

Validation of Spacecraft Software Cost Estimation Models for Flight and Ground Systems¹

Karen Lum, John Powell, Jairus Hihn

California Institute of Technology/Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109

1. Objective

NASA's Jet Propulsion Laboratory (JPL) managed by the California Institute of Technology is currently investing in ways to improve its ability to increase the cost estimation accuracy early in the project life cycle. The need to perform top-down cost and effort estimation for software projects with reasonable accuracy is of paramount importance to future and ongoing work at JPL. In response, both the JPL Costing Office and the Software Quality Improvement Project are emphasizing greater use of cost estimation tools and historical data to improve the accuracy of software cost estimates as well as the incorporation of cost uncertainty. The literature supports the use of cost estimation models to perform such top-down estimates especially when based on an organization's own historical data. [4, 5, 6, 9] Many well known software cost estimation models such as COCOMO II, SEER-SEM and PRICE S advocate local calibration to improve accuracy over the initial built-in calibration that are typically performed over a broad range of industry data. [1, 2, 8, 10] However, it is the initial objective of this study to determine the viability of three commonly used software cost estimation models – COCOMO II, SEER-SEM, and PRICE S – prior to any local calibration.

Section 2 will provide a brief description of the three models being evaluated. Section 3 contains the details of the research team's methodology. The descriptions of each project in the study are detailed in Section 4. Overall, relevant analysis and summary results will be presented in Section 5. Finally, conclusions and recommended future work will be discussed in Sections 6.

2. The Models

2.1 Overview

Software cost estimation models are a means of top-down estimation of future project effort based on characteristics that are known earlier in the lifecycle. The models discussed below have several similarities. First is a basic measure of size. The models accommodate either source lines of code (SLOC) or function points (FPs) or both as a measure of project size. Second is the mathematical form of the statistical models. Categorically the models can be thought of a specific instantiation of the basic equation $E = A * \text{Size}^B * EM$ where E is effort, A is a constant that reflects a measure of the basic organizational/technology costs, B is a scaling factor of Size

¹This work was performed for the Jet Propulsion Laboratory, California Institute of Technology, sponsored by the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

and EM is a group of effort multipliers that measure environmental factors used to adjust effort (E). The set of factors comprising EM are commonly referred to as cost drivers because they adjust the final effort estimate up or down thus driving the cost accordingly. Third, all the models discussed here are of a multiplicative form. This means that the margins for error in the estimates are expressed as a percentage. Therefore, large projects will have a larger variance in dollars than smaller projects when estimated using these models. Next, the models are similar in that they all account for the notion there exists an optimal schedule for the project as described by the model inputs. As the actual schedule for a project is factored into the model's estimate the cost/effort is raised or lowered appropriately based on the discrepancy between the proposed schedule and the schedule the model believes is optimal. It is important to note that this cost increase may be reflected as a reduced estimate of the likelihood of success at a given cost. Finally, all the models accommodate a decomposition of the project into logical sub parts. This facilitates the common case where different parts of an over all project have different characteristics. For example, one portion of the project's system may have low complexity with average programmers implementing it while another part may be highly complex with very talented programmers. Statistically, the combination of these different parts for reasonably accurate estimates is nontrivial. Simply averaging the inputs from various parts of the project and allowing the model to treat all parts of its system equally will produce wildly inaccurate estimation results. This will be demonstrated in the data of this study where DS1 was treated as a single estimation element and then properly decomposed and treated as a composition of differing project parts. The former produces an outlying data point that bears no resemblance to the estimate using decomposition. However, the estimate based on decomposition produces error rates within tolerances when compared to other projects in the study.

2.2 COCOMO II

COCOMO II is a model developed by Dr. Barry Boehm that helps you “reason about the cost and schedule implications of software decisions you may need to make.”[2] The COCOMO II cost estimation model is used by thousands of software project manager and is based on a study of hundreds of software projects. “Unlike other cost estimation models, COCOMO II is an open model, so all of the details are published.”² There are actually variations of the model, one for early software design phases and one for later software development phases. The amount of information available during the different phases of software development varies and COCOMO II incorporates this by reducing the number of cost drivers required for input during the early design phase of development versus the post architecture phases.

2.3 SEER-SEM

“SEER-SEM is a tool for software estimation, planning and project control.”[8] This model offers a sophisticated interface for building up the data in the model hierarchically by “rolling-up” user defined Computer Software Configuration Items (CSCIs) that have been modeled. In addition to various model output features such as Defined Reports and Charts, SEER-SEM makes use of Knowledge Bases (KBases). KBases are complete or partial sets of default inputs for the model. Each KBase is associated with a software industry sector (i.e. Aerospace-Missile and Unmanned Airborne, Application-Device Driver, etc...). The user may select an appropriate

² <http://www.softstarsystems.com/overview.htm>

KBase when little is known about the software project characteristics with respect to the input parameters. The KBase provides an estimate of the industry sector average for each parameter in the KBases set as input to the model. The user may then modify the inputs to customize the KBase to the organization's project(s). SEER-SEM offers the user the capability to specify these cost drivers as a range of input values when knowledge exists about a model parameter as well as the degree to which it may vary. Finally, SEER-SEM offers interoperability with a number of other useful software applications.

2.4.PRICE S

PRICE S estimates the costs and schedules of software projects. It handles a wide array of software projects types varying sizes from component level software to highly complex systems. This model has the ability to estimate the project as whole or in parts. It takes into account such things as development, modification, and life cycle costs. As with the other models, software size is a critical input and PRICE S provides sizing applications to aid in the determination of size, but unlike COCOMO II, this model is a closed model. Similar to SEER-SEM knowledge bases, PRICE S supplies industry-average values for actual input data that has not yet been specified or is unknown at the time of the estimate. A difference between PRICE S and other models is that it uses a productivity factor called PROFAC to capture the skill levels, experience, efficiency, and productivity of an organization. PROFAC tends to be consistent within an organization but is a very sensitive parameter. Although typical values for various applications are provided by the PRICE S User's Manual, it is essential to calibrate an organization's PROFAC value before accurate cost estimates can be made.

3. Methodology & Approach

3.1.Data Collection

There exist a significant overlap between the input data required for COCOMO II and SEER-SEM. Therefore, questionnaires that were originally designed for COCOMO II input data collection were reviewed and adapted for use in an effort to collect SEER-SEM data simultaneously. The revised survey was subsequently used in interviews with key software people of the various projects described below. (See Section 4) The similarity of SEER-SEM and COCOMO II parameters/cost drivers facilitated the ability to map survey answers (data) to both models. The cost drivers in the PRICE S model were not as easily mappable between the drivers from COCOMO II and SEER-SEM, although an attempt was made to translate the data between models.

The survey instrument asked the key software people to rate their piece of the project in various categories, from team capabilities, software reliability, complexity, and tool usage. Multiple (2-3) interviewers participated in each of the interviews conducted. Then interviewers compared notes taken during the interviews and cost driver ratings derived from conversation with the interviewee. All cost driver-rating discrepancies were resolved through discussion between the interviewers providing rationale. Discrepancies were noted and, when appropriate, ranges were formed for input parameters when full resolution could not be reached. Using the results from these interviews along with historical data, the models' inputs were entered into the associated

implementation tools (software) for each. Follow-up interviews were conducted as necessary for the purpose of data conflict resolution and clarification with regard to project scope and CSCI applicability. Past projects were selected in order that actual cost and effort data could be collected to compare with the estimates produced by the models.

3.2. Analysis

The software implementations of the COCOMO II, PRICE S, and SEER-SEM cost estimation models were used to semi-automate the analysis of the data collected. The data, in the form of each model's cost drivers, was entered into the models and analyzed in an "as-is" form for a reasonableness check. Based on the results of this early analysis, the data was reviewed for consistency, and follow-up interviews were conducted to ensure that the research team's understanding of the context and specifics of the survey data were consistent with the interviewee's understanding. In some case secondary resources were consulted in the form of data from previous studies/activities and alternate key personnel to ensure a coherent global perspective of data pertaining to specific subsystems and CSCIs in relation to overall project and organization data.

3.2.1. Assumptions

The next phase of the analysis involved the identification of model input data that was consistently estimated too high or too low by interviewees across all projects studied.

The research team concluded that data values collected during interviews pertaining to JPL project software's Complexity and Required Reliability ratings are downwardly biased by JPL personnel. The rating, which can range from "Very Low" to "Very High," of complexity and required reliability given by interviewees were inadvertently discussed and measured in the context of flight software at JPL during the interviews. JPL personnel tended to rate these factors too low because, when compared to the industry at large, "Nominal" complexity and "Nominal" required reliability for spacecraft software is far higher than the average for software being developed throughout the industry. That is to say flight software for spacecraft categorically has a higher required reliability and complexity. The models' view of a "Nominal" rating for these inputs did not account for this. This adjustment may be viewed as the beginnings of local calibration for the models examined through parameter adjustments.

3.2.2. Adjustments³

Some data had to be scrubbed. Specifically:

- Fault protection code was largely auto-generated. Therefore, these lines of code are counted differently from non-auto-generated code. The effort related to fault protection was "backed out".
- Some projects had management separated out as a separate category, which had to be reallocated back to the various project elements, such as ACS, CDH, Nav, etc. This was

³ Each model includes different activities in their effort prediction. In addition, each model defines its labor categories and activity phases differently. Therefore, adjustments to the model effort predictions were made based on what software development activities were included in the effort actuals, after consulting with representatives from Galorath Inc., PRICE Systems, LLC, and the University of Southern California's Center for Software Engineering.

done by calculating the percentage of effort each project element represented, and then allocating that percentage of management back to it. Also, the effort for subcontracted software elements was taken out, such as a remote agent for one of the projects.

- For Project 1, software elements that utilized automatic code generation were backed out of the effort total. Management effort was then allocated back based on the percentage of overall effort that each element represents.
- The projects were sized in deliverable source lines of code (all lines of code except comments and blanks), while the models take logical lines of code as input. Therefore, the size of each project was reduced by 25%. [7]

The following adjustments to the model estimates were justified and documented.

1. Actual effort was adjusted by allocating management and support effort to SW elements based on its overall percentage of the total effort being managed.
2. COCOMO II provides three levels of estimates – optimistic, most likely, and pessimistic. The “most likely estimate” is presented and used in calculation of the magnitude of relative error. (See Table 3 & Table 4)
3. The effort estimate for “system requirements” was backed out of the final SEER estimates since our actuals did not include much of this activity. Also system integration (integration between hardware and software) for flight software was book kept separate from other software development activities and was therefore taken out of the SEER-SEM and PRICE S estimates.
4. A percentage of CM, QA, Management, and QA were taken out of the COCOMO II and SEER-SEM estimates. Many of these activities were performed, although informally. The percentage taken out of the estimate was to account for the formal activities being book kept elsewhere.
5. The effort estimate for the following labor categories was taken out of the PRICE S estimates: CM and QA (PRICE S defines these activities as formal activities that are book kept elsewhere at a higher level). Design and programming was usually done by the same person – the programmers – therefore a percentage of design was also backed out of the PRICE S estimate. 60% of Program Management/System Engineering was also taken out of the estimate. PRICE S’s definition for these labor categories include some overall program and system-level efforts that our actuals did not include.
6. The effort for the following phases was also excluded from the PRICE-S estimate: Concept, System Requirements, Hardware/Software Integration, and Field Test.

The initial effort estimates prior to making the adjustments discussed above are presented in Table 1 and Table 2. These estimates can be compared with Table 3 and Table 4. A first-time user with little experience with cost models would probably not know to make the adjustments necessary to make the estimates comparable. The estimates in Table 3 and Table 4 are properly adjusted to activity and labor categories that are accounted for in actual effort so that they can be evaluated accurately.

Project	Adjusted Actual Effort ⁴	Unadjusted COCOMO estimate	COCOMO % MRE ⁵	Unadjusted SEER Estimate	SEER % MRE	Unadjusted SEER KBase	KBase % MRE	Unadjusted PRICE S	PRICE % MRE
Project 1	446.34	319.3	-28%	800.47	79%	305.19	-32%	1305.70	193%
Project 2	860.64	366.2	-57%	849.74	-1%	905.66	5%	1939.00	125%
Project 3	592.28	1038.2	75%	2473.71	318%	2574.36	335%	3256.30	450%
Project 4	592.28	604.4	2%	1184.53	100%	1481.62	150%	2895.50	389%
Project 5	234.7	314.9	34%	796.33	239%	589.07	151%	1268.00	440%
Project 6	230.48	81.5	-65%	146.03	-37%	577.55	151%	958.40	316%
Project 7	127.1	123.8	-3%	242.17	91%	315.01	148%	569.50	348%
Project 8	285.6	168	-41%	1282.31	349%	2518.22	782%	1806.70	533%
Project 9	72	51.3	-29%	116.98	62%	78.52	9%	295.20	310%
Project 10	314	283.8	-10%	730.23	133%	650.72	107%	1172.10	273%

TABLE 1. UNADJUSTED⁶ FLIGHT SOFTWARE ESTIMATES

Project	Adjusted Actual Effort	Unadjusted COCOMO estimate	COCOMO % MRE	Unadjusted SEER Estimate	SEER % MRE	Unadjusted SEER KBase	KBase % MRE	Unadjusted PRICE S	PRICE % MRE
Project 11	681.48	1863.7	173%	10909.6	1501%	24989.46	3567%	11,686.20	1615%
Project 12	455.22	1644.7	261%	6800.67	1394%	13849.6	2942%	4,499.30	888%
Project 13	495.5	1283	159%	6450.79	1202%	11791.73	2280%	3,287.70	564%
Project 14	631	586.2	-7%	1444.63	129%	1742.2	176%	2,048.00	225%
Project 15	433	328.2	-24%	1513.39	250%	1882.49	335%	1,535.80	255%
Project 16	499	514.8	3%	2022.09	305%	1759.64	253%	1,947.80	290%
Project 17	128	59.5	-54%	166.74	30%	808.43	532%	604.80	373%
Project 18	58	27.2	-53%	58.73	1%	255.31	340%	216.70	274%
Project 19	130	52	-60%	336.29	159%	1362.9	948%	400.40	208%

TABLE 2. UNADJUSTED⁶ GROUND SOFTWARE ESTIMATES

The models were rerun over the newly adjusted data set, the results of which are present throughout the remainder of this paper.

4. The Projects Studied

4.1. Projects

Both flight software and ground software are examined, but due to the inherently dissimilar characteristics of each, our analysis focuses on each type of software separately. Data from elements of five flight software projects from JPL were collected for a total of ten flight software data points (two of the data points are a sum up of the project's smaller software pieces). There

⁴ All effort estimates and actuals are presented in Work-Months. Actuals have been adjusted by allocating management and support effort to SW elements based on its overall percentage of the total effort being managed.

⁵ %MRE (percent magnitude of relative error) = (|Estimate - Actual| / Actual) * 100

⁶ The outputs of the models in this table have not been adjusted for activities and labor categories, as a novice user might not know to make such adjustments. However the lines of code, an input into all the models, have been adjusted from physical line counts to logical line counts.

are not many flight software projects that are conducted in-house at JPL; most are contracted out. There are nine ground software data points from eight JPL projects. Due to confidentiality considerations each software project will be referred to as projects 1-19. Projects 1-10 are flight software projects. The remainder (11-19) are ground support software projects.

4.1.1. Flight Software

Projects 1-2 are subsystems in a planetary mission that will, among other things, perform multiple in-flight encounters in the course of multi-year mission. Projects 1 and 2 Flight Software is onboard (the spacecraft) software. The segments of the software that were examined in this study are:

- Attitude Control Subsystem
- Command and Data Subsystem

Projects 3-7 are related in that projects 5-7 are in fact individual subsystems of a project that was also analyzed as a “rollup” in project 4 and as a single logical system unit in project 3. That is to say that the data was studied both as many individual segments within the project and as an overall effort. These software projects (3-7) are “the first of a series of NASA missions that will demonstrate and validate advanced technologies to develop a baseline framework for future spacecraft and missions.” The segments that were studied individually were a:

- Flight System Control Subsystem
- Attitude Control Subsystem
- Navigation and Control Subsystem

Project 8 was a NASA discovery mission, whose goal was to conduct a low-cost space exploration, while highlighting some new break-through technologies, and rapid spacecraft development concept using modern re-engineering processes. The success of Project 8’s mission was used to validate NASA’s initiatives for more efficient mission development. Project 9 was the flight software for an onboard instrument in an Earth orbiting mission. Project 9 uses a rotating dish antenna with two spot beams that sweep in a circular pattern. The instrument on the satellite is a specialized weather-sensing device. Project 10 is flight software for a spacecraft involved in a scientific space mission.

4.1.2. Ground Software

Projects 11-13 are ground based software systems for earth orbiting missions. The initial estimates provided by the models for these projects were highly inaccurate. The research team argues that this is indicative of a highly irregular set of circumstances surrounding these three projects that may be reconciled in the future work. The Project 11 mission was lost well before completion the software effort. Subsequently Project 13 was initiated which leveraged almost in its entirety, the uncompleted effort of Project 11. Finally, during Project 13, the same project team that was developing Project 11 was employed to develop Project 12. Project 12 was in essence a continuation and upgrade of the Project 11 mission. Currently the discrepancy between the actual cost and effort and the estimations provided by the models appears to be an allocation anomaly within the JPL data. Its is unclear at this time how the effort and reuse factors from these three highly related projects which experience significantly atypical lifecycles is allocated within the data and whether that allocation is appropriate for the purposes of a top-down estimation through the use of cost estimation models. Typically, inputs would be given to a model and it would be asked to produce an estimate with the presupposition that the project would be attempted with the goal of completion. To reconcile this anomaly for calibration

purposes the team must reconstruct a scenario of what the allocation with the JPL data would have been for each project if they were initiated under reasonably typical JPL circumstances (i.e. if Project 11 was not lost prematurely).

Projects 14-19 are six ground software data points consisting of subsystems from a larger ground based software system. They primarily relate to monitor and control and telemetry processing functions.

5. Results

5.1. Outliers

Project 3, represented by triangles in Figure 2, Figure 3, Figure 4, and Figure 5 is a potential outlier for all cases. Looking at Projects 5-7 software as one overall product produced overly high estimates from the actual effort, whereas breaking down software into smaller elements and then rolling up the effort (Project 4) generates estimates that are more accurate. This is consistent because smaller pieces can be developed concurrently and larger software projects tend to have lower productivity rates.

The Project 1 and Project 2 data points, represented by the star-shaped points, can also be considered as outliers for they were developed prior to the NASA initiative for more efficient software development. These projects were often underestimated by the models.

Other outliers are the projects 11, 12, and 13, represented in the figures by squares. The models greatly overestimated their costs. The unique circumstances of these projects' development are discussed in Section 4.1.2.

5.2. Flight Software Results

Project	Actual Effort	Adjusted COCOMO Estimate	COCOMO %MRE	Adjusted SEER Estimate	SEER %MRE	Adjusted SEER KBase	KBase %MRE	Adjusted PRICE S Estimate	PRICE %MRE
Project 1	446.34	276.06	-38%	528.60	18%	267.105	-40%	355.40	-20%
Project 2	860.64	317.71	-63%	571.80	-34%	779.57	-9%	622.10	-28%
Project 3	592.28	857.31	45%	1031.08	74%	1480.425	150%	915.60	55%
Project 4	592.28	499.21	-16%	749.11	26%	1351.96	128%	778.90	32%
Project 5	234.7	258.61	10%	501.85	114%	608.25	159%	288.00	23%
Project 6	230.48	66.74	-71%	92.28	-60%	469.685	104%	287.20	25%
Project 7	127.1	101.26	-20%	154.97	22%	274.025	116%	176.10	39%
Project 8	285.6	141.38	-50%	545.79	91%	1374.26	381%	545.20	91%
Project 9	72	41.59	-42%	73.84	3%	63.795	-11%	88.50	23%
Project 10	314	233.53	-26%	459.77	46%	548.6	75%	323.00	3%

TABLE 3. ADJUSTED⁷ FLIGHT SOFTWARE ESTIMATES

⁷ The outputs of the models have been adjusted by taking out any labor categories and activity phases not included in the JPL actual effort.

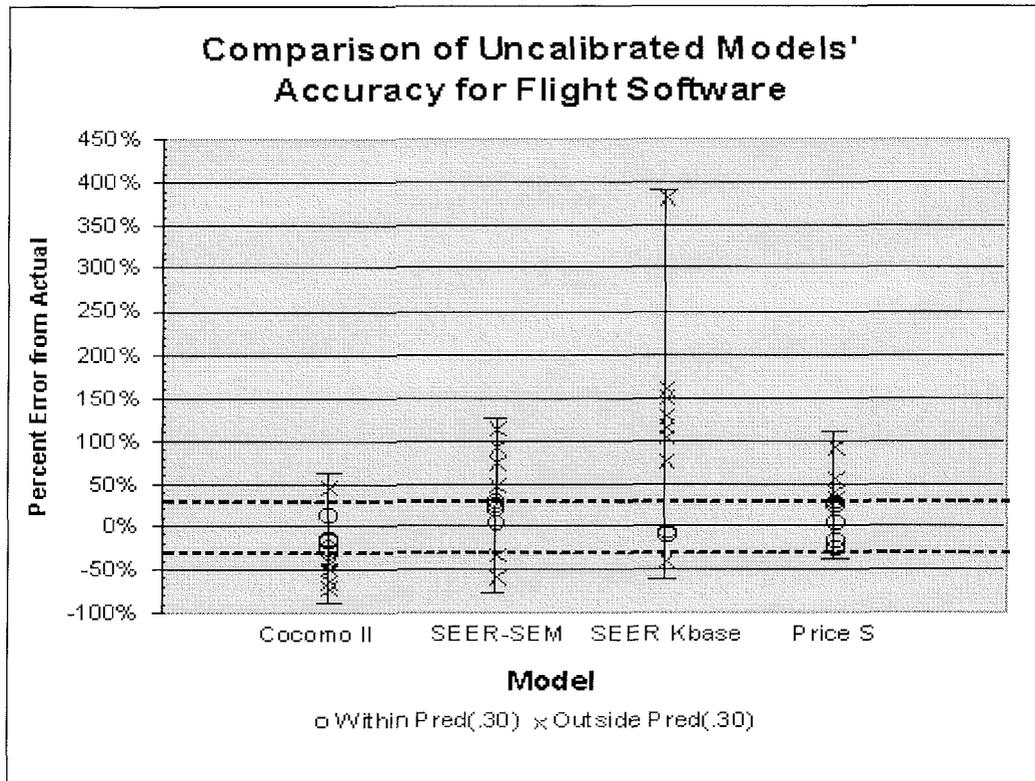


Figure 1. Comparison of Uncalibrated Models' Accuracy for Flight Software

The flight software model estimates in work-months and their percent magnitude of relative error (%MRE) for COCOMO II, SEER-SEM, the SEER-SEM knowledge bases (using only the knowledge bases and minimal inputs – SLOC, and no parameter inputs) and PRICE S are presented in Table 3. Figure 1 graphically shows the deviations from actuals of each project for each cost model. Circles are the projects that are within 30% of actuals; Xs are those that are more than 30% from actuals. The star-shaped points on Figure 2, Figure 3, Figure 4, and Figure 5 represent Projects 1 and 2 and are atypical in comparison to the remaining data points as discussed in Section 5.1. The triangular point was Project 3. (See 5.1.) Project 3 versus Project 4 was used as a sanity check. The analysis of Project 3 data point as a single piece of software is misrepresentative of the actual project development environment. The results from the model runs, including the knowledge base run show that it is better to break a project down into smaller segments to develop better cost estimates.

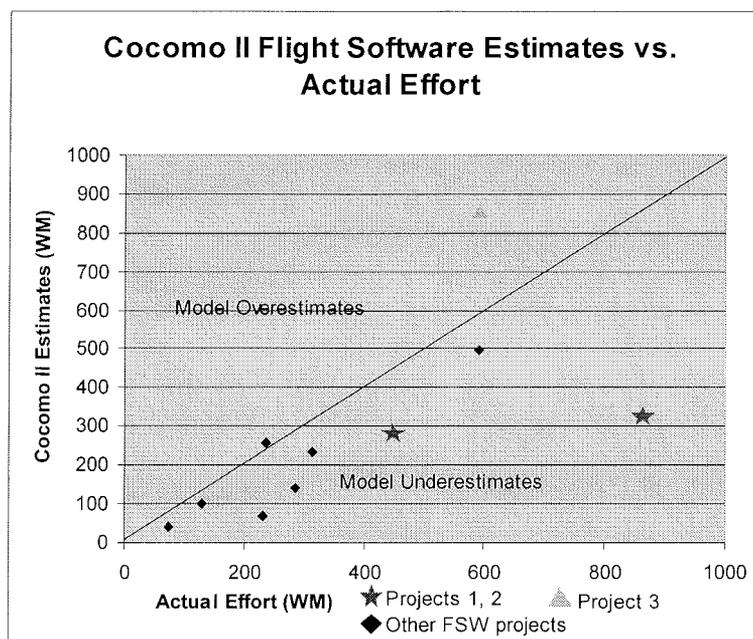


Figure 2. COCOMO II Flight Software Estimates vs. Actual Effort

On average, the COCOMO II model underestimates effort by a mean magnitude of -27%. (See Table 5) A paired difference test between the COCOMO estimates and the actuals result in a t-value of -1.51 indicating that we cannot reject the null hypothesis – that the model estimates are not different from the actuals. Without calibration, COCOMO II is accurate to within 30% of actual effort 40% of the time for flight software, while a study performed by Daly for AFIT resulted in precalibration estimates for REVIC,⁸ PRICE S, and SEER-SEM within 30% of actuals no more than 33% of the time [4]. Thus, the unadjusted COCOMO II outperformed the results from the commercial models for the JPL flight environment.

The SEER-SEM model tends to overestimate more than it underestimates (See Figure 3). On average the SEER-SEM model overestimates effort by 30%. SEER-SEM cannot be rejected with a t-value on paired two sample for means of 1.44. (See Table 5) SEER-SEM with no calibration was accurate to within 30% of actual effort 40% of the time for flight software. SEER is also outperforming results from commercial cost models for the JPL flight environment.

⁸ The Air Force Cost Analysis Agency’s computerized variant of COCOMO

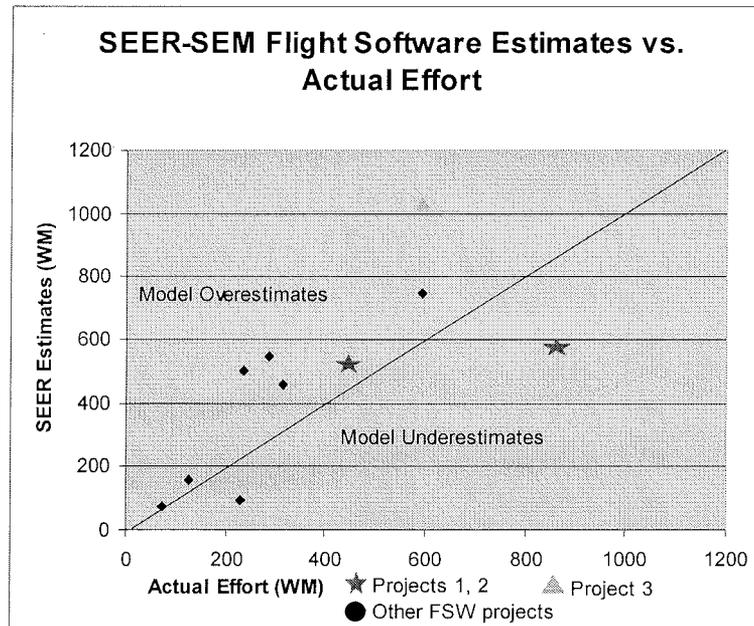


Figure 3. SEER-SEM Flight Software Estimates vs. Actual Effort

A paired difference analysis of the knowledge base-only estimates for flight software against the actual effort shows that the model cannot be accepted. Using only the SEER-SEM knowledge bases provided generates estimates that tend to be much higher than the SEER-SEM estimates with parameter inputs. The knowledge bases overestimate effort by an average of 105%, which is the highest mean magnitude of relative error (MMRE) of the models being compared. The knowledge bases generate estimates that have a large error rate despite the outliers mentioned above. (See Figure 4) Using only the knowledge bases produces estimates that are within 30% of actual effort only 20% of the time for flight software. This is very poor performance. The tests on a knowledge base-only model are not meant to disprove the validity of the knowledge bases, but rather to show that knowledge bases should not be used as the sole basis of an estimate. The SEER-SEM knowledge bases can be valuable tools if used properly and in conjunction with cost driver inputs to fill in unknown parameters.

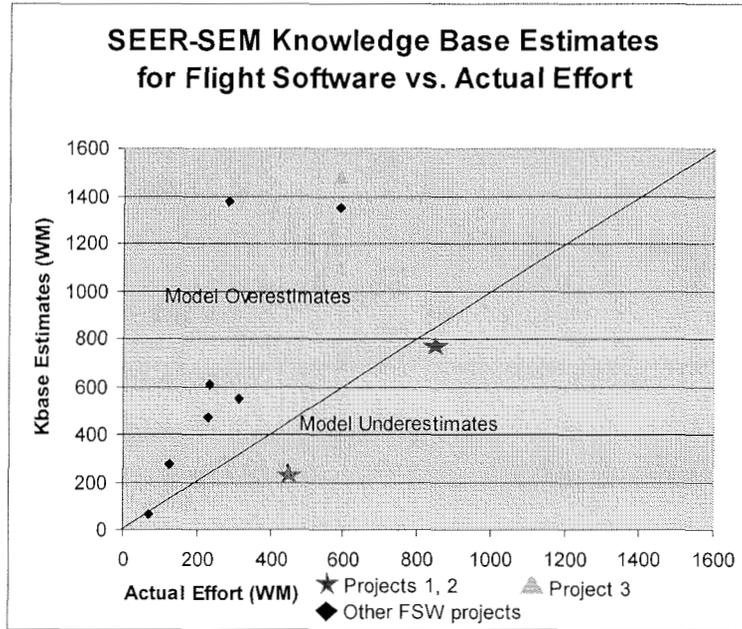


Figure 4. SEER Knowledge Base Estimates for Flight Software vs. Actual Effort

A paired difference experiment shows that PRICE S also cannot be rejected for flight software, with a t-value of 1.21 (See Table 5). PRICE S has a mean magnitude of 24%. PRICE S predicts within 30% of actuals 60% of the time for flight software, the strongest prediction level of the models being compared. This is good performance for an uncalibrated model. Figure 5 shows that Price S estimates have the strongest linear relationship with actuals.

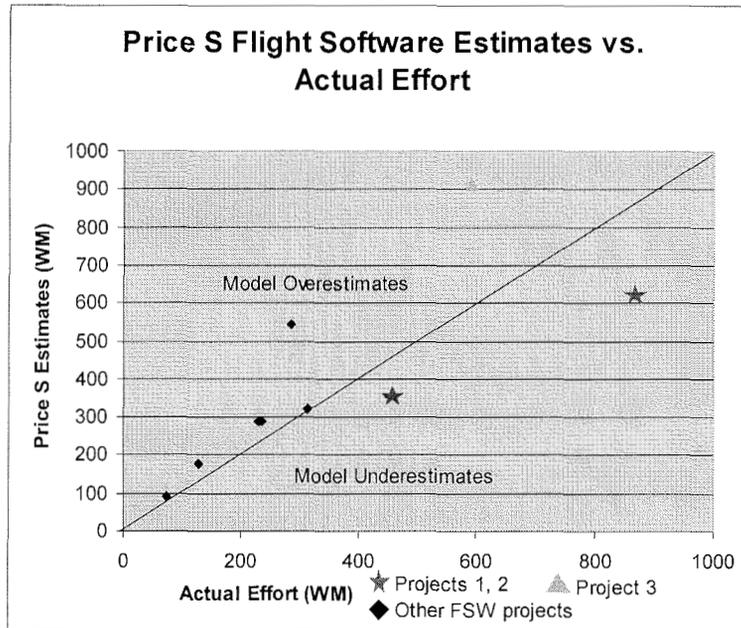


Figure 5. PRICE S Flight Software Estimates vs. Actual Effort

5.3. Ground Software Results

Project	Actual Effort (adjusted)	Adjusted COCOMO Estimate	COCOMO %MRE	Adjusted SEER Estimate	SEER %MRE	Adjusted SEER KBase	KBase %MRE	Adjusted PRICE S Estimate	PRICE %MRE
Project 11	681.48	1542.33	126%	3453.49	407%	9486.005	1292%	2,598.50	281%
Project 12	455.22	1360.09	199%	2325.06	411%	6468.53	1321%	1,119.80	146%
Project 13	495.5	1061.01	114%	2242.755	353%	6240.85	1160%	840.50	70%
Project 14	631	544.05	-14%	932.46	48%	1140.01	81%	678.70	8%
Project 15	433	304.57	-30%	976.845	126%	1231.81	184%	518.10	20%
Project 16	499	477.58	-4%	1305.19	162%	1151.42	131%	632.40	27%
Project 17	128	55.05	-57%	146.77	15%	247.565	93%	213.40	67%
Project 18	58	25.1	-57%	53.315	-8%	81.19	40%	76.40	32%
Project 19	130	48.16	-63%	239.93	85%	671.27	416%	147.70	14%

TABLE 4. ADJUSTED⁷ GROUND SOFTWARE ESTIMATES

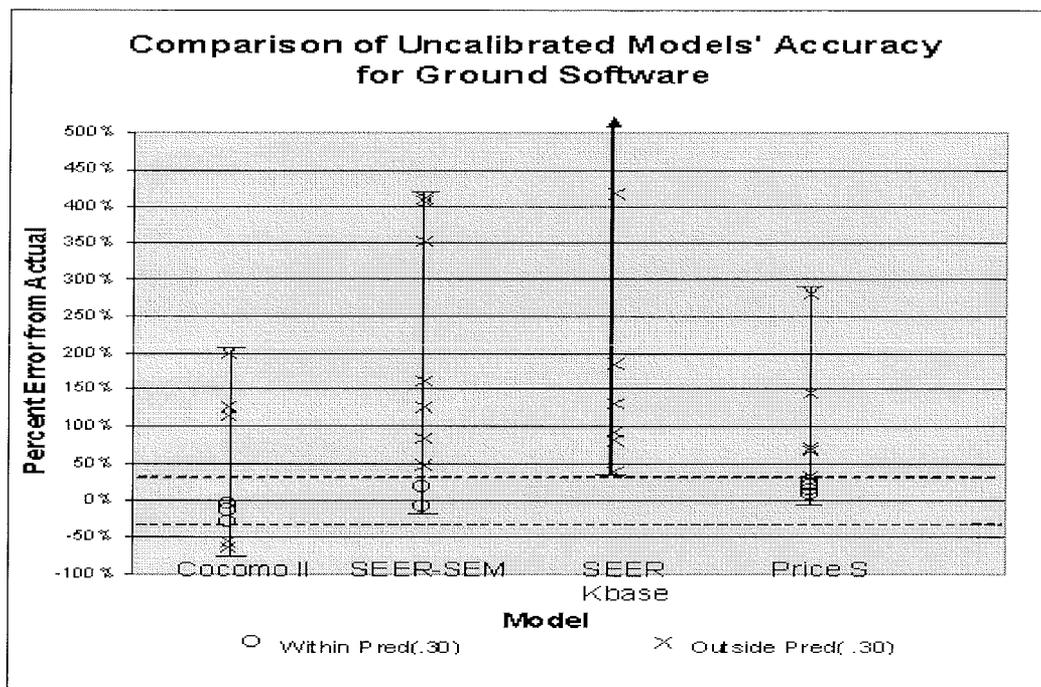


Figure 6. Comparison of Uncalibrated Models' Accuracy for Ground Software

Ground software estimates for the models are presented in Table 4. The square points in Figure 7, Figure 8, and Figure 9 are projects 11, 12, and 13 discussed previously. (See Section 5.1)

The COCOMO model tends to predict well for ground software, within 63% (the smallest range for ground software) of actuals if the outliers are ignored (See Figure 7). A paired difference experiment on the COCOMO II estimates and the actual effort results in a conclusion that the null hypothesis cannot be rejected. Thus cannot reject the model. (See Table 6) COCOMO II overestimates on average about 24% for ground software if the outliers are included. However, COCOMO tends to underestimate if the outliers are ignored. The COCOMO II model produced

estimates that were accurate within 30% of actual effort only 33% of the time for ground software with the outliers. However, with outliers excluded the model predicts with 30% of actual effort 50% of the time.

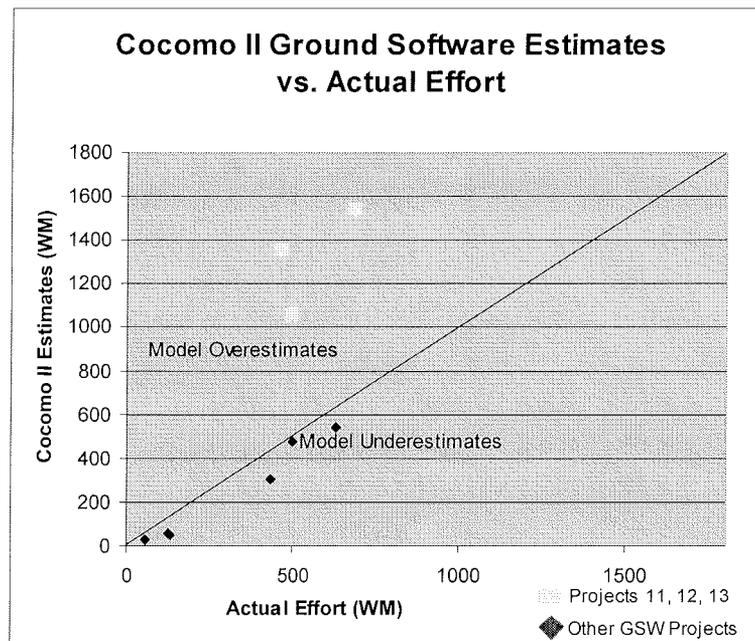


Figure 7. COCOMO II Ground Software Estimates vs. Actual Effort

The SEER model and the knowledge bases overestimate the effort for ground software by a large percentage (See Table 4). A large t-value causes us to reject the null and accept the alternate hypothesis that the model estimates are significantly different than the actual effort at the 95% level for both SEER-SEM with parameter inputs and the knowledge base-only run. (See Table 6) Even excluding the three atypical ground projects, SEER overestimates effort by an average of 72%. (See Figure 8) Only 22% of the SEER estimates were within 30% of actuals for ground software. None of the knowledge base-only estimates were within 30% of actuals. Surprisingly, throwing out the ground software outliers does not improve most test results. Only SEER-SEM has a small enough t-value to not be rejected if outliers are ignored.

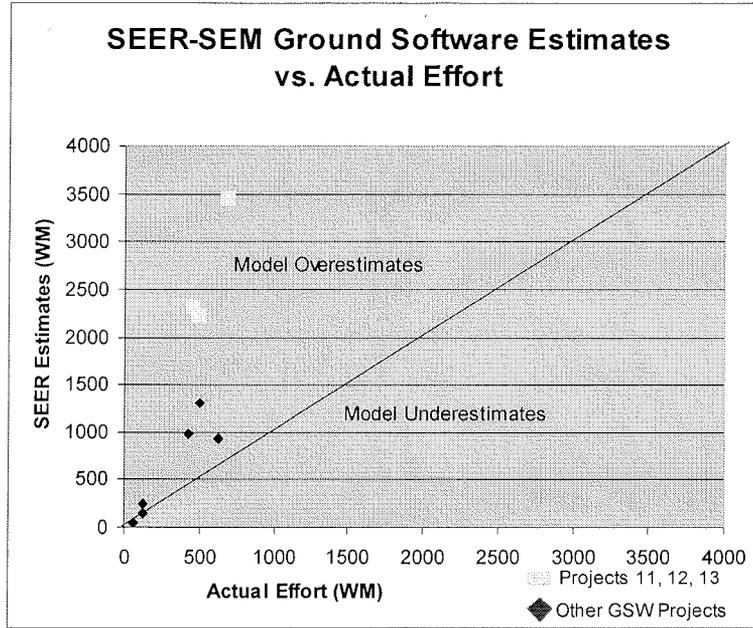


Figure 8. SEER-SEM Ground Software Estimates vs. Actual Effort

Price S also predicts well for ground software, within 67% of actual effort if the outliers are ignored. PRICE S always overestimates for ground software, by a mean magnitude of 74%. The PRICE S estimates are the strongest predictors of ground software costs out of all the models, with 44% of the estimates falling within 30% of actuals. There is a strong linear relationship between estimates and actuals, whether or not the outliers are ignored. (See Figure 9) However, throwing out the outliers will result in the model being rejected. (See Table 6)

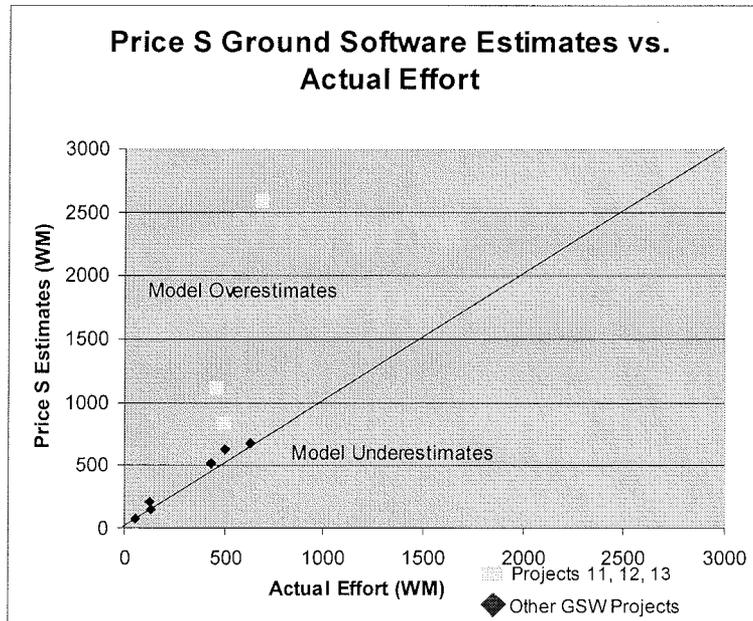


Figure 9. PRICE S Ground Software Estimates vs. Actual Effort

5.4. Flight and Ground Combined

Looking at all the data points as a whole, the COCOMO model and PRICE S are the only models in which the null (that the model estimates are not significantly different from actual effort) cannot be rejected. (See Table 7) With high t-values, the SEER model and the knowledge base estimates are significantly different from the actual effort.

Flight Software Model Estimates vs. Actual Effort

	All Flight Software Data points				Without Projects 1, 2, & 3 Outliers			
	COCOMO II	SEER-SEM	KBase	PRICE S	COCOMO II	SEER-SEM	KBase	PRICE S
MMRE ⁹	-27%	30%	105%	24%	-31%	35%	136%	34%
RMS ¹⁰	214.25	220.80	535.09	167.55	96.39	171.04	539.28	125.94
RRMS ¹¹	0.57	0.59	1.42	0.45	0.36	0.65	2.03	0.47
Pred (.25) ¹²	30%	30%	20%	50%	43%	29%	14%	67%
Pred (.30) ¹³	40%	40%	20%	60%	57%	43%	14%	67%
Pearson r ¹⁴	0.66	0.72	0.54	0.8	0.92	0.85	0.81	0.94
t-value ¹⁵	-1.51	1.44	2.55	1.21	-2.89	1.85	2.78	2.51
p-value	0.17	0.18	0.03	0.26	0.03	0.11	0.03	0.05
critical t	2.26				2.45			

TABLE 5. FLIGHT SOFTWARE RESULTS

Ground Software Model Estimates vs. Actual Effort

	All Ground Software Data points				Without Projects 11, 12, & 13 Outliers			
	COCOMO II	SEER-SEM	KBase	PRICE S	COCOMO II	SEER-SEM	KBase	PRICE S
MMRE	24%	178%	524%	74%	-37%	72%	158%	28%
RMS	461.55	1303.09	4059.55	688.87	79.18	418.18	521.32	76.66
RRMS	1.18	3.34	10.41	1.77	0.25	1.34	1.66	0.24
Pred (.25)	22%	22%	0%	33%	33%	17%	0%	50%
Pred (.30)	33%	22%	0%	44%	50%	17%	0%	67%
Pearson r	0.76	0.79	0.64	0.75	0.99	0.91	0.90	0.99
t-value	1.46	2.75	2.33	1.79	-4.45	2.25	3.54	3.5
p-value	0.18	0.03	0.05	0.11	0.01	0.07	0.02	0.02
critical t	2.31				2.57			

TABLE 6. GROUND SOFTWARE RESULTS

⁹ Mean Magnitude of Relative Error (MMRE) = $\Sigma \text{MRE}/n$

¹⁰ Root Mean Square (RMS) = $[(1/n) * \Sigma (\text{Estimate} - \text{Actual})^2]^{1/2}$

¹¹ Relative Root Mean Square (RRMS) = $\text{RMS}/(\Sigma \text{Actuals}/n)$

¹² Pred (.25) = percent of estimates that predict within 25% of actuals

¹³ Pred (.30) = percent of estimates that predict within 30% of actuals

¹⁴ Pearson r = linearity or relationship

¹⁵ Reject the null hypothesis that Actuals are equal to Model Estimates if t-values > critical t at the 95% significance level

Combined Flight and Ground Software Data Points

	All Software Data points			
	COCOMO II	SEER-SEM	KBase	PRICE S
MMRE	-3%	100%	304%	48%
RMS	353.65	911.04	2820.81	502.86
RRMS	0.92	2.38	7.38	1.31
Pred (.25)	26%	26%	11%	42%
Pred (.30)	37%	32%	11%	53%
Pearson r	0.63	0.59	0.44	0.65
t-value	0.60	2.63	2.43	1.98
p-value	0.55	0.02	0.03	0.06
critical t	2.1			

TABLE 7. ALL SOFTWARE DATA POINTS TEST RESULTS

According to Ferens and Christensen, “a model’s estimate is accurate when MMRE < 0.25, RRMS is < 0.25, and Pred (.25) < .75.”[3] Although the models only meet one or two of these criteria at a time (See Table 5, Table 6, and Table 7), they show promising signs for improved accuracy with calibration. Based on this study, various calibration options exist for the models. More data is required to make a definitive determination as to the calibration of COCOMO II, SEER-SEM, and PRICE S.

6. Conclusions

JPL, because its primary focus is developing and operating deep space science missions, has many characteristics that make it unique from other organizations that develop software. At the same time, we share many things in common. Therefore, we wanted to determine whether some cost estimation tools could be used “right out of the box” or even if they were applicable to the JPL environment at all. It was found that all three of the “uncalibrated” models being evaluated – COCOMO II, SEER-SEM, and PRICE S – were able to predict within similar ranges based on the measures we used to evaluate the models. On average 50% of the model estimates are predicting within 30% of the actuals. Given that these models are unadjusted for JPL’s local environment, they performed much better than we had originally expected. However, adjusting the models for the local environment should improve this performance. Our goal is to have a set of tools that are able to predict 80% of the time within 30% of the actual effort.

There are strengths and weaknesses to all three of the models. All models predict better for flight software than ground software in general. COCOMO II has strong results for both flight and ground software. SEER predicts well for flight software but not as well for ground. The knowledge bases, if used alone, are very poor predictors of JPL software in general. PRICE S is the strongest predicting model for flight software and ground software. However, its prediction range is wider than COCOMO II’s. In all three models, careful consideration must be made as to which labor and activity categories should be included.

COCOMO II has fewer features but is also the easiest model to use. SEER-SEM and PRICE S are more sophisticated. PRICE S requires less input parameters, but it is more difficult to

understand how the inputs impact the estimated cost. SEER-SEM has many input parameters that can be confusing at times, but the user can use the knowledge bases when parameters are unknown. In addition, having knowledge bases gives us the possibility of creating custom knowledge bases that can better fit the JPL environment. All three require some training before they can be properly used. Training should consist of determining what labor and activities categories should be included or excluded and how that should be done.

All three models show promise for viability but need “calibration” for JPL’s flight and ground software environment. Clearly, individual calibrations of SEER-SEM, PRICE S, and COCOMO II for flight and ground software are needed to improve prediction level. Although the models predict within a reasonable range, it is our goal after adjusting the models – by either calibration or some other consistent method – to get 80% of the estimates within 30% of actual effort. This will require collecting additional data.

As a final note, we would like to express our gratitude to NASA’s Independent Project Assessment Office for providing funding to enable us to work more closely with Galorath Inc. and PRICE Systems, LLC to calibrate their respective models to the JPL environment.

References

1. B. Boehm, *Software Engineering Economics*, Englewood Cliffs, New Jersey, Prentice-Hall, Inc: 1981.
2. B. Boehm, et al., *Software Cost Estimation with COCOMO II*, Upper Saddle River, New Jersey, Prentice Hall PTR: 2000.
3. D. Ferens and D. Christensen, “Calibrating Software Cost Models to Department of Defense Databases – A Review of Ten Studies,” February 1, 1998.
4. D. Ferens and B. Daly, “A Quantitative Comparison of Popular Software Scheduling Models,” *1991 ISPA Conference Proceedings*, vol. X, pp. SW43-SW59.
5. D. Ferens and S. Stukes, “Software Cost Model Calibration,” *The International Society of Parametric Analysts 17th Annual Conference Proceedings*, 1995, pp. SW57-SW64.
6. R. Jack and M. Mannion, “Improving the software cost estimation process,” *Software Quality Management*, vol. 1, 1995, pp. 245-256.
7. D. Reifer, B. Boehm, and S. Chulani, “The Rosetta Stone: Making COCOMO 81 Estimates Work with COCOMO II,” *Crosstalk*, February 1999, pp. 11-15.
8. *SEER-SEM Version 5.1 and Later User’s Manual*, Galorath Incorporated, March 2000 update.
9. R. Stutzke, “Software Estimating Technology: A Survey,” *CrossTalk*, May 1996, Volume 9, No. 5.
10. *Your Guide to PRICE S: Estimating Cost and Schedule of Software Development and Support*, Mt. Laurel, New Jersey, PRICE Systems, LLC: 1998.