEF data (even random data) when 8 parameter scale factors are available to be assigned. The mean improvement from random data was 9 percentage points; the improvement when applying actual SEF data was 11 percentage points, a difference of only 2 percentage points. As a result, one cannot claim with confidence that the model represented by the eight parameter scale factors in Table 4-9 truly models software development productivity in the FDD.

Table 4-9. Effect of Multiple SEF Parameters on Productivity Estimate for Subset of 23 AGSSs and Simulators

	Parameter Scale Factor (RMS percent deviation based on actual SEF data)[1] [RMS percent deviation based on random SEF values]	
SEF Parameter	**Real SEF Data**	**Random SEF Data (35 Trials)**
None	(0.340)	(0.340)
PM01	0.032	N/A
PM02	0.038	N/A
PM03	-0.048	N/A
PM06	0.180	N/A
TM__[2]	0.031	N/A
PC__[3]	-0.089	N/A
EN25	0.053	N/A
EN26	0.134	N/A
RMS % Deviation	(0.231)	
Maximum	[0.314]	
Mean	[0.252]	
Minimum	[0.192]	

1 Percent deviation expressed as a decimal number, i.e., 0.340 means 34.0%.

2 SEF parameters TM11, TM12, TM14, and TM15 averaged together to form one parameter.

3 SEF parameters PC16, PC19, PC20, PC22, PC23, and PC24 averaged together to form one parameter.

4.4.3 Other Productivity Analyses

After attempts to optimize Equations 4-4 and 4-5 failed to make a strong case for the benefit of productivity multipliers, the project sample was broadened slightly. Ten somewhat experimental projects from the SEL database were added to the original set. Using this enlarged and more diverse set of 43 projects, more attempts were then made to optimize Equations 4-4 and 4-5. Adding these 10 projects to the analysis, however, did not improve the ability to derive productivity multipliers.

As part of this study, various statistical tests were also performed on the set of 33 AGSSs and simulator projects to analyze the degree of association between the SEF values and productivity. For each subset of projects in Table 4-4, each SEF parameter was evaluated to compare its distribution of values (1-to-5 scale) to the distribution of final project productivities. Table 4-10 lists the statistical measures of association used. The measures were of three types: nominal measures, ordinal measures, and measures involving interval scales. Nominal measures provide no information about ranking or direction; they place items in bins such as "red," "blue," and "green." Ordinal measures include information about ranking and direction such as "good," "better," "best." Interval measures add an interval scale. This investigation of measures of association did not shed any more light on the relationship between SEF parameters and productivity within the FDD.

Table 4-10. Statistical Measures of Association Used

Nominal Measures:
Pearson *chi*-square
Goodman and Kruskal's *lambda*
Goodman and Kruskal's *tau*
Ordinal Measures:
Spearman correlation coefficient
Mantel-Haenszel *chi*-square
Somers's *d*
Interval Data Measures:
Pearson correlation coefficient, *r*
eta coefficient

4.5 Linear Regression Analysis of Correlations Between SEF Parameters and Effort

The second approach used to validate the usefulness of including SEF-type influences in effort estimates adopted traditional linear regression methods. This approach sought to determine correlations between technical effort (the dependent variable) on the one hand and DLOC and the SEF parameters on the other hand. The two project sets analyzed are shown in Table 4-3.

4.5.1 Linear Regressions Without SEF Data

This approach first tested the datasets without including any SEF variables, using linear regression to derive the values of *a* and *b* in Equation 4-6.

$$\text{Technical\_hours} = a + b \times \text{DLOC} \qquad (4\text{-}6)$$

The linear regression results for Equation 4-6 are tabulated in Table 4-11 and Table 4-12. The R-squared value was 0.623 and the RMS percent deviation was 78 percent for the older dataset. The R-squared value was 0.951 and the RMS percent deviation was 41 percent for the recent dataset. Figure 4-1 graphs the results of the linear regression for the older dataset. Figure 4-2 graphs the results of the linear regression for the recent dataset.

Table 4-11. Linear Regression Results for Equation 4-3 (24 Older FORTRAN Projects)

	a	b
Constants	3478.10	0.18
Standard Error	3485.23	0.03
R-squared	0.623	
RMS percent deviation	78%	
Degrees of freedom	22	

Table 4-12. Linear Regression Results for Equation 4-3 (15 Recent FORTRAN Projects)

	a	b
Constants	-1461.74	0.30
Standard Error	5836.44	0.02
R-squared	0.951	
RMS percent deviation	41%	
Degrees of freedom	13	

4.5.2 Linear Regressions With Actual and Random SEF Data

Next, individual SEF parameters were included in the equation and linear regression was used to derive the values of *a*, *b*, and *c* in Equation 4-7, where SEF_n represents the *n*th SEF parameter. Again, this was done for both the older dataset and for the recent dataset.

$$\text{Technical\_hours} = a + b \times \text{DLOC} + c \times \text{SEF}_n \qquad (4\text{-}7)$$

Equation 4-7 was computed 35 times for each dataset, once for each of the 35 SEF parameters. The regressions were then repeated with 35 sets of random integers (1 to 5) substituted for the actual SEF parameter values. For the older dataset, the resulting R-squared values varied from 0.623 to 0.723 when actual SEF values were used, and from 0.623 to 0.691 when the random numbers were substituted. The results are graphed in Figure 4-3. For the recent dataset, the resulting R-squared values varied from 0.951 to 0.968 when actual SEF values were used, and from 0.951 to 0.965 when the random numbers were substituted. The results are graphed in Figure 4-4.

For the older dataset, the resulting RMS percent deviations varied from 64 percent to 117 percent when actual SEF values were used, and from 69 percent to 105 percent when the random numbers were substituted. The results are graphed in Figure 4-5. For the recent dataset, the resulting RMS percent deviation values varied from 27 percent to 44 percent when actual SEF values were used, and from 31 percent to 50 percent when the random numbers were substituted. The results are graphed in Figure 4-6.

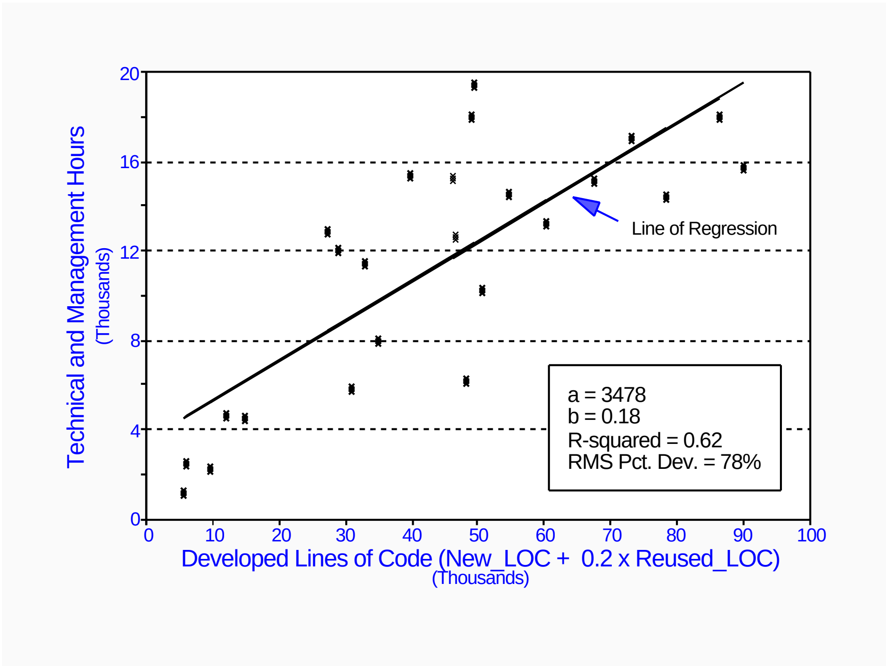


Figure 4-1. Effort as a Function of DLOC for 24 Older FORTRAN Projects

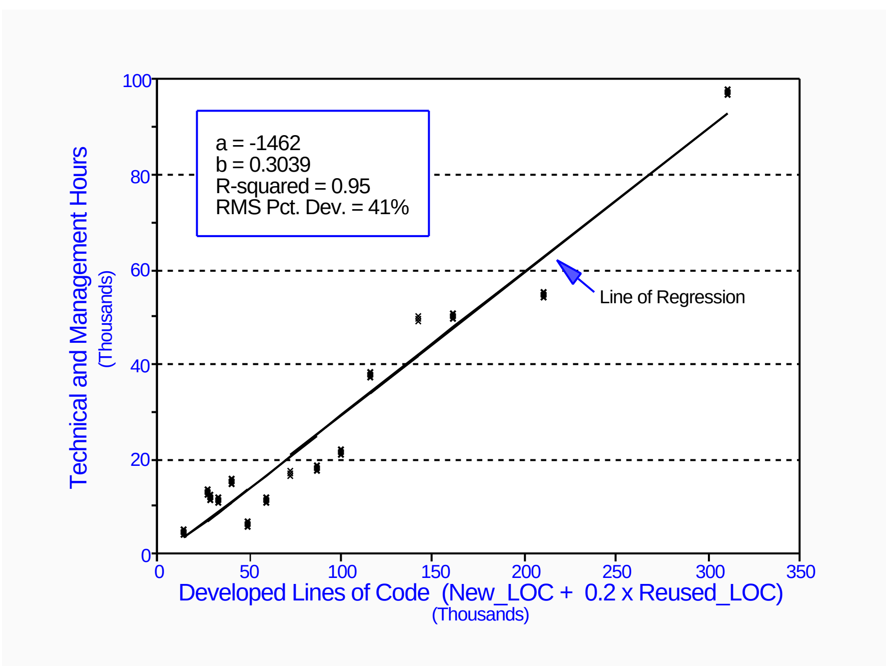


Figure 4-2. Effort as a Function of DLOC for 15 Recent FORTRAN Projects

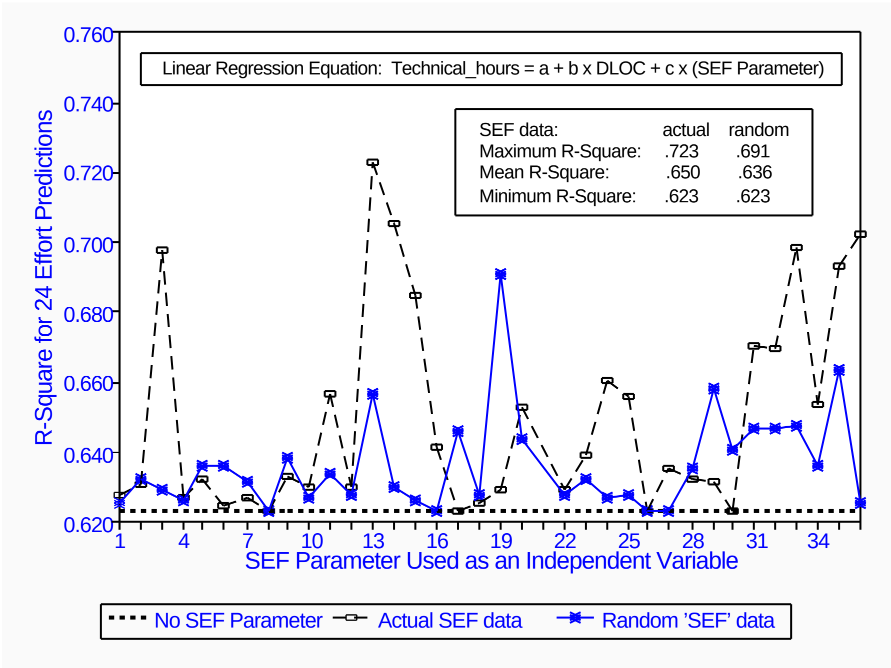


Figure 4-3. Accuracy of Effort Prediction (Measured by R-Squared) for 24 Older FORTRAN Projects

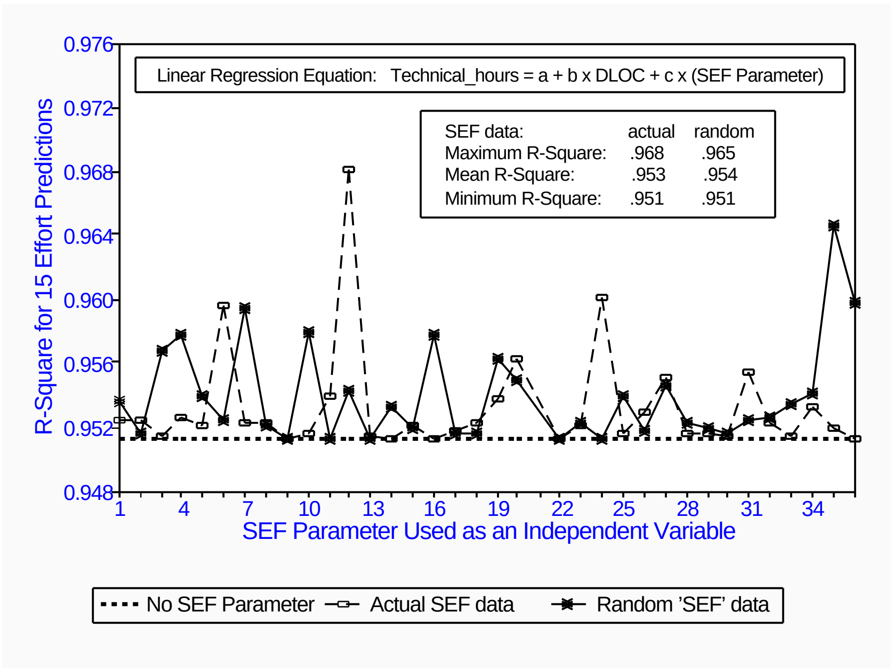


Figure 4-4. Accuracy of Effort Prediction (Measured by R-Squared) for 15 Recent FORTRAN Projects

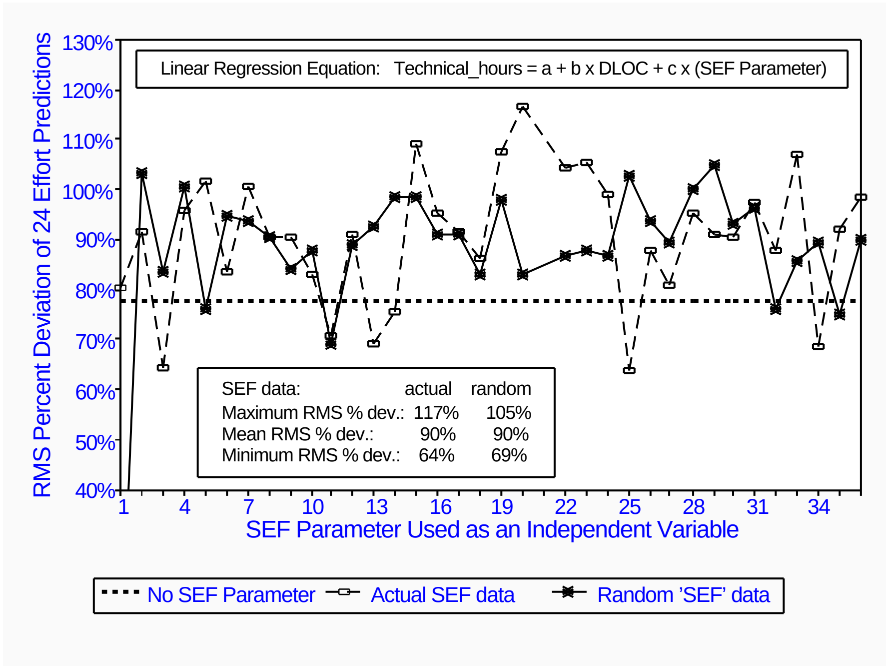


Figure 4-5. Accuracy of Effort Prediction (Measured by RMS Percent Deviation) for 24 Older FORTRAN Projects

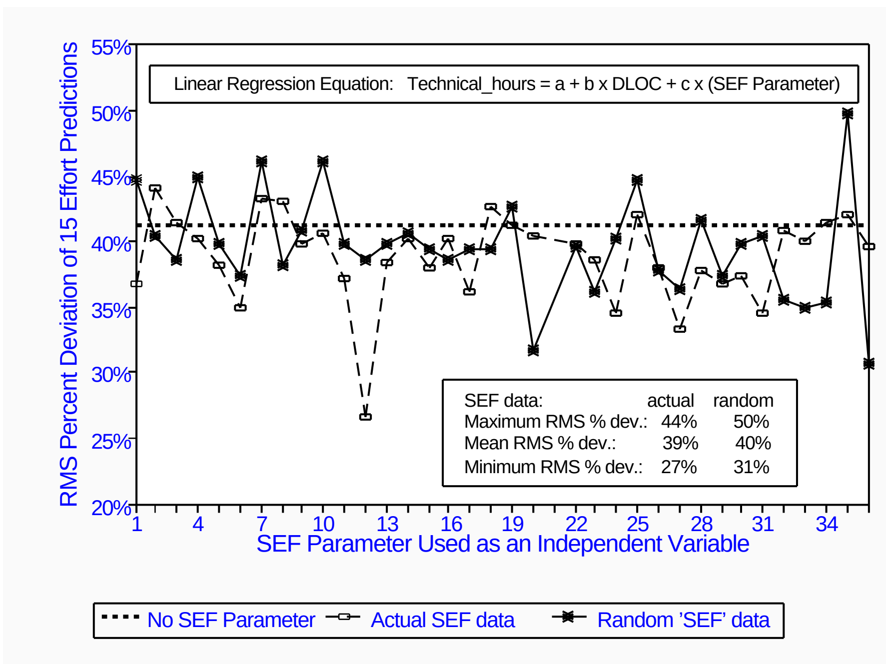


Figure 4-6. Accuracy of Effort Prediction (Measured by RMS Percent Deviation) for 15 Recent FORTRAN Projects

If Figures 4-3 and 4-4 had shown fairly good agreement on which of the SEF parameters provided the highest improvement in the R-squared values, one would feel confident that this approach had truly identified several SEF parameters that significantly affected effort. But there is very little agreement between Figures 4-3 and 4-4. For the older dataset, the eight SEF parameters showing the most improvement in R-squared values are numbers 3, 13, 14, 15, 31, 33, 35, and 36. For the recent dataset, the eight SEF parameters showing the most improvement in R-squared values are numbers 6, 11, 12, 19, 20, 24, 27, and 31. Only one value, number 31, appears both in the group of eight from the older dataset and in the group of eight from the recent dataset. SEF parameter number 13, on the other hand, shows a very significant improvement in R-squared for the older dataset but virtually no improvement for the more recent dataset. The improvement observed on the older dataset may have been due merely to a chance association between DLOC, SEF13, and technical effort for that dataset. This explanation gains credence from the number of random SEF values that demonstrate significant improvements in R-squared values.

Figures 4-5 and 4-6 show which SEF parameters (and sets of random integers) resulted in the most reduction in the RMS percent deviation in the predicted technical effort for the older dataset and for the recent dataset, respectively. Again, there is no relationship between the actual SEF parameters that showed the most reduction for the older dataset and those actual SEF parameters that showed the most reduction for the recent dataset. Of the 35 sets of random numbers, several again provided significant improvement in the RMS percent deviation.

Figure 4-6 demonstrates that for the recent dataset, 5 of the 35 SEF parameters showed improvements of 6 percentage points or more. Likewise, 4 of the 35 random number sets resulted in improvements of the RMS percent deviation of 6 percentage points or more. Furthermore, in comparing the 70 linear regressions for the recent dataset, half with real SEF data and half with random data, 2 of the 3 most dramatic improvements in the RMS percent deviation were due to random numbers rather than to actual SEF data.

4.6 Conclusions

Each of the two approaches used in this section points to some SEF variables that seem to improve predictions of either productivity or effort. Closer scrutiny, however, reveals that there is no evidence of a causal link between these parameters and either productivity or effort. The SEF parameters that "improve" predictions vary with the subset of projects and the timeframe of the projects; there is no continuity that would suggest the discovery of a causal relationship. Moreover, the handful of such parameters that result in improved fits between the model and the data is about what would be expected from a set of 35 parameters that are *unrelated* to productivity or effort. This was demonstrated in each approach by showing that random numbers substituted for the SEF values provided about the same frequency and degree of improvement as did the actual SEF values.

The SEF data in the SEL database provide no demonstrable evidence that inclusion of estimates for such factors as problem complexity or team experience will significantly improve a manager's estimate of project effort. When making estimates for project effort, managers are still encouraged to include such factors as problem complexity or team experience based on their own personal experience, but the database of experience represented by the SEF data in the SEL database provides no guidelines.

Section 5. Schedule Analysis

This section validates the current schedule models using over 30 projects from the Flight Dynamics environment. These projects included telemetry simulators, dynamics simulators, and AGSSs; some were developed in FORTRAN and others in Ada.

Section 5.1.1 develops and compares schedule duration models for several different subsets of projects. This analysis indicates that separate formulas should be used to estimate schedule duration for AGSSs and simulators. The same schedule duration model should be used for both Ada and FORTRAN simulators. (So far all AGSSs in the FDD have been developed in FORTRAN.) Section 5.1.2 validates the schedule duration model by comparing its predictions with the actual schedule durations for the completed projects. Section 5.1.3 shows the impact of growth on schedule for typical projects and those projects with high reuse. Section 5.2 discusses the distribution of schedule by life-cycle phase.

5.1 Total Schedule Duration

5.1.1 Formulating a Schedule Prediction Model

This section formulates schedule duration models for several subsets of projects: FORTRAN projects, Ada projects, AGSSs, telemetry simulators, and dynamics simulators. Each model is based on the actual end-of-project effort and schedule. Table 5-1 lists the data for the projects analyzed in this section, and Table 5-2 presents schedule data grouped by application type and development language.

The study deemed projects as schedule outliers when they differed by more than 25 percent from the average of the other projects in their category; these outliers are footnoted in Table 5-2. The Gamma Ray Observatory Dynamics Simulator (GROSS) and the Gamma Ray Observatory AGSS (GROAGSS) were also eliminated from the next step of the analysis because their durations, following the Challenger disaster, were unusually long. SAMPEXTS was excluded because it piloted a stream lined life-cycle resulting from high reuse.

The COCOMO optimal formula for computing project duration (without corrections for factors such as complexity) is

$$\text{Duration} = 3.3\ (\text{staff months})^{.3}$$

The projects examined in this study were evaluated according to a generalized formula:

$$\text{Duration} = \text{COEFF} \times (\text{staff months})^{.3}$$

Table 5-1. Project Data Used in Schedule Analysis

Project	Type	Lang.	Devel. Period[1]	Duration (Weeks)	Effort (Hours)
PAS	AGSS	F	05/76 - 09/77	69	15760
ISEEB	AGSS	F	10/76 - 09/77	50	15262
AEM	AGSS	F	02/77 - 03/78	57	12588
SEASAT	AGSS	F	04/77 - 04/78	54	14508
SMM	AGSS	F	04/78 - 10/79	76	14371
MAGSAT	AGSS	F	06/78 - 08/79	62	15122
FOXPRO	AGSS	F	02/79 - 10/79	36	2521
DEA	AGSS	F	09/79 - 06/81	89	19475
DEB	AGSS	F	09/79 - 05/81	83	17997
DESIM	TS	F	09/79 - 10/80	56	4466
ERBS	AGSS	F	05/82 - 04/84	97	49476
DERBY	DS	F	07/82 - 11/83	72	18352
GROSS	DS	F	12/84 - 10/87	145	15334
COBEDS	DS	F	12/84 - 01/87	105	12005
ASP	AGSS	F	01/85 - 09/86	87	17057
GROAGSS	AGSS	F	08/85 - 03/89	188	54755
GROSIM	TS	F	08/85 - 08/87	100	11463
COBSIM	TS	F	01/86 - 08/87	82	6106
COBEAGSS	AGSS	F	06/86 - 09/88	116	49931
GOADA	DS	A	06/87 - 04/90	149	28056
GOFOR	DS	F	06/87 - 09/89	119	12804
GOESAGSS	AGSS	F	08/87 - 11/89	115	37806
GOESIM	TS	A	09/87 - 07/89	99	13658
UARSAGSS	AGSS	F	11/87 - 09/90	147	89514
UARSDSIM	DS	F	01/88 - 06/90	128	17976
UARSTELS	TS	A	02/88 - 12/89	94	11526
EUVEAGSS	AGSS	F	10/88 - 09/90	102	21658
EUVETELS	TS	A	10/88 - 05/90	83	4727
EUVEDSIM	DS	A	10/88 - 09/90	121[2]	20775[2]
SAMPEXTS	TS	A	03/90 - 03/91	48	2516
SAMPEX	AGSS	F	03/90 - 11/91	85	4598
POWITS	TS	A	03/90 - 05/92	111	11695

[1] Design phase through acceptance test phase.

[2] Duration adjusted by +15% and Effort adjusted by +10% because EUVEDSIM did not have an acceptance test phase. These values are consistent with those of the *Ada Size Study Report*.

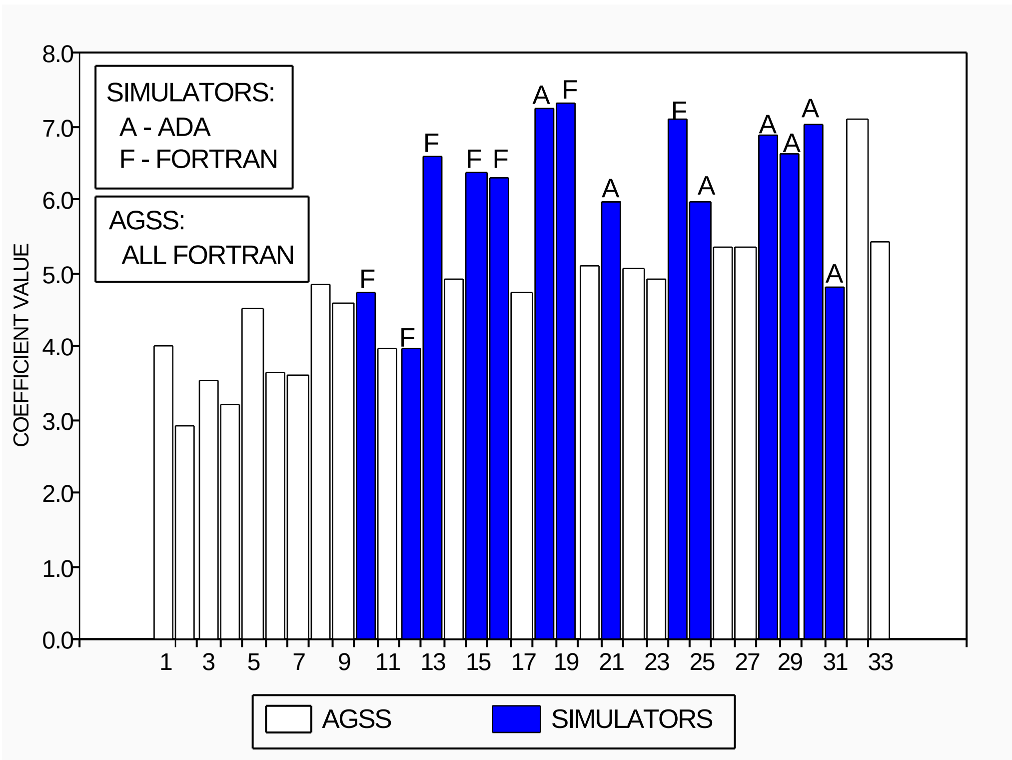
Key:
- A — Ada
- AGSS — Attitude Ground Support System
- DS — Dynamics Simulator
- F — FORTRAN
- TS — Telemetry Simulator

Table 5-2. Project Duration Formula Coefficients

	FORTRAN	**COEFF**[1]	**ADA**	**COEFF**
TELEMETRY SIMULATORS	DESIM	4.72	GOESIM	5.97
	GROSIM	6.36	UARSTELS	5.97
	COBSIM	6.30	EUVETELS	6.88
			SAMPEXTS	4.81
			POWITS	7.02
	AVG.	5.79		6.13
	AVG.[3]	5.79		6.13
DYNAMICS SIMULATORS	DERBY	3.98[2]	GOADA	7.24
	GROSS	5.87	EUVEDSIM	6.44
	COBEDS	6.58		
	GOFOR	7.32		
	UARSDSIM	7.11		
	AVG.	6.17		6.84
	AVG.[3]	6.72		6.84
	FORTRAN	**COEFF**	**FORTRAN**	**COEFF**
AGSS	PAS	3.99	ERBS	3.98
	ISEEB	2.92[2]	ASP	4.91
	AEM	3.52	GROAGSS	7.48[2]
	SEASAT	3.20[2]	COBEAGSS	4.73
	SMM	4.52	GOESAGSS	5.11
	MAGSAT	3.63	UARSAGSS	5.05
	FOXPRO	3.61	EUVEAGSS	5.36
	DEA	4.83	SAMPEX	7.10[2]
	DEB	4.61		
	AVG.			4.62
	AVG. [3]			4.45

[1] COEFF = Schedule Duration/(Staff Months)$^{.3}$.
[2] Outliers - More than 25% different from average.
[3] Average values excluding outliers.

in which the Coefficient, COEFF, is derived by substituting the actual schedule duration (calendar months) and actual effort (staff months) for each of these completed projects and then solving for the Coefficient for each project. Figure 5-1 compares the duration formula Coefficients for all projects except GROAGSS and GROSS.



1. PAS	12. DERBY	23. UARS_2
2. ISEEB	13. COBEDS	24. UARSDSIM
3. AEM	14. ASP	25. UARSTELS
4. SEASAT	15. GROSIM	26. EUVEAGSS
5. SMM	16. COBSIM	27. EUVE_2
6. MAGSAT	17. COBEAGSS	28. EUVETELS
7. FOXPRO	18. GOADA	29. EUVEDSIM
8. DEA	19. GOFOR	30. POWITS
9. DEB	20. GOESAGSS	31. SAMPEXTS
10. DESIM	21. GOESIM	32. SAMPEX
11. ERBS	22. UARSAGSS	33. SAMPEX_2

Figure 5-1. Coefficients of Schedule Duration Formula

Plotting these data in this way, beginning with project 13, COBEDS, and ending with project 30, POWITS, the data reveal that the simulator projects show a pattern of larger Coefficient values than the AGSSs.

The next step of this study was to develop schedule-duration formulas in staff months (SM) by type of project. To develop the necessary formulas, the study selected project data beginning with COBEDS, begun in 1984, which corresponds roughly to the time at which the original *Recommended Approach to Software Development* (Reference 9) was adopted as the standard software development process in the FDD.

An optimizing technique, based on the lowest RMS percent deviation, was used to solve for the optimal Coefficient and exponent values for the schedule-duration formula. The projects (listed in Table 5-3) were grouped and analyzed by type: AGSSs, telemetry simulators, dynamics simulators, and all simulators (telemetry and dynamics). Table 5-4 presents the results of the optimizing process, first solving for the best Coefficient using the exponent .3 for each project type, and then solving for the best Coefficient and exponent for each project type.

Table 5-3. Projects Used in Formulating Schedule Equations

	FORTRAN	COEFF*
TELEMETRY SIMULATORS	GROSIM	6.36
	COBSIM	6.30
	GOESIM	5.97
	UARSTELS	5.97
	EUVETELS	6.88
	POWITS	7.02
DYNAMICS SIMULATORS	COBEDS	6.58
	GOFOR	7.32
	UARSDSIM	7.11
	GOADA	7.24
	EUVEDSIM	6.44
AGSS	ASP	4.91
	COBEAGSS	4.73
	GOESAGSS	5.11
	UARS_2	4.92
	EUVEAGSS	5.36
	SAMPEX_2	5.42

*COEFF = Schedule Duration/(Staff Months)$^{.3}$

All eight cases shown in Table 5-4 have RMS percent deviation results of less than 10 percent; therefore, the study recommends using only one formula for all simulators. Also, for the sake of consistency and simplicity, the exponent .3 should be used for both AGSSs and simulators.

Figure 5-2 graphs actual project duration in weeks as a function of actual staff-months of effort for each project included in this step of the analysis. The figure also depicts two separate curves: one for simulators and one for AGSSs using a .3 exponent with the best Coefficient.

In conclusion, the Cost and Schedule Estimation Study recommends these formulas:

$$\text{AGSS Duration} = 5.0(SM)^{.3}$$

$$\text{Simulator Duration} = 6.7(SM)^{.3}$$

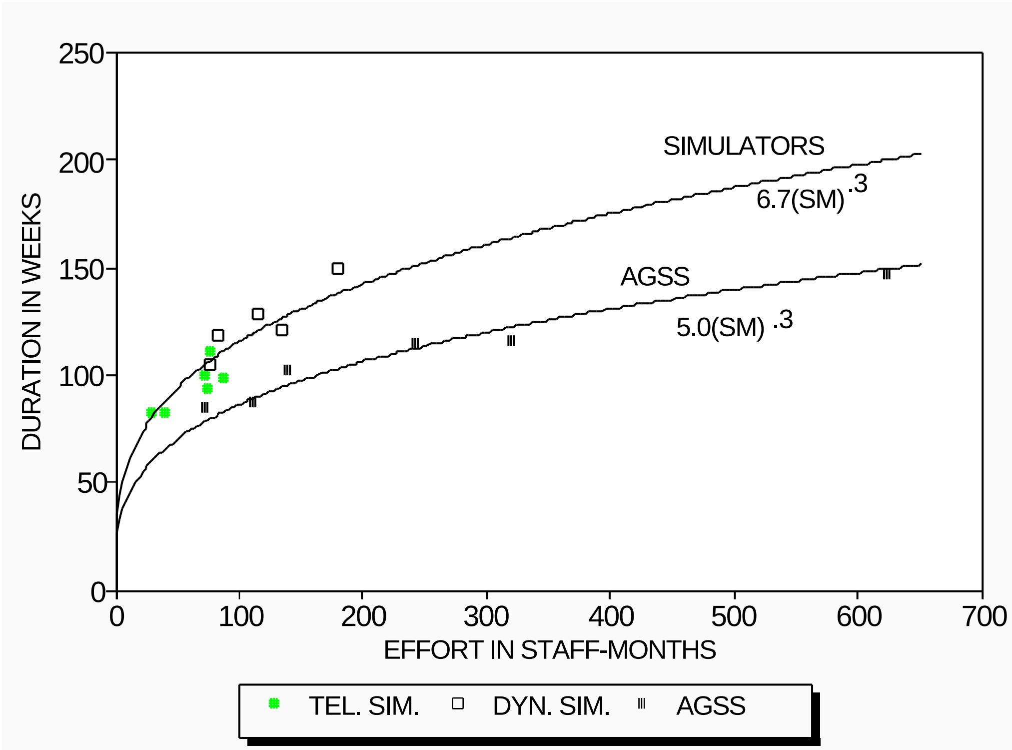
The simulator-duration formula is recommended for computing schedule duration for both telemetry and dynamics simulator projects, whether they are written in FORTRAN or Ada. The corresponding duration formulas based on support hours as well as technical and management hours are

$$\text{AGSS Duration} = 4.9(SM)^{.3}$$

$$\text{Simulator Duration} = 6.5(SM)^{.3}$$

Table 5-4. Summary of Duration Formulas

SOLVING DURATION FORMULA FOR COEFFICIENT ONLY (EXPONENT .3 IS CONSTANT)				
	AGSS	TELEMETRY SIMULATORS	DYNAMICS SIMULATORS	ALL SIMULATORS
Formula	$5.0(SM)^{.3}$	$6.4(SM)^{.3}$	$6.9(SM)^{.3}$	$6.7(SM)^{.3}$
RMS	4.9%	6.2%	6.5%	7.8%
SOLVING DURATION FORMULA FOR BOTH COEFFICIENT AND EXPONENT				
	AGSS	TELEMETRY SIMULATORS	DYNAMICS SIMULATORS	ALL SIMULATORS
Formula	$6.3(SM)^{.29}$	$6.7(SM)^{.29}$	$6.0(SM)^{.33}$	$5.3(SM)^{.35}$
RMS	3.8%	6.1%	6.4%	7.4%



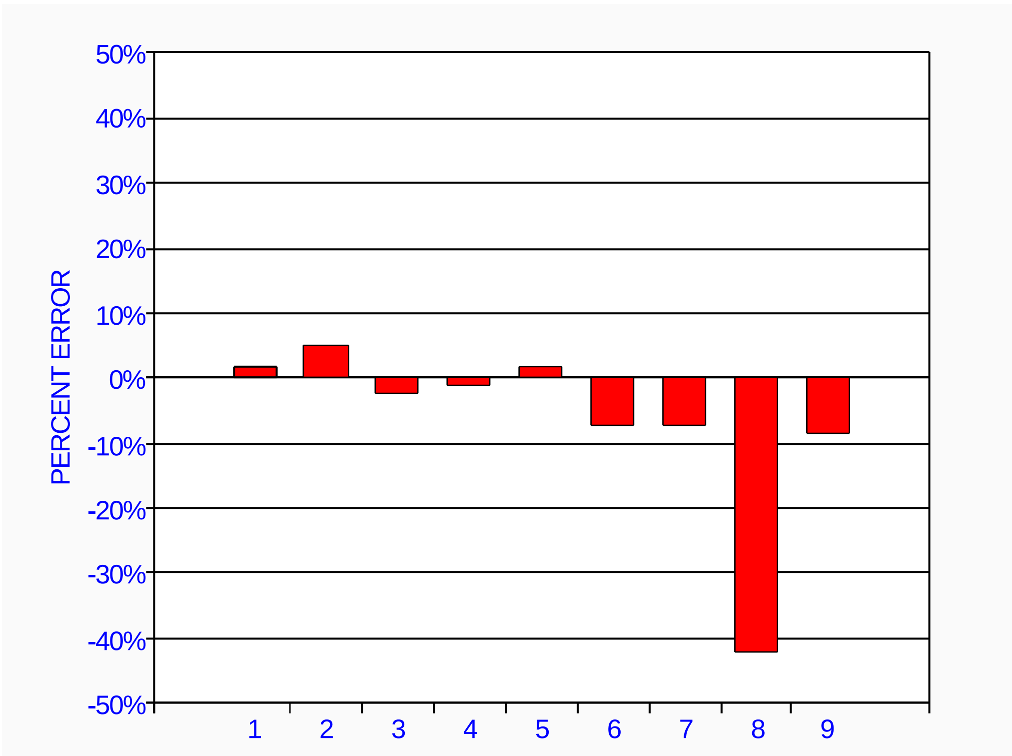
Telemetry Simulators:	Dynamics Simulators:	AGSSs:
EUVETELS	COBEDS	SAMPEX_2
COBSIM	GOFOR	ASP
GROSIM	UARSDSIM	EUVEAGSS
UARSTELS	EUVEDSIM	GOESAGSS
POWITS	GOADA	COBEAGSS
GOESIM		UARS_2

Figure 5-2. Schedule Duration Versus Technical and Management Effort

5.1.2 Accuracy of Schedule Duration Model Based on Final Projects Statistics

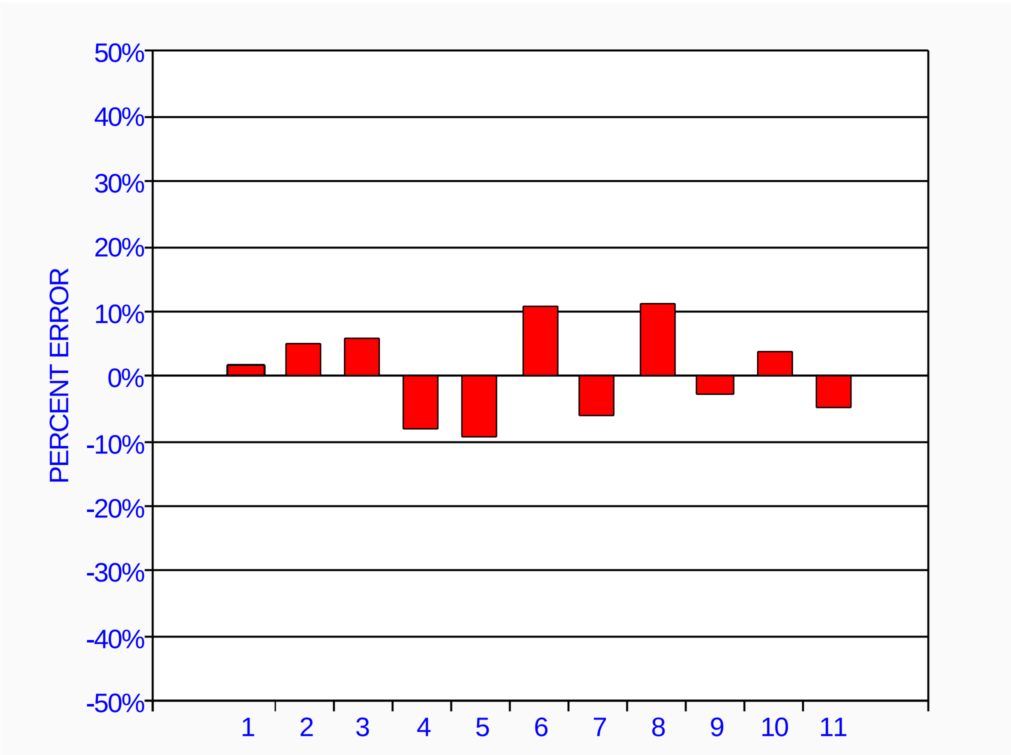
This section presents the accuracy of the schedule model formulated in Section 5.1. The first comparison is of the AGSS actual schedules with the estimated schedules, applying the AGSS COEFF, 5.0, to the schedule duration model. The same type of comparison is presented for simulators, combining telemetry and dynamics simulators, including those developed in FORTRAN and Ada. In the case of the simulators, the simulator COEFF, 6.7, is substituted in the schedule model.

Figure 5-3 shows the accuracy of applying the AGSS COEFF parameter to the schedule model, and Figure 5-4 shows the accuracy of applying the simulator COEFF parameter to the schedule model. SAMPEX, is an outlier and was not used in formulating the schedule model. However, SAMPEX_2 fits the model better; it is within the ± 10 percent accuracy range. All of the simulator projects fit the schedule duration model well; they are all close to the 10 percent accuracy range.



1. ASP	4. UARSAGGS	7. EUVE_2
2. COBEAGGS	5. UARS_2	8. SAMPEX
3. GOESAGGS	6. EUVEAGSS	9. SAMPEX_2

Figure 5-3. Accuracy of Schedule Duration Model for AGSSs (Based on Actual Technical and Management Effort)



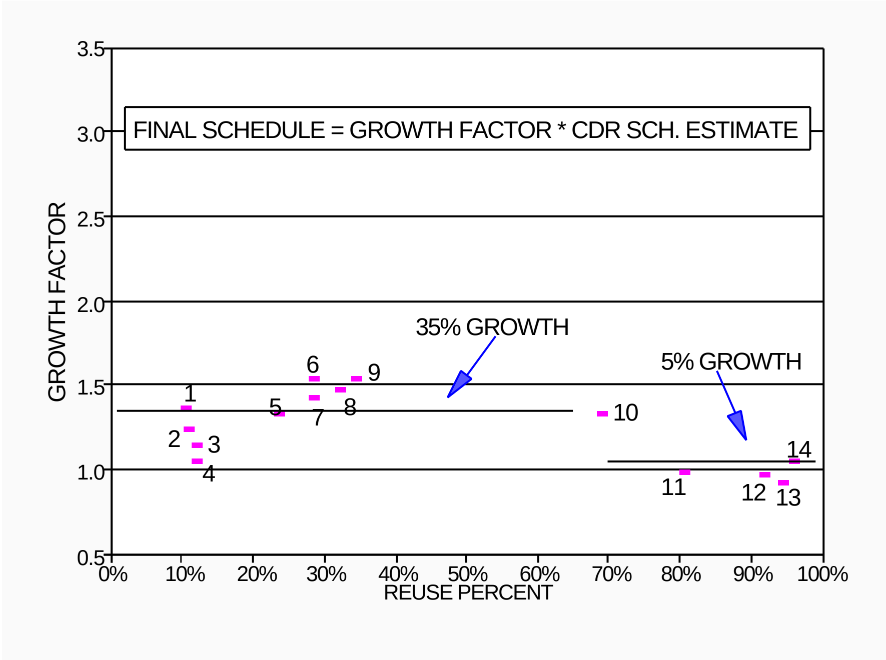
1. COBEDS	5. GOFOR	9. EUVETELS
2. GROSIM	6. GOESIM	10. EUVEDSIM
3. COBSIM	7. UARSDSIM	11. POWITS
4. GOADA	8. UARSTELS	

Figure 5-4. Accuracy of Schedule Duration Model for Simulators (Based on Actual Technical and Management Effort)

5.1.3 Analysis of Schedule Growth

Section 3.2.2 presented a scatter plot of the growth in a project's DLOC from the CDR estimate to the end-of-project DLOC versus the project's reuse. Likewise, the study examined the growth in schedule for these same projects by reuse level. Figure 5-5 presents a plot of the schedule growth factor, obtained by dividing the final duration by the estimated project schedule at the time of the CDR. The graph depicts that the percentage of growth, or growth factor, for schedule is smaller than the growth in DLOC, as seen in Section 3.2.2. For projects with less than 70-percent reuse, the growth is in the range of 35 percent, and for projects with reuse above 70-percent reuse, the growth is near zero percent.

The schedule growth rates presented in Figure 5-5 provide a historical pattern of how schedules were extended in the past. Schedule extensions were typically granted when there was a slip in the launch date. These extensions cannot be anticipated and are not built into the planning process. The schedule planning for a project is further discussed in Section 7.



1. COBSIM	6. GOADA	11.EUVEAGSS
2. UARSAGSS	7. GOESIM	12. SAMPEX
3. COBEAGSS	8. GOFOR	13. SAMPEXTS
4. GOESAGSS	9. UARSTELS	14. EUVETELS
5. UARSDSIM	10. POWITS	

Figure 5-5. Schedule Growth Factors (Actual Duration Divided by CDR Estimate)

5.2 Distribution of Schedule by Life-Cycle Phase

Successful software development project planning requires the manager to set *meaningful* intermediate milestones. In the SEL environment, major intermediate milestones define the end of the life-cycle phases. Thus, it is important to understand what percent of the total project duration is spent in the life-cycle phases on the typical project. This forms the profile model of schedule distribution against which completed projects can be compared and assessed. Based on this profile model, a planning model for schedule distribution is derived that, when put in place at the start of the project, will lead to project results similar to the profile model.

Tables 5-5 and 5-6 present the analysis of current SEL project data to determine the current profile model for schedule distribution among the life-cycle phases. The relationship between the SEL planning model (from Reference 5) and this profile model is addressed in Section 7.

The study examined percent of schedule distributed by life-cycle phase, beginning chronologically with the ERBS project. The projects are combined into groups, with an average and standard deviation for each group. The low–reuse model combines the AGSS, FORTRAN simulator, and Ada project and does not include the high–reuse projects. The high–reuse model, a preliminary model is based on combining both Ada and FORTRAN projects with a reuse level of 65 percent and higher; the model shows high variability. The actual percentages are provided to allow the reader to model other combinations of projects. Appendix C examines the projects in more detail, listing those with the most and least percentages of schedule in each phase as well as the most and least percentages of effort by phase and percentages effort by activity.

Table 5-5. Percentage of Schedule Distribution by Phase

Project	Reuse Percent	Design	Code	System Test	Acceptance Test
ERBS	13%	43.3%	34.0%	12.4%	10.3%
COBEDS	27%	34.3%	22.9%	31.4%	11.4%
ASP	13%	29.9%	31.0%	14.9%	24.1%
GROSIM	18%	35.0%	39.0%	17.0%	9.0%
COBSIM	11%	28.0%	40.2%	18.3%	13.4%
COBEAGSS	12%	26.7%	26.7%	20.7%	25.9%
GOADA	29%	27.5%	28.9%	30.9%	12.8%
GOFOR	32%	25.2%	27.7%	31.9%	15.1%
GOESAGSS	12%	27.0%	38.3%	16.5%	18.3%
GOESIM	29%	34.3%	29.3%	8.1%	28.3%
UARSAGSS	11%	30.6%	36.1%	16.3%	17.0%
UARSDSIM	24%	25.8%	45.3%	7.0%	21.9%
UARSTELS	35%	31.9%	29.8%	10.6%	27.7%
EUVEAGSS	78%	37.3%	33.3%	14.7%	14.7%
EUVETELS	96%	26.5%	42.2%	12.0%	19.3%
POWITS	69%	26.1%	31.5%	8.1%	34.2%
SAMPEXTS	95%	47.9%	8.3%	16.7%	27.1%
SAMPEX	92%	45.9%	14.1%	22.4%	17.6%

Table 5-6. Models for Schedule Distribution by Phase

Model	Design	Code	Test
5 AGSS PROJECTS	32 ± 6%	33 ± 4%	35 ± 8%
4 FORTRAN SIMULATORS	29 ± 4%	38 ± 6%	33 ± 8%
3 ADA SIMULATORS	31 ± 3%	29 ± 0%	40 ± 3%
12 LOW REUSE PROJECTS	30 ± 5%	34 ± 6%	36 ± 7%
5 HIGH REUSE PROJECTS	37 ± 9%	26 ± 13%	37 ± 6%

Section 6. Conclusions and Recommendations

6.1 Conclusions

The study concludes the following:

- The standard SEL effort estimation equation, based on a size estimate adjusted for reuse, is best for predicting effort in the FDD environment. Of the three effort model parameters—productivity, cost to reuse code, and growth factor—the productivity and reuse costs vary with language, whereas the growth factor varies with the level of reuse. The effort model parameters do not depend on the application type (that is, AGSS, telemetry simulator, or dynamics simulator).

- Developed lines of code (DLOC) (total SLOC adjusted for reuse) is an accurate basis for estimating total project effort. For FORTRAN projects, DLOC should be computed with a 20-percent weight given to reused SLOC. (The 20-percent weighting is the reuse cost parameter.) For Ada projects, DLOC should be computed with a 30-percent weight given to reused SLOC.

  Note: The significant cost savings evidenced by SAMPEX AGSS and SAMPEXTS, two recent projects with very high reuse levels, suggest a divergence from the 30-percent and 20-percent reuse costs. For such high-reuse projects as these, a much lower reuse cost may be appropriate, perhaps as low as 10 percent. SAMPEXTS piloted a streamlined development process for high reuse projects, combining some documents and combining the PDR with the CDR; the project's low reuse cost may result from these process changes as well as from the percentage of reused code. Data from more high-reuse projects are needed before certifying this as a trend.

- The productivity experienced on recent FORTRAN AGSSs varied from 3 to 5 DLOC per technical staff/technical management hour. For planning purposes, a conservative productivity value of 3.5 DLOC per technical staff/technical management hour is recommended. When support staff hours are included in the plan, an overall productivity of 3.2 DLOC per hour should be used.

- The productivity on recent Ada projects showed less variability than did the FORTRAN projects. For planning purposes, a productivity of 5.0 DLOC per technical staff/technical management hour is recommended. When support staff hours are included in the plan, an overall productivity of 4.5 DLOC per hour should be used.

- The SEF data in the SEL database provide no demonstrable evidence that inclusion of estimates for such factors as problem complexity or team experience will significantly improve a manager's estimate of project effort. When making estimates for project effort, managers are still encouraged to include such factors as problem complexity or team experience *based on their own personal experience*, but the database of experience represented by the SEF data in the SEL database provides no guidelines.

- For projects with moderate to low code reuse (less than 70 percent), the post-CDR growth in DLOC due to requirements changes and TBDs is commensurate with past SEL experience: 40 percent. For projects with high code reuse (70 percent or more), the post-CDR growth in DLOC is only about half as much: 20 percent.

- An exponential model like COCOMO can be used to predict the duration of projects from total project effort; the COCOMO multiplicative factor of 3.3 must be replaced with a factor of 5.0 for AGSSs (6.7 for simulators) when based on technical staff/technical management hours and 4.9 for AGSSs (6.5 for simulators) when support hours are also included.

- For projects with moderate to low code reuse, the post-CDR growth in schedule is 35 percent. For projects with high reuse, the post-CDR growth in schedule is 5 percent.

- Based on the final project statistics for moderate to low reuse projects (less than 70-percent code reuse), the distribution of the total effort and schedule among the life-cycle phases is as follows:

Phase	**Effort**	**Schedule**
Design:	24 ± 3%	30 ± 5%
Code:	45 ± 6%	34 ± 6%
Test:	31 ± 5%	36 ± 7%

- Based on the final project statistics for high-reuse projects (70 percent or more code reuse), the distribution of the total effort and schedule among the life-cycle phases is as shown below. The larger standard deviations for high-reuse projects demonstrate that the development process for high-reuse projects is still evolving, resulting in significant variability in the effort distribution. As more high-reuse projects are completed, it should become possible to more accurately model the high-reuse projects.

Phase	**Effort**	**Schedule**
Design:	26 ± 14%	37 ± 9%
Code:	38 ± 12%	26 ± 13%
Test:	36 ± 3%	37 ± 6%

- Based on the final project statistics for low-reuse projects, the distribution of the total effort among the software development activities is

Activity	**Effort**
Design:	$21 \pm 4\%$
Code:	$26 \pm 4\%$
Test:	$25 \pm 5\%$
Other:	$28 \pm 9\%$

- Based on the final project statistics for high-reuse projects, the distribution of the total effort among the software development activities is

Activity	**Effort**
Design:	$17 \pm 5\%$
Code:	$17 \pm 6\%$
Test:	$32 \pm 6\%$
Other:	$34 \pm 8\%$

- Requirements changes and system growth can cause project effort and schedule to diverge from their predicted distributions in the manager's initial plan. In order to minimize the effects of requirements changes and system growth on project cost and schedule, a manager should usually *plan* for the following distributions of the total effort and schedule among the life-cycle phases. (See Section 7 for a full discussion of how to apply SEL planning models and relate them to baseline models for effort and schedule.)

Phase	**Effort**	**Schedule**
Design:	30%	35%
Code:	40%	30%
Test:	30%	35%

6.2 Recommendations

For future projects developed within the FDD environment, the following recommendations are made:

- The initial effort estimate should be based on the standard SEL effort estimation model with an appropriate growth factor applied:

$$\text{Effort} = \text{DLOC} \times \text{Growth Factor} / \text{Productivity}$$

Note: Although the SEF data in the SEL database provide no guidelines for adjusting this initial effort estimate to account for such factors as team experience or problem complexity, managers are still encouraged to include such factors based on their own personal experience.

- DLOC should be computed as follows:

$$\text{DLOC} = \text{new SLOC} + (\text{reuse cost}) \times \text{reused SLOC}$$

Language	**Reuse Cost**
FORTRAN	0.2
Ada	0.3

Note: The 30-percent reuse cost for Ada projects was proposed by the *Ada Size Study Report* (Reference 1). At that time only a small number of completed Ada projects were available for analysis, and the Ada process had been evolving from project to project. Since that time only one additional Ada project (POWITS) was completed and had its data verified in time to be included in this study. Today, therefore, the 30-percent Ada reuse cost represents the best model available for FDD Ada simulators, but as more Ada projects are closed out, the Ada reuse cost may need to be reevaluated.

- The total project effort should be computed using the following productivities:

	Productivity (DLOC per hour)	
Type of Effort	**FORTRAN**	**Ada**
Technical and Management only	3.5	5.0
Technical, Management, and Support	3.2	4.5

- The initial effort estimate (DLOC/productivity) should be multiplied by an appropriate growth factor, which varies with the code reuse level. The recommended post-CDR growth factors are as follows:

Code Reuse Level	**Growth Factor**
Less than 70%	1.4
70% or more	1.2

- The schedule duration should be computed in calendar months, using the total project effort estimate in staff months (155 hours per staff month). The effort estimate should include the growth factor. The coefficient, COEFF, of the schedule duration formula varies with the project type and is not dependent on the development language.

$$\text{Schedule Duration} = \text{COEFF} \times (\text{Effort})^{0.3}$$

Type of Effort	COEFF AGSS	Simulator
Technical and Management only	5.0	6.7
Technical, Management, & Support	4.9	6.5

- The following percentages are still valid for planning the effort and schedule within various life-cycle phases:

Phase	Effort	Schedule
Design:	30%	35%
Code:	40%	30%
Test:	30%	35%

Section 7. Applying the Planning Models

This section explains how to apply the SEL planning models and relates them to the updated baseline models for effort and schedule.

Previous sections presented updated profiles of the behavior of typical software development projects in the FDD. These models provide a baseline against which actual project data (referred to as project actuals) can be monitored and compared. They represent what is expected to actually happen on a typical Flight Dynamics project.

Since the baseline models are based on project actuals, they naturally include project changes that result from additional information gained as projects progress throughout their life cycle, such as requirements clarification and modification, TBD definition, or launch date changes. Models such as productivity, reuse cost, and duration formulas can be applied directly when estimating a project's overall cost and schedule but the baseline models for life-cycle phase distribution of effort and schedule must be adjusted when planning the project. Therefore, the SEL planning models for life-cycle phase distribution differ from the baseline models.

At the start of a project, the project lead/manager has a limited amount of project information on which to base the plan. Thus, the plan should be based on organizational planning models that anticipate predictable changes and allow some flexibility for the project to react to unpredictable changes. The SEL planning models have been developed by applying management judgment based on many years of experience in this environment.

Cost estimation and planning is a multistep process. First, the size of the job is estimated based on project requirements, then effort and calendar time are estimated and allocated according to SEL planning models. This plan is then adjusted to handle project growth and to provide a schedule buffer.

7.1 Creating the Initial Plan

7.1.1 Estimating Effort

System size continues to be the SEL's best indicator of the amount of work to be done. Thus, effort estimates should be based on the estimated size of the system adjusted for reuse, the *developed lines of code* (DLOC), as was explained in Section 3.

To estimate DLOC, overall system size (SLOC) must first be determined based on requirements and previous similar systems and the amount of code to be reused. Based on the language, DLOC is computed as follows:

$$\text{DLOC} = \text{New SLOC} + \text{reuse cost} \times \text{Reused SLOC} \qquad (7\text{-}1)$$

where reuse cost = 0.2 for FORTRAN
= 0.3 for Ada

Then, total project effort (including support hours) is estimated as follows:

$$\text{Effort (hours)} = \text{DLOC/productivity} \qquad (7\text{-}2)$$

where productivity = 3.2 DLOC/hour for FORTRAN
= 4.5 DLOC/hour for Ada

This estimates the total amount of effort (technical, management, and support) that would be required to develop the system if (1) the size estimate is correct and (2) nothing changes.

7.1.2 Determining the Schedule

Very often the schedule for Flight Dynamics mission projects is driven by the launch date. This is often out of the project manager's control; however, the SEL schedule model can be used as a gauge to assess the level of risk resulting from the project-imposed schedule. When schedules are not predetermined, the SEL model provides a good method for determining a reasonable delivery date.

The typical and minimum project durations are determined as follows:

$$\text{Typical duration (m)} = 4.9 \times \text{Effort(sm)}^{.3} \text{ for AGSSs}$$
$$= 6.5 \times \text{Effort(sm)}^{.3} \text{ for simulators} \qquad (7\text{-}3)$$

$$\text{Minimum duration (m)} = .75 \times \text{Typical duration}$$

A planned project end date that falls between the minimum and typical durations is achievable. The closer that it falls to the minimum duration, the larger the risk.

7.1.3 Planning the Life-Cycle Phases

The planning models presented in Section 3 of Reference 6 (given here in Table 7-1) should be followed to distribute time and effort to the life-cycle phases. The design phase consists of requirements analysis, preliminary design, and detailed design. The test phase includes both system test and acceptance test.

Table 7-1. Life-Cycle Planning Model

Phase	Percent of Schedule	Percent of Effort
Design	35	30
Code	30	40
Test	35	30

The following hypothetical example should be considered:

Project FDAGSS is a FORTRAN AGSS.
Size = 99,000 DLOC
Effort = 200 staff months (30,900 hours)
Duration = 24 months (104 weeks)

Phase	Months	Staff Months
Design	8.4	60
Code	7.2	80
Test	8.4	60

Figure 7-1 shows a smooth staffing profile that reflects this distribution. Peak staffing is at 11 people. This plan is based on the amount of effort that would be required to develop the FDAGSS if (1) the size estimate is correct and (2) nothing changes.

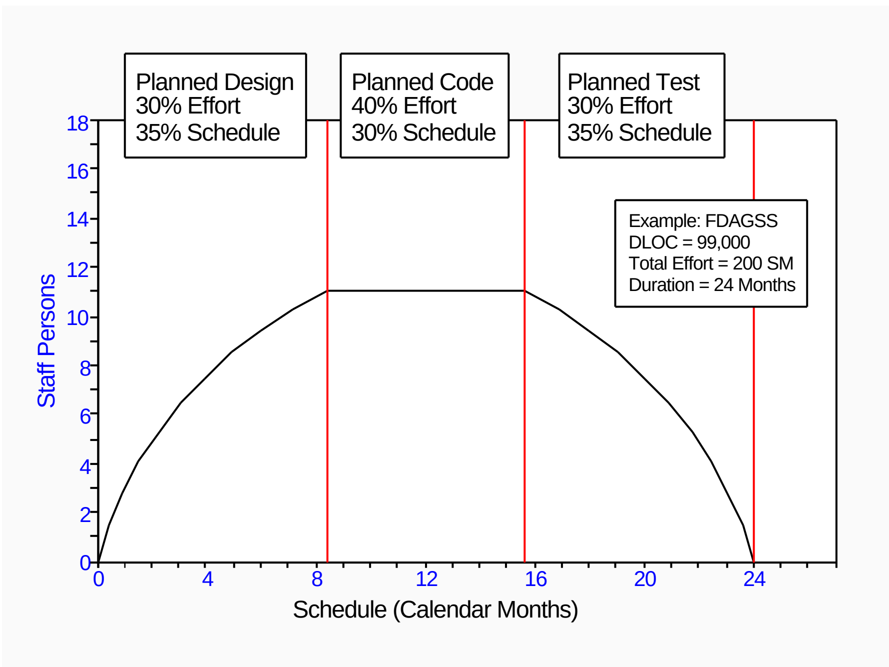


Figure 7-1. Staffing Plan Based on Initial Size

7.2 Planning for Success

7.2.1 Planning for Growth

System growth is a good measure of change. Flight Dynamics systems typically grow 40 percent over the size estimate at PDR/CDR (usually, size estimates change very little between PDR and CDR). Section 3.2.2 of this report confirms that this is still valid for flight dynamics projects with less than 70-percent reused code. Projects with higher reuse tend to grow less; based on limited SEL experience, 20-percent growth can be expected on high-reuse systems.

Although the cause of the growth varies from project to project, the amount of growth is very consistent. Thus, projects should be planned to anticipate this growth. Because size is a good indicator of effort in this environment, a 40-percent size growth typically results in an equal growth in effort, but the effect on schedule is less predictable. This is because changes in schedule are usually tied to launch dates. So, if a system grows by 40 percent and the launch does not slip, 40 percent more staff will be needed to meet the original schedule. If, however, the launch also slips, fewer staff will be added, but for a longer period of time, to meet the new delivery date.

To plan for growth, the initial effort estimate should be adjusted as follows:

$$\text{Adjusted effort} = \text{Effort} \times \text{growth factor} \qquad (7\text{-}4)$$

where growth factor = 1.4 for typical systems
= 1.2 for high (>70%) reuse systems

This effort should be distributed over the life-cycle phases as shown in Table 7-1. This increases the staffing level for each phase of the life cycle proportionately. Since the changes in the schedule cannot be predicted, this adjusted effort should be distributed over the original schedule.

Figure 7-2 shows the adjusted staffing profile based on 40-percent growth for the FDAGSS system with no schedule change. Peak staffing is now at 15.5 people.

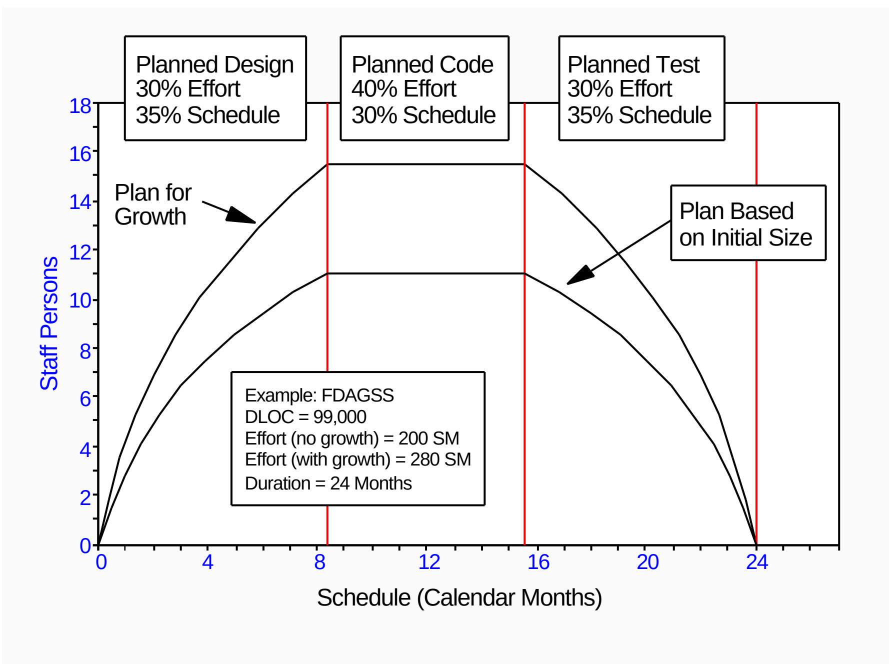


Figure 7-2. Staffing Plan That Anticipates Growth

This is a plan that will lead to success. Although system growth does not occur until after PDR and mostly after CDR, it is important to staff in anticipation of growth in the design phases. This allows the necessary staff to be fully trained when the growth occurs, resulting in higher productivity in the later life-cycle phases. If the project waits until the growth occurs to staff up, the learning curve of the additional staff will increase, rather than relieve, the burden on the original team.

It is important to remember that plans are not set in stone; they are expected to change as the project gets more and revised information. When the mission schedule changes, the software development schedule is also likely to change.

Mission delays often result from a delay in completing the spacecraft, which in turn usually causes a delay in resolving all of the TBDs in the requirements document. The software development project schedule should also be changed correspondingly, not to provide schedule relief, but to provide ample time to respond to mission changes.

An extension in the schedule will not require additional effort, but it does mean that less staff will be needed during the peak period. The staffing profile should be flattened and stretched to cover the new duration. As soon as this change is known, the project manager should adjust the plan and make corresponding staffing adjustments.

In reality, the mission schedule often changes about mid-way through the project. This results in a stretched schedule after the completion of the design phases; i.e., the end-of-design date remains fixed, while the end-of-code and testing phase dates are usually adjusted in accordance with the new schedule.

Figure 7-3 shows the relationship of the likely project actuals to the original plans created at project start for the FDAGSS. It should be noted that the actual amount of effort and time spent in the design phases ends up being a smaller percentage of the overall project when compared to the original plan. The curve for the likely actuals is based on the models for end-of-project effort and schedule as presented in the preceding sections of this report: (1) the initial total effort estimate of 200 staff months (including support hours) is multiplied by 1.4 to estimate the final total effort; (2) the total duration is computed from this total effort using Equation 7-3; (3) the effort distribution by phase follows the end-of-project percentages for moderate to low-reuse projects, as shown in Table 3-2; (4) the schedule distribution by phase follows the end-of-project percentages for moderate to low reuse projects, as shown in Table 5-6.

7.2.2 Additional Planning Considerations

In addition to staffing projects aggressively in anticipation of growth, it is also wise to set reasonably challenging schedules. A series of little unexpected problems and the effect of human nature in dealing with change typically cause a project to finish slightly later than planned. Thus, the wise manager will build a buffer into the schedule when planning; the SEL recommends a 10-percent buffer. This should be applied during initial planning and all subsequent changes to the schedule. Caution is advised; care should be taken not to reduce the project duration below the minimum (calculated based on effort adjusted for growth). Only one buffer is advised; care should be taken that only one manager applies this rule; otherwise, unrealistic goals will place the project at risk. Careful documentation of the planning process will guard against this problem.

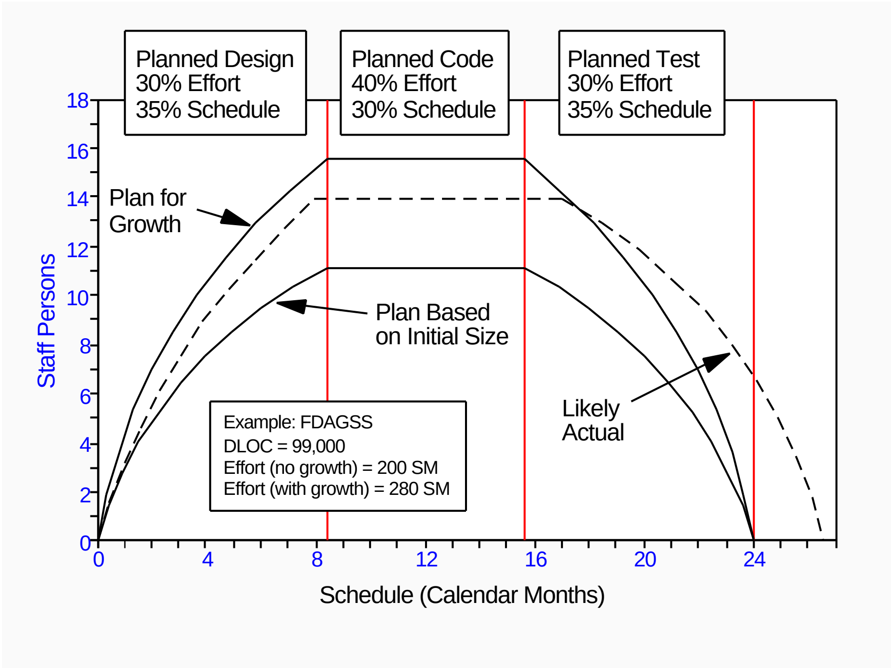


Figure 7-3. Plan Versus Actuals

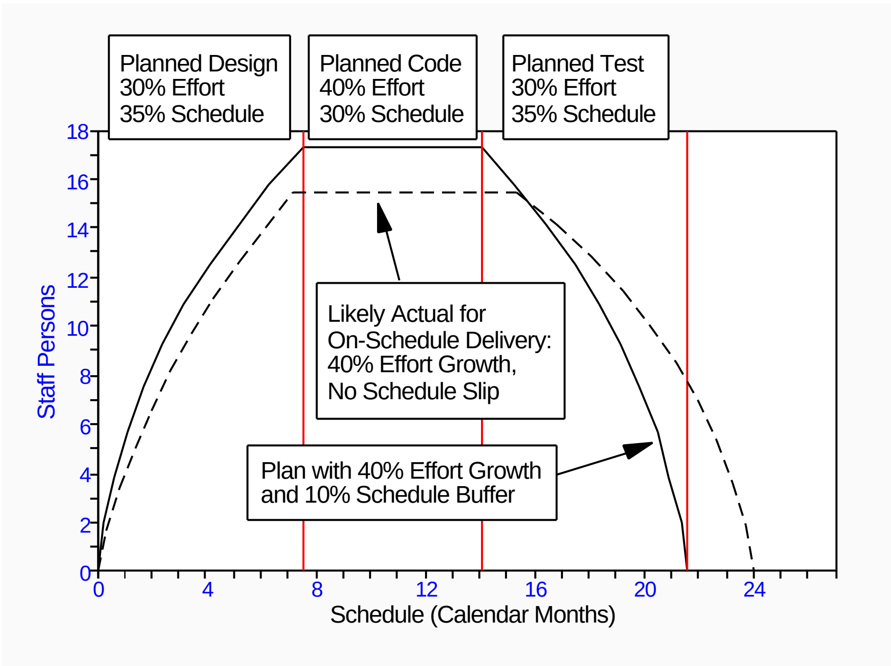


Figure 7-4. Plan for Success

Figure 7-4 shows a plan for success for project FDAGSS. It shows a staffing profile that can absorb 40-percent growth with a 2-month schedule buffer before final delivery. This project should be successful if the project can staff according to their plan in the early phases.

7.3 Reconciling Planning Models With Baseline Models

If a project manager could precisely predict the actual project end date at the beginning of the project, the SEL planning model would predict the correct staffing profile; only the phase end dates would be different.

Figure 7-5 demonstrates this, using project FDAGSS as an example. Here the FDAGSS schedule has slipped by 2-1/2 months. (The new duration, 26.6 months, would be typical for a 280-staff-month project (200 staff months + 40 percent growth) in Flight Dynamics. The dashed lines show the likely staffing profile and phase-end dates for FDAGSS (taken from Figures 7-3 and 7-4 and using the baseline effort and schedule distribution models). The solid lines show the staffing profile and phase-end dates that would have been predicted by the SEL planning model if this schedule had been known at the start of the project. The curves are remarkably similar, demonstrating the validity of the SEL planning models.

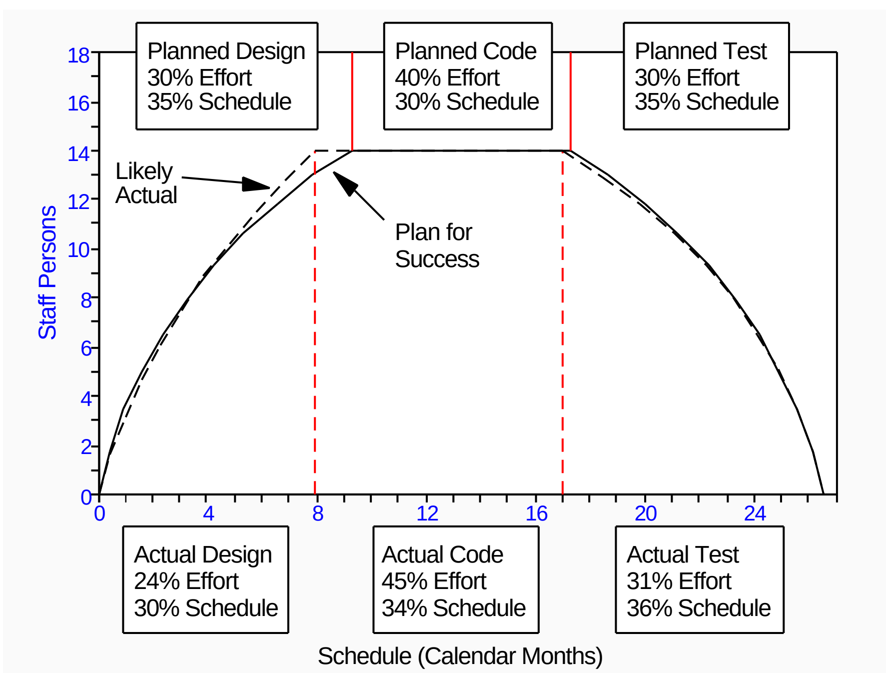


Figure 7-5. Planning Model Versus Baseline Model (Expected Actuals)

Appendix A. Summary of Cost and Schedule Models

Appendix A. Summary of Cost and Schedule Models

This appendix presents a summary of cost and schedule models that have been recommended in the FDD over the last 14 years. The models are taken from the following seven documents:

- SEL-79-002, *The Software Engineering Laboratory: Relationship Equations,* Karl Freburger and Victor Basili, May 1979

- SEL-81-205, *Recommended Approach to Software Development*, Frank McGarry, Jerry Page, Suellen Eslinger, Victor Church, and Phillip Merwarth, April 1983

- SEL-81-205, *Recommended Approach to Software Development, Revision 3,* Linda Landis, Sharon Waligora, Frank McGarry, Rose Pajerski, Mike Stark, Kevin Orlin Johnson, Donna Cover, June 1992

- SEL-83-001**,** *An Approach to Software Cost Estimation,* Frank McGarry, Jerry Page, David Card, Michael Rohleder, and Victor Church, February 1984

- SEL-84-101**,** *Manager's Handbook for Software Development, Revision 1,* Linda Landis, Frank McGarry, Sharon Waligora, et al, November 1990

- *Software Engineering Laboratory (SEL) Ada Size Study Report*, Steve Condon and Myrna Regardie, September 1992

- *SEAS System Development Methodology (SSDM) Standards and Procedures (S&P),* "Standard and Procedure 1102: Software Development Estimation," Computer Sciences Corporation, January 1993

The models are presented in the accompanying matrix. Models of the same type are grouped in the same column. Models from the same document appear in the same row. If a document does not contain a model of a particular type *"N/A"* (not applicable) appears in the field. Page references to the documents appear in brackets beneath each model. Notes and a glossary for the matrix appear at the end of the appendix.

DOCUMENT	SIZE ESTIMATES		
	SIZE FORMULAS	END OF PHASE	UNCERTAINTY
The SEL: Relationship Equations ('79)	N/A	N/A	N/A
Recommended Approach ('83)	DELOC = New ELOC + 0.2(Reused ELOC) DELOC = New ELOC + 0.2(Reused ELOC) DELOC = New ELOC + 0.2(Reused ELOC) DELOC = New ELOC + 0.2(Reused ELOC) DELOC = New ELOC + 0.2(Reused ELOC) [p. C-5]	Preproject Require. Analysis Preliminary Design Detailed Design Implementation System Test [p. C-4]	1 0.7 0.5 0.3 0.12 0.05 [p. C-4]
Cost Estimation ('84)	LOC = 7500 x (No. of Subsystems) LOC = 125 x (Number of Modules) DELOC = New ELOC + 0.2(Reused ELOC) LOC = 1.11 x (Current SLOC) [p. 3-6, 3-8]	Require. Definition Require. Analysis Preliminary Design Detailed Design Implementation System Test [p. 4-2]	1 0.75 0.5 0.3 0.12 0.05 [p. 4-2]
Manager's Handbook ('90)	SLOC = 11,000 x (No. of Subsystems) SLOC = 190 x (Number of Units) DLOC = 200 x (New Units + (0.2 x Reused Units)) SLOC = 1.26 x (Current SLOC) [p. 3-3]	Require. Analysis Preliminary Design Detailed Design Implementation System Test [p. 3-3]	0.75 0.4 0.25 0.1 0.05 [p. 3-3]
Recommended Approach ('92)	N/A	N/A	N/A
Ada Size Study ('92)	Ada DLOC = New SLOC + 0.3(Reused SLOC) FORTRAN DLOC = New SLOC + 0.2(Reused SLOC) [p. 3-3, 4-5, 5-1,2]	N/A	N/A
SSDM S&P 1102	Weighted DSI = 1.0 x (Newly Developed DSI) + W1 x (Adapted DSI) + W2 x (Converted DSI) + W3 x (Transported DSI) where W1, W2, and W3 are supplied by user.		N/A

DOCUMENT	EFFORT FORMULAS (1)	SCHEDULE DURATION FORMULAS (2)
The SEL: Relationship Equations ('79)	SMon = 0.029 x (Modules)exp(1.319) [p. 15]	Months = 5.45 x (KSLOC)exp(0.203) Months = 2.453 x (Modules)exp(0.275) Months = 5.104 x (SMon)exp(0.210) [p. 32, 41]
Recommended Approach ('83)	SMon = 8(F1)(F2)(F3)...(KDELOC)exp(1.05) [p. C-4]	No equation. Instead guidelines are provided in 2 tables on team size, phase-in, phase-out, and length of participation for team members. Guidelines depend on schedule type and project leader experience. [p. C-9]
Cost Estimation ('84)	SMon = 8.45(F1)(F2)(F3)...(KDELOC)exp(1.05) End of Phase: Formula: Require. Analysis SHr = 1850 x Subsystems Preliminary Design SHr = 30 x Modules Detailed Design SHr = 0.3 x DLOC Implementation SHr = 1.33 x (Current SHr) System Test SHr = 1.05 x (Current SHr) [p. 3-6, 4-4]	Weeks/(Staff Member) = 45 x (No. of Subsystems) Weeks/(Staff Member) = 0.75 x (No. of Modules) Week/(Staff Member) = 1.0 x Developed Module Weeks/(Staff Member) = 1.42 x Current Duration Weeks/(Staff Member) = 1.11 x Current Duration [p. 3-6]
Manager's Handbook ('90)	Require. Analysis SHr = 3000 x Subsystems Preliminary Design SHr = 52 x Units Detailed Design SHr = 0.31 x DLOC Implementation SHr = 1.43 x (Current SHr) System Test SHr = 1.11 x (Current SHr) [p. 3-3]	Weeks/(Staff Member) = 83 x (No. of Subsystems) Weeks/(Staff Member) = 1.45 x (No. of Units) Week/(Staff Member) = 0.0087 x DLOC Weeks/(Staff Member) = 1.54 x Current Duration Weeks/(Staff Member) = 1.18 x Current Duration [p. 3-3]
Recommended Approach ('92)	N/A	N/A
Ada Size Study ('92)	SHr = DLOC / Productivity [p. 3-2]	Ada: Months = 6.5 x (Staff Months)exp(0.3) FORTRAN: Months = 5.0 x (Staff Months)exp(0.3) [p. 4-3]
SSDM S&P 1102	SHr = Weighted DSI / Adjusted Productivity Adjusted Estimated Effort = SHr / 155 (3) RLC = [supplied by manager] (3)	Optimum Months = (DMC)x(RLC)exp(DMX) Minimum Months = 75% of Optimum Months where Duration Model Coefficient (DMC) and Duration Model Exponent (DMX) are supplied by the user.

DOCUMENT	PHASE	OVERALL EFFORT	ADA EFFORT	FORTRAN EFFORT	OVERALL SCHEDULE	ADA SCHEDULE	FORTRAN SCHEDULE
		EFFORT AND SCHEDULE DISTRIBUTION					
The SEL: Relationship Equations ('79)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Recommended Approach ('83)		1. Compute the relative amount of effort in each of 3 types of activities: design, code, & test. 2. Compute the staff hours of effort in each activity. 3. Compute the staff hours of effort in each of 6 life-cycle phases. [p. C-9,10]					
Cost Estimation ('84)	Require. Analysis Preliminary Design Detailed Design Implementation System Test Acceptance Test	0.06 0.08 0.16 0.45 0.2 0.05 [p. 4-8]	N/A N/A N/A N/A N/A N/A	N/A N/A N/A N/A N/A N/A	0.05 0.1 0.15 0.4 0.2 0.1 [p. 4-8]	N/A N/A N/A N/A N/A N/A	N/A N/A N/A N/A N/A N/A
Manager's Handbook ('90)	Require. Analysis Preliminary Design Detailed Design Implementation System Test Acceptance Test	0.06 0.08 0.16 0.4 0.2 0.1 [p. 3-1]	 (4) 0.32 0.29 0.19 0.2 [p. 6-4]	 (4) 0.3 0.34 0.16 0.2 [p. 6-4]	0.12 0.08 0.15 0.3 0.2 0.15 [p. 3-1]	N/A N/A N/A N/A N/A N/A	N/A N/A N/A N/A N/A N/A
Recommended Approach ('92)		N/A	N/A	N/A	N/A	N/A	N/A
Ada Size Study ('92)	Design Implementation System Test Acceptance Test	N/A N/A N/A N/A	0.26 0.42 0.17 0.15 [p. 3-12]	N/A N/A N/A N/A	N/A N/A N/A N/A	0.29 0.33 0.17 0.21 [p. 4-8]	0.33 0.29 0.19 0.19 [p. 4-8]
SSDM S&P 1102		N/A	N/A	N/A	N/A	N/A	N/A

DOCUMENT	COMPLEXITY FACTOR (F1)		TEAM EXPERIENCE FACTOR (F2)	
	DEFINITION	VALUE	DEFINITION	VALUE
The SEL: Relationship Equations ('79)	N/A	N/A	N/A	N/A
Recommended Approach ('83)	Project/Environment: Old/Old (4A) Old/New New/Old New/New [p. C-5]	Effort Factor: 0.45 0.65 0.65 1 [p. C-5]	Average Experience: 10 8 6 4 2 1 [p. C-6]	Effort Factor: 0.5 0.6 0.8 1 1.4 2.5 [p. C-6]
Cost Estimation ('84)	Old/Old (4A) Old/New New/Old New/New [p. 4-5]	0.45 0.65 0.65 1 [p. 4-5]	10 8 6 4 2 1 [p. 4-5]	0.5 0.6 0.8 1 1.4 2.5 [p. 4-5]
Manager's Handbook ('90)	Old/Old (4A) Old/New New/Old New/New [p. 3-4]	1 1.4 1.4 2.3 [p. 3-4]	10 8 6 4 2 1 [p. 3-4]	0.5 0.6 0.8 1 1.4 2.6 [p. 3-4]
Recommended Approach ('92)	N/A	N/A	N/A	N/A
Ada Size Study ('92)	N/A	N/A	N/A	N/A
SSDM S&P 1102	Risk: Lower Typical Higher	Product. Factor: 1.1 1 0.9	Risk: Lower Typical Higher	Product. Factor: 1.1 1 0.9

DOCUMENT	SCHEDULE FACTOR (F3)		MEMORY/TIMING CONSTRAINTS FACTOR (F4)		REQUIREMENTS INSTABILITY FACTOR (F5)	
	DEFINITION	VALUE	DEFINITION	VALUE	DEFINITION	VALUE
The SEL: Relationship Equations ('79)	N/A	N/A	N/A	N/A	N/A	N/A
Recommended Approach ('83)	Schedule: Fast Average Slow [p. C-6]	Effort Factor: 1.15 1 0.85 [p. C-6]	N/A	N/A	N/A	N/A
Cost Estimation ('84)	Fast Average Slow [p. 4-6]	1.15 1 0.85 [p. 4-6]	N/A	N/A	N/A	N/A
Manager's Handbook ('90)	N/A	N/A	N/A	N/A	N/A	N/A
Recommended Approach ('92)	N/A	N/A	N/A	N/A	N/A	N/A
Ada Size Study ('92)	N/A	N/A	N/A	N/A	N/A	N/A
SSDM S&P 1102	N/A	N/A	Risk: Lower Typical Higher	Product. Factor: 1.1 1 0.9	Risk: Lower Typical Higher	Product. Factor: 1.1 1 0.9

DOCUMENT	ENGINEERING METHODS FACTOR (F6)		DEVELOPMENT TOOLS FACTOR (F7)		DATA VOLUME FACTOR (F8)	
	DEFINITION	VALUE	DEFINITION	VALUE	DEFINITION	VALUE
The SEL: Relationship Equations ('79)	N/A	N/A	N/A	N/A	N/A	N/A
Recommended Approach ('83)	N/A	N/A	N/A	N/A	N/A	N/A
Cost Estimation ('84)	N/A	N/A	N/A	N/A	N/A	N/A
Manager's Handbook ('90)	N/A	N/A	N/A	N/A	N/A	N/A
Recommended Approach ('92)	N/A	N/A	N/A	N/A	N/A	N/A
Ada Size Study ('92)	N/A	N/A	N/A	N/A	N/A	N/A
SSDM S&P 1102	Risk: Lower Typical Higher	Product. Factor: 1.1 1 0.9	Risk: Lower Typical Higher	Product. Factor: 1.1 1 0.9	Risk: Lower Typical Higher	Product. Factor: 1.1 1 0.9

DOCUMENT	WORK RATE GUIDE (KDLOEC/WEEK)			
	PROJECT/ENVIRONS	FAST SCHEDULE	AVERAGE SCHEDULE	SLOW SCHEDULE
The SEL: Relationship Equations ('79)	N/A	N/A	N/A	N/A
Recommended Approach ('83)	N/A	N/A	N/A	N/A
Cost Estimation ('84)	Old/Old (4A) Old/New New/Old New/New [p. 4-7]	>0.24 >0.17 >0.17 >0.11 [p. 4-7]	0.24-0.16 0.17-0.10 0.17-0.10 0.11-0.07 [p. 4-7]	<0.16 <0.10 <0.10 <0.07 [p. 4-7]
Manager's Handbook ('90)	N/A	N/A	N/A	N/A
Recommended Approach ('92)	N/A	N/A	N/A	N/A
Ada Size Study ('92)	N/A	N/A	N/A	N/A
SSDM S&P 1102	N/A	N/A	N/A	N/A

DOCUMENT	TEAM SIZE GUIDE MIN. LEADER EXPERIENCE VS. MAX. TEAM SIZE				STAFFING GUIDELINE (PCT. OF SENIOR PERSONNEL AND ANALYSTS) PROJECT/ENVIRON.	SENIOR(6)	ANALYSTS
The SEL: Relationship Equations ('79)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Recommended Approach ('83)	Years Experience: Applicable: Organization: As Leader: Maximum Team (5): [p. C-12]	 6 4 3 5-9	 5 3 1 3-6	 4 2 0 1-3	Old/Old (4A) Old/New New/Old New/New [p. C-15]	25-33% 33-50% 33-50% 50-67%	25-33% 25-33% 33-50% 33-50%
Cost Estimation ('84)	Years Experience: Applicable: Organization: As Leader: Maximum Team (5): [p. 4-9]	 6 4 3 5-9	 5 3 1 3-6	 4 2 0 1-3	Old/Old (4A) Old/New New/Old New/New [p. 4-10]	25-33% 33-50% 33-50% 50-67%	25-33% 25-33% 33-50% 33-50%
Manager's Handbook ('90)	Years Experience: Applicable: Organization: As Leader: Maximum Team (5): [p. 3-5]	 6 4 3 5-9	 5 3 1 2-6	 4 2 0 1-3	Old/Old (4A) Old/New New/Old New/New [p. 3-5]	25-33% 33-50% 33-50% 50-67%	25-33% 25-33% 33-50% 33-50%
Recommended Approach ('92)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Ada Size Study ('92)	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SSDM S&P 1102	N/A	N/A	N/A	N/A	N/A	N/A	N/A

DOCUMENT	ANALYSIS SUPPORT		CPU HOURS	COMPUTER RUNS
	SUPPORT TYPE	ADDED COST		
The SEL: Relationship Equations ('79)	N/A	N/A	N/A	N/A
Recommended Approach ('83)	N/A	N/A	N/A	N/A
Cost Estimation ('84)	Requirements spec.: Data simulation: Acceptance test: Requirements clarif.:	25% 5% 5% 10% [p. 3-14]	CPU=0.009xDLOC [p. 3-10]	N/A
Manager's Handbook ('90)	N/A	N/A	CPU=0.0008xSLOC [p. 3-5]	Runs = 0.29 x SLOC [p. 3-5]
Recommended Approach ('92)	N/A	N/A	N/A	N/A
Ada Size Study ('92)	N/A	N/A	N/A	N/A
SSDM S&P 1102	N/A	N/A	N/A	N/A

DOCUMENT	DOCUMENTATION			
	TOTAL PAGES	DOCUMENTS	% OF TOTAL PAGES	ADDED COST (% OF BASIC DEV.COST)
The SEL: Relationship Equations ('79)	Pages = 34 x (Modules)exp(0.662) [p. 27]	N/A	N/A	N/A
Recommended Approach ('83)	N/A	N/A	N/A	N/A
Cost Estimation ('84)	Pages = 0.04 x DLOC [p. 3-11]	Design description: Test plans: User documents: Component prologs: Devel./Management Plan:	33% 7% 41% 16% 3% [p. 3-12]	No user documents: 0% Informal documents: 5% Formal documents: 16% [p. 3-12]
Manager's Handbook ('90)	Pages = 120 + (0.026 x SLOC) Cost = 4 Staff Hrs./page [p. 3-7]	N/A	N/A	N/A
Recommended Approach ('92)	N/A	N/A	N/A	N/A
Ada Size Study ('92)	N/A	N/A	N/A	N/A
SSDM S&P 1102	N/A	N/A	N/A	N/A

DOCUMENT	EARLY ESTIMATING PARAMATERS			
	SCALE	SIZE	COST	SCHEDULE
The SEL: Relationship Equations ('79)	N/A	N/A	N/A	N/A
Recommended Approach ('83)	N/A	N/A	N/A	N/A
Cost Estimation ('84)	Subsystem Module Devel. Module (8) DLOC	7500 LOC(6A) 125 LOC (7) [p. 3-6]	1850 HRS (6A) 30 HRS (7) 0.3 HRS (9) [p. 3-6]	45 Wks/SS/Person (6A) 0.75 Wks/Module/Person (7) 1.0 Wks/DModule/Person (7) [p. 3-6]
Manager's Handbook ('90)	Subsystem Unit Devel. Unit (8) DLOC	11,000 SLOC(6A) 190 SLOC (7) 200 DLOC (9) [p. 3-3]	3000 HRS (6A) 52 HRS (7) 0.31 HRS (9) [p. 3-3]	83 Wks/SS/Person (6A) 1.45 Wks/Unit/Person (7) [p. 3-3]
Recommended Approach ('92)	N/A	N/A	N/A	N/A
Ada Size Study ('92)	N/A	N/A	N/A	N/A
SSDM S&P 1102	N/A	N/A	N/A	N/A

DOCUMENT	COST OF REHOSTING SOFTWARE					
	SYSTEM RELATIONSHIP	RELATIVE COST (10) FORTRAN	RELATIVE COST (10) ADA	TESTING EFFORTS (11) FORTRAN	TESTING EFFORTS (11) ADA	NEW CODE
The SEL: Relationship Equations ('79)	N/A	N/A	N/A	N/A	N/A	N/A
Recommended Approach ('83)	N/A	N/A	N/A	N/A	N/A	N/A
Cost Estimation ('84)	COMPATIBLE SIMILAR DISSIMILAR	15-21% 22-32% 33-50% [p. 3-15]	N/A N/A N/A	67-70% 61-66% 55-60% [p. 3-15]	N/A N/A N/A	0-3% 4-14% 15-32% [p. 3-15]
Manager's Handbook ('90)	COMPATIBLE SIMILAR DISSIMILAR	10-16% 15-18% 20-40% [p. 3-7]	5-11 10-15 18-30	55-70% 45-55% 40-50% [p. 3-7]	36-40 30-35 25-30	0-3% 4-14% 15-32% [p. 3-7]
Recommended Approach ('92)	N/A	N/A	N/A	N/A	N/A	N/A
Ada Size Study ('92)	N/A	N/A	N/A	N/A	N/A	N/A
SSDM S&P 1102	N/A	N/A	N/A	N/A	N/A	N/A

Notes:

----

(1) Effort denoted is either total Staff Hours (SHr) or total Staff Months (SMon).

(2) Duration denoted is either weeks per staff member or total calendar months.

(3) Represents total estimated labor in staff months. This may vary from the Recommended Labor Commitment (RLC), used by the project manager to compute project duration.

(4) Included in Detailed Design percentage

(4A) The project type (e.g., orbit determination, simulator) is "OLD" when the organization has more than 2 years experience with it.
The environment type (e.g., IBM 4341, VAX 8810) is "OLD" when the organization has more than 2 years experience with it.

(5) Team size, not counting team leader

(6) More than 5 years experience in development related activities

(6A) Estimate at end of requirements analysis

(7) Estimate at end of preliminary design

(8) Number of developed units = N + 0.2R,
Where N = number of New and Extensively Modified units
R = Number of Slightly Modified and Verbatim units

(9) Estimate at end of detailed design

(10) Percent of original development costs

(11) Percent of total rehosting cost

Glossary:

Adapted Code = Reused code requiring changes to 25% or more of the lines, also known as 'Extensively Modified' Code

Adjusted Estimated Effort: estimated staff months to complete the project (SSDM S&P 1102)

Compatible = Systems designed to be plug compatible (e.g., IBM S/360 and 4341).

Converted Code = Reused code requiring changes to less than 25% of the lines, also known as 'Slightly Modified' Code

DELOC = Developed Executable Lines of Code

Dissimilar = Systems with differences in most characteristics of architecture and organization (e.g., IBM S/360 and PDP 11/70).

DLOC = Developed Lines of Code

DMC = Duration Model Coefficient

DMX = Duration Model Exponent

DSI = Delivered Source Instructions

E = Effort (in total staff hours, unless otherwise specified)

ELOC = Executable Lines of Code

KDELOC = 1000s of Developed Executable Lines of Code

LOC = Lines of Code

New SLOC = SLOC of New and Extensively Modified units

Reused SLOC = SLOC of Slightly Modified and Verbatim units

RLC = Recommended Labor Commitment (in staff months), a figure used in SSDM to compute project duration. The RLC may differ from the Adjusted Estimated Effort.

SHr = total Staff Hours of effort

Similar = Systems (e.g., IBM 4341 and VAX 8810) with some key architectural characteristics, such as word size.

SLOC = Source Lines of Code (includes blank lines)

SMon = Staff Months of effort

SS = Subsystems

Transported Code = Reused code requiring no changes, also known as 'Verbatim' Code

Appendix B. Sample Subjective Evaluation Form

SUBJECTIVE EVALUATION FORM

Name: ____________________

Project: ____________________ Date: ____________________

IIndicate response by circling the corresponding numeric ranking.

I. PROBLEM CHARACTERISTICS

1. Assess the intrinsic difficulty or complexity of the problem that was addressed by the software development.

1	2	3	4	5
Easy		Average		Difficult

2. How tight were schedule constraints on project?

1	2	3	4	5
Loose		Average		Tight

3. How stable were requirements over development period?

1	2	3	4	5
Loose		Average		High

4. Assess the overall quality of the requirements specification documents, including their clarity, accuracy, consistency, and completeness.

1	2	3	4	5
Low		Average		High

5. How extensive were documentation requirements?

1	2	3	4	5
Low		Average		High

6. How rigorous were formal review requirements?

1	2	3	4	5
Low		Average		High

II. PERSONNEL CHARACTERISTICS: TECHNICAL STAFF

7. Assess overall quality and ability of development team.

1	2	3	4	5
Low		Average		High

8. How would you characterize the development team's experience and familiarity with the application area of the project?

1	2	3	4	5
Low		Average		High

9. Assess the development team's experience and familiarity with the development environment (hardware and support software).

1	2	3	4	5
Low		Average		High

10. How stable was the composition of the development team over the duration of the project?

1	2	3	4	5
Loose		Average		High

FOR LIBRARIAN'S USE ONLY

Number: ____________________ Entered by: ____________________

Date: ____________________ Checked by: ____________________

6

SUBJECTIVE EVALUATION FORM

III. PERSONNEL CHARACTERISTICS: TECHNICAL MANAGEMENT

11. Assess the overall performance of project management.

1	2	3	4	5
Low		Average		High

12. Assess project management's experience and familiarity with the application.

1	2	3	4	5
Low		Average		High

13. How stable was project management during the project?

1	2	3	4	5
Low		Average		High

14. What degree of disciplined project planning was used?

1	2	3	4	5
Low		Average		High

15. To what degree were project plans followed?

1	2	3	4	5
Low		Average		High

IV. PROCESS CHARACTERISTICS

16. To what extent did the development team use modern programming practices (PDL, top-down development, structured programming, and code reading)?

1	2	3	4	5
Low		Average		High

17. To what extent did the development team use well-defined or disciplined procedures to record specification modifications, requirements questions and answers, and interface agreements?

1	2	3	4	5
Low		Average		High

18. To what extent did the development team use a well-defined or disciplined requirements analysis methodology?

1	2	3	4	5
Low		Average		High

19. To what extent did the development team use a well-defined or disciplined design methodology?

1	2	3	4	5
Low		Average		High

20. To what extent did the development team use a well-defined or disciplined testing methodology?

1	2	3	4	5
Low		Average		High

IV. PROCESS CHARACTERISTICS

21. What software tools were used by the development team? Check all that apply from the list that follows and identify any other tools that were used but are not listed.

- ☐ Compiler
- ☐ Linker
- ☐ Editor
- ☐ Graphic display builder
- ☐ Requirements language processor
- ☐ Structured analysis support tool
- ☐ PDL processor
- ☐ ISPF
- ☐ SAP
- ☐ CAT
- ☐ PANVALET
- ☐ Test coverage tool
- ☐ Interface checker (RXVP80, etc.)
- ☐ Language-sensitive editor
- ☐ Symbolic debugger
- ☐ Configuration Management Tool (CMS, etc.)
- ☐ Others (identify by name and function)

22. To what extent did the development team prepare and follow test plans?

1	2	3	4	5
Low		Average		High

SUBJECTIVE EVALUATION FORM

IV. PROCESS CHARACTERISTICS (CONT'D)

23. To what extent did the development team use well-defined and disciplined quality assurance procedures (reviews, inspections, and walkthroughs)?

1	2	3	4	5
Low		Average		High

24. To what extent did development team use well-defined or disciplined configuration management procedures?

1	2	3	4	5
Low		Average		High

V. ENVIRONMENT CHARACTERISTICS

25. How would you characterize the development team's degree of access to the development system?

1	2	3	4	5
Low		Average		High

26. What was the ratio of programmers to terminals?

1	2	3	4	5
8:1	4:1	2:1	1:1	1:2

27. To what degree was the development team constrained by the size of main memory or direct-access storage available on the development system?

1	2	3	4	5
Low		Average		High

28. Assess the system response time: were the turnaround times experienced by the team satisfactory in light of the size and nature of the jobs?

1	2	3	4	5
Poor		Average		Very Good

29. How stable was the hardware and system support software (including language processors) during the project?

1	2	3	4	5
Low		Average		High

30. Assess the effectiveness of the software tools.

1	2	3	4	5
Low		Average		High

VI. PRODUCT CHARACTERISTICS

31. To what degree does the delivered software provide the capabilities specified in the requirements?

1	2	3	4	5
Low		Average		High

32. Assess the quality of the delivered software product.

1	2	3	4	5
Low		Average		High

33. Assess the quality of the design that is present in the software product.

1	2	3	4	5
Low		Average		High

34. Assess the quality and completeness of the delivered system documentation.

1	2	3	4	5
Low		Average		High

35. To what degree were software products delivered on time?

1	2	3	4	5
Low		Average		High

36. Assess smoothness or relative ease of acceptance testing.

1	2	3	4	5
Low		Average		High

6

Appendix C. Effort and Schedule Detailed Distribution

Appendix C. Effort and Schedule Detailed Distribution

Appendix C captures some of the detailed work performed during the Cost and Schedule Estimation Study in the area of distributing effort and schedule by life-cycle phase and distributing effort by activity. This portion of the analysis was an attempt to group the distributions and search for trends by examining the projects in the highest and lowest, ranges along with their project characteristics such as reuse percent or language type. Projects, subsequent to and including COBEDS, listed in Tables C-1 through C-3 were examined to determine the five projects with the highest and the lowest percentages of effort or schedule for a particular life-cycle phase. The distribution of effort by activity was also examined in a similar manner. Tables C-4 through C-9 show the results of these analyses. There tends to be high variability among projects as to the distribution of effort and schedule by phase as well as the distribution of effort by activity.

Two patterns are noted here, but the study attributed no conclusions or significance to these patterns. The first is that projects with the lowest percentage of coding activity tend to be high-reuse projects. The second is that the projects with the highest percentages of coding activity are FORTRAN projects, but, at the same time, all are low-reuse projects.

Because of the time limitations of the study, the analysis in this area was limited; the data are archived here to provide a basis for future analysis.

Table C-1. Effort Distribution by Phase

	DESIGN %	CODE %	ST %	AT %
PAS	18.0%	57.1%	15.4%	9.6%
ISEEB	21.9%	57.1%	12.8%	8.2%
AEM	19.4%	50.4%	16.7%	13.6%
SEASAT	25.5%	49.4%	10.9%	14.3%
SMM	29.6%	41.7%	16.7%	12.0%
MAGSAT	21.9%	38.7%	19.1%	20.3%
FOXPRO	19.1%	28.4%	21.7%	30.9%
DEA	16.4%	49.9%	10.3%	23.5%
DEB	20.0%	49.7%	15.1%	15.2%
DESIM	35.5%	44.0%	10.7%	9.7%
ERBS	21.9%	50.9%	16.0%	11.2%
DERBY	29.1%	45.7%	11.2%	14.0%
COBEDS	32.8%	29.2%	32.6%	5.4%
ASP	22.7%	42.8%	18.5%	16.0%
GROSIM	20.5%	43.5%	27.4%	8.6%
COBSIM	25.4%	42.3%	22.4%	9.9%
COBEAGSS	23.1%	38.2%	23.3%	15.4%
GOADA	27.7%	41.8%	24.2%	6.3%
GOFOR	15.5%	31.5%	39.9%	13.1%
GOESAGSS	18.7%	54.4%	16.9%	9.9%
GOESIM	28.5%	44.2%	9.4%	18.0%
UARSAGSS	19.4%	49.6%	17.6%	13.4%
UARSDSIM	18.0%	46.0%	8.5%	27.4%
UARSTELS	24.6%	39.4%	15.2%	20.8%
EUVEAGSS	14.0%	48.3%	22.2%	15.5%
EUVETELS	26.1%	40.6%	13.2%	20.1%
EUVEDSIM	21.5%	44.7%	23.8%	10.0%
POWITS	13.6%	47.0%	11.9%	27.5%
SAMPEXTS	48.1%	18.0%	18.3%	15.5%
SAMPEX	26.4%	16.3%	36.8%	20.5%

Table C-2. Effort Distribution by Activity

	DESIGN %	CODE %	TEST %	OTHER %
PAS	6.7%	25.2%	13.6%	54.5%
ISEEB	16.2%	22.7%	10.0%	51.1%
AEM	19.7%	25.9%	15.9%	38.5%
SEASAT	14.2%	26.7%	14.0%	45.1%
SMM	26.4%	27.1%	14.3%	32.2%
MAGSAT	25.4%	25.3%	18.2%	31.0%
FOXPRO	32.2%	27.1%	17.1%	23.7%
DEA	15.1%	18.8%	24.8%	41.2%
DEB	20.0%	21.8%	16.3%	42.0%
DESIM	28.9%	23.4%	14.3%	33.5%
ERBS	18.3%	29.2%	16.7%	35.8%
DERBY	26.5%	13.1%	14.9%	45.5%
COBEDS	24.4%	20.8%	16.1%	38.7%
ASP	14.6%	21.2%	23.7%	40.4%
GROSIM	22.0%	32.6%	15.4%	30.0%
COBSIM	22.5%	31.2%	14.4%	31.9%
COBEAGSS	24.1%	22.2%	27.7%	26.1%
GOADA	19.2%	27.8%	23.7%	29.3%
GOFOR	11.7%	18.5%	39.2%	30.7%
GOESAGSS	25.3%	31.8%	24.6%	18.3%
GOESIM	19.2%	22.8%	23.6%	34.4%
UARSAGSS	24.0%	29.1%	28.8%	18.1%
UARSDSIM	18.1%	33.9%	27.4%	20.6%
UARSTELS	19.3%	27.5%	33.3%	19.9%
EUVEAGSS	21.5%	25.0%	31.3%	22.2%
EUVETELS	15.2%	16.8%	26.2%	41.8%
EUVEDSIM	21.0%	30.0%	21.4%	27.6%
POWITS	9.2%	18.9%	40.8%	31.1%
SAMPEXTS	16.7%	16.6%	26.8%	39.9%
SAMPEX	14.5%	6.4%	30.5%	48.6%

Table C-3. Schedule Distribution by Phase

	DESIGN %	CODE %	ST %	AT %
PAS	27.5%	46.4%	13.0%	13.0%
ISEEB	42.0%	42.0%	8.0%	8.0%
AEM	28.1%	45.6%	15.8%	10.5%
SEASAT	31.5%	44.4%	9.3%	14.8%
SMM	31.6%	31.6%	11.8%	25.0%
MAGSAT	30.6%	38.7%	14.5%	16.1%
FOXPRO	44.4%	27.8%	11.1%	16.7%
DEA	36.0%	47.2%	4.5%	12.4%
DEB	38.6%	37.3%	12.0%	12.0%
DESIM	50.0%	35.7%	7.1%	7.1%
ERBS	43.3%	34.0%	12.4%	10.3%
DERBY	36.1%	31.9%	11.1%	20.8%
COBEDS	34.3%	22.9%	31.4%	11.4%
ASP	29.9%	31.0%	14.9%	24.1%
GROSIM	35.0%	39.0%	17.0%	9.0%
COBSIM	28.0%	40.2%	18.3%	13.4%
COBEAGSS	26.7%	26.7%	20.7%	25.9%
GOADA	27.5%	28.9%	30.9%	12.8%
GOFOR	25.2%	27.7%	31.9%	15.1%
GOESAGSS	27.0%	38.3%	16.5%	18.3%
GOESIM	34.3%	29.3%	8.1%	28.3%
UARSAGSS	30.6%	36.1%	16.3%	17.0%
UARSDSIM	25.8%	45.3%	7.0%	21.9%
UARSTELS	31.9%	29.8%	10.6%	27.7%
EUVEAGSS	37.3%	33.3%	14.7%	14.7%
EUVETELS	26.5%	42.2%	12.0%	19.3%
EUVEDSIM	27.3%	35.5%	22.3%	14.9%
POWITS	26.1%	31.5%	8.1%	34.2%
SAMPEXTS	47.9%	8.3%	16.7%	27.1%
SAMPEX	45.9%	14.1%	22.4%	17.6%

Table C-4. Analysis of Activity Effort Distribution—Highest Percentages

FIVE PROJECTS WITH HIGHEST *DESIGN* PERCENTAGES

PROJECTS	DESIGN %	REUSE %	TYPE	LANG
GOESAGSS	25.3 %	12.1 %	AGSS	F
COBEDS	24.4 %	26.9 %	DS	F
COBEAGSS	24.1 %	12.1 %	AGSS	F
UARSAGSS	24.0 %	11.0 %[1]	AGSS	F
COBSIM	22.5 %	10.7 %	TS	F

FIVE PROJECTS WITH HIGHEST *CODE* PERCENTAGES

PROJECTS	CODE %	REUSE %	TYPE	LANG
UARSDSIM	33.9 %	23.6 %	DS	F
GROSIM	32.6 %	17.9 %	TS	F
GOESAGSS	31.8 %	12.1 %	AGSS	F
COBSIM	31.2 %	10.7 %	TS	F
UARSAGSS	29.1 %	11.0 %[1]	AGSS	F

FIVE PROJECTS WITH HIGHEST *TEST* PERCENTAGES

PROJECTS	TEST %	REUSE %	TYPE	LANG
POWITS	40.8 %	69.2 %	TS	A
GOFOR	39.2 %	32.4 %	DS	F
UARSTELS	33.3 %	34.8 %	TS	A
EUVEAGSS	31.3 %	78.0 %[1]	AGSS	F
SAMPEX	30.5 %	92.1 %	AGSS	F

FIVE PROJECTS WITH HIGHEST *OTHER* PERCENTAGES

PROJECTS	OTHER %	REUSE %	TYPE	LANG
SAMPEX	48.6 %	92.1 %	AGSS	F
EUVETELS	41.8 %	96.2 %	TS	A
ASP	40.4 %	12.9 %	AGSS	F
SAMPEXTS	39.9 %	94.6 %	TS	A
COBEDS	38.7 %	26.9 %	DS	F

[1] Reuse percent excludes ACME portion of project.

Table C-5. Analysis of Activity Effort Distribution—Lowest Percentages

FIVE PROJECTS WITH LOWEST *DESIGN* PERCENTAGES

PROJECTS	DESIGN %	REUSE %	TYPE	LANG
POWITS	9.2 %	69.2 %	TS	A
GOFOR	11.7 %	32.4 %	DS	F
SAMPEX	14.5 %	92.1 %	AGSS	F
ASP	14.6 %	12.9 %	AGSS	F
EUVETELS	15.2 %	96.2 %	TS	A

FIVE PROJECTS WITH LOWEST *CODE* PERCENTAGES

PROJECTS	CODE %	REUSE %	TYPE	LANG
SAMPEX	6.4 %	92.1 %	AGSS	F
SAMPEXTS	16.6 %	94.6 %	TS	A
EUVETELS	16.8 %	96.2 %	TS	A
GOFOR	18.5 %	32.4 %	DS	F
POWITS	18.9 %	69.2 %	TS	A

FIVE PROJECTS WITH LOWEST *TEST* PERCENTAGES

PROJECTS	TEST %	REUSE %	TYPE	LANG
COBSIM	14.4 %	10.7 %	TS	F
GROSIM	15.4 %	17.9 %	TS	F
COBEDS	16.1 %	26.9 %	DS	F
GOESIM	23.6 %	28.8 %	TS	A
GOADA	23.7 %	28.5 %	DS	A

FIVE PROJECTS WITH LOWEST *OTHER* PERCENTAGES

PROJECTS	OTHER %	REUSE %	TYPE	LANG
UARSAGSS	18.1 %	11.0 %[1]	AGSS	F
GOESAGSS	18.3 %	12.1 %	AGSS	F
UARSTELS	19.9 %	34.8 %	TS	A
UARSDSIM	20.6 %	23.6 %	DS	F
EUVEAGSS	22.2 %	78.0 %[1]	AGSS	F

[1] Reuse percent excludes ACME portion of project.

Table C-6. Analysis of Phase Effort Distribution—Highest Percentages

FIVE PROJECTS WITH HIGHEST *DESIGN* PERCENTAGES

PROJECTS	DESIGN %	REUSE %	TYPE	LANG
SAMPEXTS	48.1 %	94.6 %	TS	A
COBEDS	32.8 %	26.9 %	DS	F
GOESIM	28.5 %	28.8 %	TS	A
GOADA	27.7 %	28.5 %	DS	A
SAMPEX	26.4 %	92.1 %	AGSS	F

FIVE PROJECTS WITH HIGHEST *CODE* PERCENTAGES

PROJECTS	CODE %	REUSE %	TYPE	LANG
GOESAGSS	54.4 %	12.1 %	AGSS	F
UARSAGSS	49.6 %	11.0 %[1]	AGSS	F
EUVEAGSS	48.3 %	78.0 %[1]	AGSS	F
POWITS	47.0 %	69.2 %	TS	A
UARSDSIM	46.0 %	23.6 %	DS	F

FIVE PROJECTS WITH HIGHEST *ST* PERCENTAGES

PROJECTS	ST %	REUSE %	TYPE	LANG
GOFOR	39.9 %	32.4 %	DS	F
SAMPEX	36.8 %	92.1 %	AGSS	F
COBEDS	32.6 %	26.9 %	DS	F
GROSIM	27.4 %	17.9 %	TS	F
GOADA	24.2 %	28.5 %	DS	A

FIVE PROJECTS WITH HIGHEST *AT* PERCENTAGES

PROJECTS	AT %	REUSE %	TYPE	LANG
POWITS	27.5 %	69.2 %	TS	A
UARSDSIM	27.4 %	23.6 %	DS	F
UARSTELS	20.8 %	34.8 %	TS	A
SAMPEX	20.5 %	92.1 %	AGSS	F
EUVETELS	20.1 %	96.2 %	TS	A

[1] Reuse percent excludes ACME portion of project.

Table C-7. Analysis of Phase Effort Distribution—Lowest Percentages

FIVE PROJECTS WITH LOWEST *DESIGN* PERCENTAGES

PROJECTS	DESIGN %	REUSE %	TYPE	LANG
POWITS	13.6 %	69.2 %	TS	A
EUVEAGSS	14.0 %	78.0 %[1]	AGSS	F
GOFOR	15.5 %	32.4 %	DS	F
UARSDSIM	18.0 %	23.6 %	DS	F
GOESAGSS	18.7 %	12.1 %	AGSS	F

FIVE PROJECTS WITH LOWEST *CODE* PERCENTAGES

PROJECTS	CODE %	REUSE %	TYPE	LANG
SAMPEX	16.3 %	92.1 %	AGSS	F
SAMPEXTS	18.0 %	94.6 %	TS	A
COBEDS	29.2 %	26.9 %	DS	F
GOFOR	31.5 %	32.4 %	DS	F
COBEAGSS	38.2 %	12.1 %	AGSS	F

FIVE PROJECTS WITH LOWEST *ST* PERCENTAGES

PROJECTS	ST %	REUSE %	TYPE	LANG
UARSDSIM	8.6 %	23.6 %	DS	F
GOESIM	9.4 %	28.8 %	TS	A
POWITS	11.9 %	69.2 %	TS	A
EUVETELS	13.2 %	96.2 %	TS	A
UARSTELS	15.2 %	34.8 %	TS	A

FIVE PROJECTS WITH LOWEST *AT* PERCENTAGES

PROJECTS	AT %	REUSE %	TYPE	LANG
COBEDS	5.4 %	26.9 %	DS	F
GOADA	6.3 %	28.5 %	DS	A
GROSIM	8.6 %	17.9 %	TS	F
COBSIM	9.9 %	10.7 %	TS	F
GOESAGSS	9.9 %	12.1 %	AGSS	F

[1] Reuse percent excludes ACME portion of project.

Table C-8. Analysis of Schedule Distribution—Highest Percentages

FIVE PROJECTS WITH HIGHEST *DESIGN* PERCENTAGES

PROJECTS	DESIGN %	REUSE %	TYPE	LANG
SAMPEXTS	47.9 %	94.6 %	TS	A
SAMPEX	45.9 %	92.1 %	AGSS	F
EUVEAGSS	37.3 %	78.0 %[1]	AGSS	F
GROSIM	35.0 %	17.9 %	TS	F
GOESIM	34.3 %	28.8 %	TS	A

FIVE PROJECTS WITH HIGHEST *CODE* PERCENTAGES

PROJECTS	CODE %	REUSE %	TYPE	LANG
UARSDSIM	45.3 %	23.6 %	DS	F
EUVETELS	42.2 %	96.2 %	TS	A
COBSIM	40.2 %	10.7 %	TS	F
GROSIM	39.0 %	17.9 %	TS	F
GOESAGSS	38.3 %	12.1 %	AGSS	F

FIVE PROJECTS WITH HIGHEST *ST* PERCENTAGES

PROJECTS	ST %	REUSE %	TYPE	LANG
GOFOR	31.9 %	32.4 %	DS	F
COBEDS	31.4 %	26.9 %	DS	F
GOADA	30.9 %	28.5 %	DS	A
SAMPEX	22.4 %	92.1 %	AGSS	F
COBEAGSS	20.7 %	12.1 %	AGSS	F

FIVE PROJECTS WITH HIGHEST *AT* PERCENTAGES

PROJECTS	AT %	REUSE %	TYPE	LANG
POWITS	34.2 %	69.2 %	TS	A
GOESIM	28.3 %	28.8 %	TS	A
UARSTELS	27.7 %	34.8 %	TS	A
SAMPEXTS	27.1 %	94.6 %	TS	A
COBEAGSS	25.9 %	12.1 %	AGSS	F

[1] Reuse percent excludes ACME portion of project.

Table C-9. Analysis of Schedule Distribution—Lowest Percentages

FIVE PROJECTS WITH LOWEST *DESIGN* PERCENTAGES

PROJECTS	DESIGN %	REUSE %	TYPE	LANG
GOFOR	25.2 %	32.4 %	DS	F
UARSDSIM	25.8 %	23.6 %	DS	F
POWITS	26.1 %	69.2 %	TS	A
EUVETELS	26.5 %	96.2 %	TS	A
COBEAGSS	26.7 %	12.1 %	AGSS	F

FIVE PROJECTS WITH LOWEST *CODE* PERCENTAGES

PROJECTS	CODE %	REUSE %	TYPE	LANG
SAMPEXTS	8.3 %	94.6 %	TS	A
SAMPEX	14.1 %	92.1 %	AGSS	F
COBEDS	22.9 %	26.9 %	DS	F
COBEAGSS	26.7 %	12.1 %	AGSS	F
GOFOR	27.7 %	32.4 %	DS	F

FIVE PROJECTS WITH LOWEST *ST* PERCENTAGES

PROJECTS	ST %	REUSE %	TYPE	LANG
UARSDSIM	7.0 %	23.6 %	DS	F
GOESIM	8.1 %	28.8 %	TS	A
POWITS	8.1 %	69.2 %	TS	A
UARSTELS	10.6 %	34.8 %	TS	A
EUVETELS	12.0 %	96.2 %	TS	A

FIVE PROJECTS WITH LOWEST *AT* PERCENTAGES

PROJECTS	AT %	REUSE %	TYPE	LANG
GROSIM	9.0 %	17.9 %	TS	F
COBEDS	11.4 %	26.9 %	DS	F
GOADA	12.8 %	28.5 %	DS	A
COBSIM	13.4 %	10.7 %	TS	F
EUVEAGSS	14.7 %	78.0 %[1]	AGSS	F

[1] Reuse percent excludes ACME portion of project.

Abbreviations and Acronyms

AEM	Applications Explorer Mission
AGSS	Attitude Ground Support System
ASP	Attached Payloads
BBXRT	Broadband X-ray Telescope
CDR	critical design review
COBE	Cosmic Background Explorer
COBEAGSS	COBE Attitude Ground Support System
COBEDS	COBE Dynamics Simulator
COBSIM	COBE Telemetry Simulator
COCOMO	Constructive Cost Model
DEA	Dynamics Explorer A
DEB	Dynamics Explorer B
DEDET	DEB Definitive Attitude Determination System
DERBY	ERBS Dynamics Simulator
DESIM	ERBS Telemetry Simulator
DLOC	developed lines of code
DSPLBLDR	GESS Display Builder
ERBS	Earth Radiation Budget Satellite
EUVE	Extreme Ultraviolet Explorer
EUVEAGSS	EUVE Attitude Ground Support System
EUVEDSIM	EUVE Dynamics Simulator
EUVETELS	EUVE Telemetry Simulator
FDD	Flight Dynamics Division
FOCS	FPSS Off-Null Calibrating System
FOXPP	FOCS Preprocessor
FOXPRO	FOCS Processor
FPSS	fine-pointing Sun sensor
GESS	Graphics Executive Support System
GOADA	GOES Dynamics Simulator (Ada)
GOES	Geostationary Operational Environmental Satellite
GOESAGSS	GOES Attitude Ground Support System
GOESIM	GOES Telemetry Simulator
GOFOR	GOES Dynamics Simulator (FORTRAN)
GRO	Gamma Ray Observatory
GROAGSS	GRO Attitude Support System
GRODY	GRO Dynamics Simulator
GROSIM	GRO Telemetry Simulator
GROSS	GRO Dynamics Simulator
GSFC	Goddard Space Flight Center
GSOC	guide star selection and occultation
ISEEB	International Sun-Earth Explorer B
ISEEC	International Sun-Earth Explorer C
MAGSAT	magnetic satellite

PAS	panoramic attitude sensor
PDR	preliminary design review
POWITS	POLAR/WIND Telemetry Simulator
RMS	root-mean-square
SAMPEX	Solar, Anomalous, and Magnetospheric Particle Explorer
SAMPEXTP	SAMPEX Telemetry Processor
SAMPEXTS	SAMPEX Telemetry Simulator
SEASAT	Ocean Studies Satellite
SEF	Subjective Evaluation Form
SEL	Software Engineering Laboratory
SM	staff months
SMM	Solar Maximum Mission
TBD	to be determined
UARS	Upper Atmosphere Research Satellite
UARSAGSS	UARS Attitude Ground Support System
UARSDSIM	UARS Dynamic Simulator
UARSTELS	UARS Telemetry Simulator

References

1. NASA/Goddard Space Flight Center, *ADA Size Study Report*, S. Condon and M. Regardie (CSC), September 1992

2. NASA/Goddard Space Flight Center, Software Engineering Laboratory (SEL), *Data Collection Procedures for the Software Engineering Laboratory (SEL) Database*, G. Heller, J. Valett, M. Wild, March 1992

3. NASA/Goddard Space Flight Center, Software Engineering Laboratory (SEL), *Software Engineering Laboratory (SEL) Cleanroom Process Model*, S. Green, et al., November 1991

4. Barry W. Boehm, *Software Engineering Economics*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981

5. NASA/Goddard Space Flight Center, *An Approach to Software Cost Estimation*, F. McGarry, G. Page, D. Card, et al., February 1984

6. NASA/Goddard Space Flight Center, Software Engineering Laboratory (SEL), *Manager's Handbook for Software Development*, Revision 1, L. Landis, F. McGarry, S. Waligora, et al., November 1990

7. Computer Sciences Corporation (CSC), *SEAS System Development Methodology (SSDM) Standards and Procedures*, M. Plett, et al. (CSC), June 1990

8. *Quattro Pro Version 4.0 User's Guide*, Borland International, Inc., 1992

9. NASA/Goddard Space Flight Center, Software Engineering Laboratory (SEL), *Software Engineering Laboratory (SEL) Recommended Approach to Software Development*, Revision 3, L. Landis and S. Waligora (CSC), June 1992