

Auditable Parametric Based Estimating Systems

Abstract

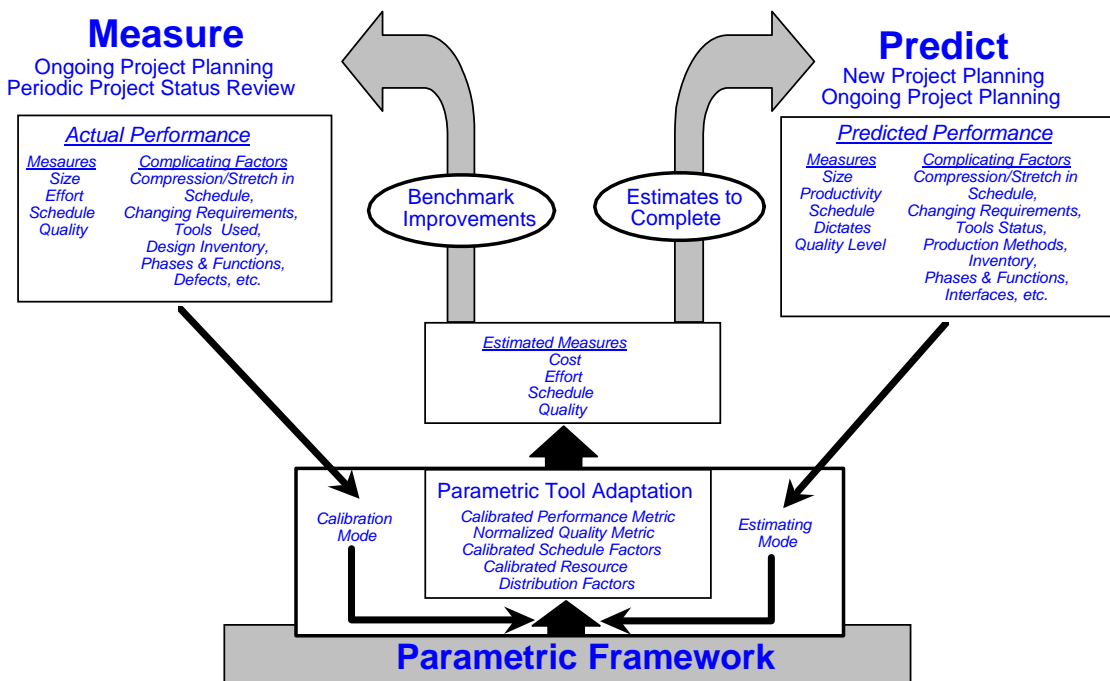
Over the past 25 years, parametric estimating has emerged from a beginning as an occasional tool for quick and cursory analysis to become a serious alternative to traditional estimating for business operations. Part of the transition from analytical tool to business system involves establishing credibility of the parametric based estimating system and its component parts - the parametric tool, the database upon which the tool is calibrated, and the organization infrastructure for use and maintenance. The recent past has seen the development of criteria upon which to evaluate the credibility of a parametric estimating system. Most criteria stress practices for data management, tool adaptation, and system use that conform to documented performance. At the same time, the criteria recognize the dynamic business environment of today, where practices and organization structures can change rapidly. Parametric estimating is viewed as an important element of integrated business operations that efficiently adapt to the most current, accurate, and complete information available. This paper will present characteristics of parametric estimating systems that are audit worthy, or auditable. An example is developed to illustrate the premise that an auditable system is built rather than acquired.

Introduction

An estimating system based on parametric techniques is composed of three primary components:

1. Parametric Model(s) or Methods
2. Parametric Database
3. Implementation, or Adaptation of the Model/Technique to the Database

The figure below illustrates the process by which a parametric estimating system evolves. It shows a dynamic environment of “measure-predict-improve” in constant motion.



Audit of an estimating system that uses parametric methods will naturally involve review of all three of the above components. Each must satisfy whatever audit criteria is established for acceptance before the estimating system can be certified.

The Defense Contract Audit Agency (DCAA) has specified that a contractor must apply the following six criteria before submitting a contract price proposal using parametrics:

1. Logical relationship
2. Verifiable data
3. Significant statistical relationship
4. Fairly accurate prediction
5. Easy to monitor
6. Estimate support

This paper will present an example of a parametric based estimating system designed to satisfy criteria like that listed above. The focus here will be on the second and third components of the parametric estimating system. Model/Method audit deals with criteria of use of logical relationships, and certification that any computerized implementation of the model/method conforms to the documentation of those relationships. This involves a rigorous mathematical evaluation performed on a non-recurring basis. This paper will presume the use of a parametric method certified to be logical.

Parametric Estimating System Building

Many organizations have mistaken the purchase of a parametric estimating model with the building of a parametric estimating system. A similar thing happens with the personal computer (PC) when unprepared buyers, caught in the craze, purchase a system with little idea as to its use. They pay \$2000 to \$3000 for a multi-media system, bring it home, connect the components, turn it on, and then say, "Now what?" It is important to understand that, like effective use of a PC system, effective parametric estimating capability is built, rather than purchased.

Auditable parametric estimating system building involves at least five steps, as summarized below.

1. Survey data sets available for use
2. Extract appropriate information from the surveyed sets
3. Construct a parametric database using extracted information
4. Calibrate parametric tool(s) to the parametric database
5. Use, monitor, and maintain system

1. Survey Data Sets Available For Use

There is no reason to believe that parametric estimating requires collection of any information other than that already present in an organizations archives. Records and reports describing the technical, cost, and schedule aspects of completed projects form a master set from which to extract information pertinent to the specific scope and capability of parametric technique(s) adopted. Such reports are generically classified as follows.

- Project Schedule Reports
- Project Financial Reports
- Project Development Plan
- Project Status Reports
- Inspection and Test Reports
- Test Plans and Procedures
- Project Experience Databases (Metrics)

A benefit of using this type of information is that most, if not all of it, has already been verified. This addresses an important audit criterion.

2. *Extract Appropriate Information From the Surveyed Sets*

In practice, project reports are often very voluminous and intimidating. A key to building a successful parametric based estimating system is being able to sift through mountains of information to capture only the information pertinent to effective estimating. This involves a high degree of persistence and an organized structure for the data collected. If the information search can be categorized, then it is possible to expedite the process. In building an audit worthy parametric based estimating system, three categories of information are usually sufficient. These are:

1. Cost/Effort/Schedule Information
2. Technical Information
3. Project Peculiar Information

Cost/Effort/Schedule Information should include expenditures of both labor and currency, with identification of purchases versus direct costs for the performing organization. Expenditures should be identified by project time phase and class of performer. The time unit can be defined by phase length (e.g. preliminary design) or fixed calendar time (e.g. month), which ever is most suitable to the situation. The class of performer is usually defined by labor category (e.g. quality assurance) or procurement activity (e.g. materials). Project milestones for reviews and deliveries will help to better define and differentiate expenditures by phase. This type of information is common to most estimating systems, making it relatively easy to find and verify. The other two categories of information are generally not associated with traditional estimating systems. However, they usually contain the discriminating information used within a parametric system to produce credible estimates.

Technical Information useful in parametric estimating system building will identify the quantity and quality of end item expected from completion of a project. This information will differentiate a bullet from a missile, a desk telephone from a cellular telephone, and an integrated office network from a world-wide military Command & Control System. The specific technical information collected should match the discriminating capabilities of the parametric tool(s) being used. Among the more popular items of technical information are:

- Measures of Size (weight for hardware, lines of code for software, as examples)
- Technology/Tools/State-of-Art (manufacturing and design processes, as examples)
- Quality (parts and materials specified, test level, as examples)
- Design & Manufacturing Heritage (use of off-the-shelf design and/or software, as examples)
- Function (performance - what the deliverable is intended to do)

Technical Information, like that above, is collected and reported for practically all projects. However, it is not usually confined to one document. In addition, the format for presentation may be specialized to the extent that it is difficult to recognize. A few generic sources for technical information collection will be identified shortly. For now, it should be recognized that technical information extraction is one of the two most challenging, yet important aspects of building an auditable parametric based estimating system. The other formidable hurdle is in extracting Project Peculiar Information.

Project Peculiar Information is a more qualitative set of data about the technical aspects of a project. Here is where end user requirements, applicable standards for development and production, problems or complications in carrying out the project, and skills of project resources can be recorded. Traditional estimating systems often ignore these issues, probably because they are difficult to quantify. Nevertheless, time and again these are the things that are blamed for cost and schedule over-runs of projects. By ignoring them, as we often do when we estimate from the bottom up or by analogy, we practically ensure that the estimate will not match the expenditure. Thus, it is imperative that this category of information be collected and qualified within a parametric based estimating system. The table below provides some examples of how this can be done. The specific qualifiers

used to build any one system should conform to the parametric technique(s) employed and the kind of projects to which the parametric estimating system will be employed.

Project Peculiar Information Qualifiers

Sub-Category	Potential Qualifiers
End User Requirements	Application: Consumer, Industrial, Government, Military, etc. Safety: Of use, In production, Of peripheral items, etc. Criticality of Failure: Minor, Moderate, Major, Intolerable, etc.
Applicable Standards	Consumer, Commercial, Military, Custom, etc. Logistics Support Analysis: None, Consolidated, Specialized, etc. Degree of Configuration Control Degree of Quality Assurance
Problems and/or Complications	Compression or stretch of schedules (by phase) Availability of tools and resources Unexpected events Dependence on other projects
Resource Skills	Experience of people with similar projects Experience of people with design technology Experience of people with production method

Data extraction is performed by selecting the verifiable data to which the parametric tool is sensitive, so that the resulting estimating system will be easy to monitor for accurate predictions. This aspect addresses at least three audit criteria.

3. Construct a Parametric Database Using Extracted Information

Once parametric information has been extracted, it needs to be formatted and stored for easy use and maintenance. Structures should be established for logical sorting and access by related data fields. Many a useful database never materialized due to concern that it would not be as comprehensive as necessary. This is a pervasive trap. Every useful database grows over time to encompass more than its original intent. Growth is a sign of vitality. It should not be combated by adding unnecessary information in an attempt to satisfy every anticipated future database requirement. To do so, increases risk of strangling information digestion. A good rule to follow in constructing a Parametric Database is to optimize the information content to include only as much information as needed for project differentiation. The information fields will correspond to the parameters of the tool(s).

The database must be easy to monitor and be supportive of the estimating process if the parametric estimating system is to meet the monitoring and estimating support criteria.

4. Calibrate Parametric Tool(s) to the Parametric Database

It is difficult to envision a parametric estimating tool being used in an auditable system if it cannot be calibrated. The purpose of calibration is to align the estimating tool to the experience of the user, just as an automobile seat is aligned to the contour of its user. The same seat may require different alignment settings for every user. In estimating, calibration is achieved by aligning only those parameters that are of a qualitative nature to documented performance. The process must treat as hard facts any data that is clearly verified. Such data are not to be re-aligned to match performance. For example, a software development of 100,000 source lines of code would be calibrated as a 100,000 source lines of code data point - not 108,500 or 98,200 or some other number of lines of code. This parameter is not adjusted to fit a result. It is a fact that must be treated as absolute.

A calibration fundamental is to focus entirely on those tool parameters designed for alignment. They are usually few in number. All other parameter values are as extracted from records. With this, the actual performance (cost/effort/schedule) drives alignment of the calibration factors. Subsequent fine tuning can then be undertaken, if desired. Later, this will be illustrated with an example.

The level of calibration is frequently a question in building a parametric estimating system. That is, if the database is structured in a project hierarchy of records accumulated into subtotals which are in turn accumulated into a project grand totals, should the calibration be undertaken by record, subtotal, or project total? The answer to this seems to be yes to calibration at all three levels. Certain aspects of calibration are best carried out at higher levels of aggregation than others. This too will be illustrated by the example presented later. The level of calibration should be governed by the goal of building estimating rationale that is ready-to-use for any anticipated new requirement. The relevant criteria addressed are the presence of significant statistical relationship and accurate predictions.

5. Use, Monitor, and Maintain System

Like any ongoing venture, an estimating system requires careful use and maintenance. Improvements to the system should be made as often as performance monitoring indicates the necessity. Each of the three parametric estimating system primary components has unique upkeep requirements.

The expectation that the parametric database will grow with use has already been cited. As with its creation, the growth should be limited to only that information deemed to be applicable. Perhaps the most important aspect of database maintenance is that the contents be refreshed with new data as it becomes available, while, at the same time, old data is purged. Acceptable audit criteria refers to this process as use of current, accurate, and complete information. This practice should govern all actions associated with parametric database maintenance.

Effective use of the parametric tool(s) requires widespread application within an organization. Usually, an application will only become widespread after it is understood and accepted. And, training is the most effective way to convey understanding. Training of the tool(s) will take several forms, each targeting a different audience. Tool users will need detailed training, while those providing information to or consuming the results of the users can gain necessary understanding with a more general, higher level training course. Acceptance of tools usually results from demonstrated performance. This means monitoring of tool use to articulate positive results and take corrective action when the results are negative. Negative results may have occurred due to mis-application or lack of sufficient tool sensitivity to important cost driving parameters. Mis-application problems can be addressed with more in-depth training and monitoring. Insufficient tool sensitivity usually indicates the need for database growth and evolution of the tool(s) to the increased scope of the database. Thus, negative results are indicative of process vitality. They should not only be expected, but welcomed as opportunities for improvement.

It seems clear that use of the parametric estimating system is key to its improvement. Of course, use is what produces the results for which a parametric estimating system is acquired in the first place. Remembering that the concept of a parametric estimating system is an integrated system of database and tools in active day-to-day use, removal of any component will disable the system. So, while it is possible to take the parametric tool away from its associated database, doing so removes it from the parametric estimating *system*. Any use of the tool outside these boundaries is without audit. Parametric estimating system builders and those who use and audit such systems must be rigorous in ensuring that the tools and databases are used synchronously.

Constant monitoring is a good way to establish practices that meet the criterion of easy to monitor. Refreshing the database helps to maintain the verifiable data criterion, while providing the foundation for improving the process. More significant statistical relationships should evolve to improve the prediction accuracy. When a system is working this way, estimating support is a daily business practice. Therefore, an organization that uses, monitors, and maintains a parametric based estimating system is practically assured of being audit worthy.

Parametric Estimating System Building - An Example

Background

An example of the building of an audit worthy parametric system comes from a case study for a software development company referred to as the Garf Company. Garf is the developer of Information and Training Systems. Projects usually entail purchase of commercial processing equipment and integration of requirements via custom developed software. The company employs a 2167a Software Development Standard for all projects, regardless of the customer (2167a is a Military Standard). The parametric system of this example will be constructed around information extracted from three projects, one of which is in process. The projects are listed below.

Garf Company Parametric Database Projects

- AN/SIM-13: Sonar Operations Training System
- DI-MIS: Warehouse Distribution System
- CRL-MIS: Transportation Control System

Parametric Database Building

Garf company reports were reviewed for information pertinent to a calibration database. The table following summarizes the results of that search. Replicas of some of the actual reports are found in the appendix to this paper.

Results of Garf Reports Survey

Report Title	Information Extracted
<i>Software Experience Database Report</i>	Metrics of completed projects (SIM-12 and DI-MIS only). Project schedule, customer, and functional description. Configuration item core development effort, allocated by phase. Configuration item size in source lines of code, allocated by type, language, and state (new or reused). Allocated project level effort for Systems Engineering & Program Management, Quality Assurance, Configuration Management, and System Test.
<i>CRL-MIS Software Status Report</i>	Component level status. Size (Source Lines of Code - SLOC) in three domains - data (SLOD), new executable (NEW SLOC), reused executable (REUSE SLOC). Core development status by phase. Configuration item core development schedule length.
<i>CRL-MIS Monthly Financial Report</i>	Allocated effort (Labor Months) associated with status report. Allocated project level effort.
<i>Project Development Plan</i>	Qualified 2167a impact on DI-MIS and CRL-MIS customer liaison (non-military customers).
<i>Inspection Reports</i>	Enhanced software application understanding - qualitative measure.
<i>Test Reports</i>	Documented quality performance (Defects per K SLOC) and schedule slippage.

The next step in building the parametric estimating system is to extract information from the above reports to use in adapting the parametric estimating tool(s) to it. The particular tool used in this example is the PRICE Software estimating model - PRICE S. The specific data extracted will be that for which there is both PRICE S sensitivity and estimating system need. The list below summarizes the information extracted for inclusion in the parametric database.

Extracted Report Information
<ul style="list-style-type: none"> • Project, Configuration Item, and Component IDs • Start and End Dates • Software Development Language • Type of Software by Size (SLOC) <ul style="list-style-type: none"> Function: Execution & Character Heritage: New & Reused • Allocated Labor Month Expenditures <ul style="list-style-type: none"> Core Development by Configuration Item Project Level Where Complete

In addition to the above information, the parametric database must contain calibration data from adaptation of the tool to this information. This additional data results from fitting the information to PRICE S. The specific data items are the productivity factor and effort allocation parameters, as described below.

Calibrated Parametric Data
PROFAC for core development by configuration item 5 phase allocation parameters of core development 3 category allocation parameters of project effort

Parametric Tool Mapping

A premise of this paper is that building and using a parametric estimating system does not require collection of new information. Existing sources should contain the needed data. We have exemplified the process of review and extraction. Often, there is one other action required before tool adaptation can occur. It is translation of the nomenclature of the extracted data from its source to that of the parametric tool. That translation is called a mapping, and it must be performed within comparable definition bounds of the source document and the parametric tool. For example, suppose a report contains automobile information with a data field called *drivetrain*. In addition, suppose that data from that source is being extracted for use with a parametric estimating system that is not sensitive to *drivetrain*, but does use a parameter called *transmission-differential subsystem*. Since these different nomenclatures refer to a similar, if not identical assembly, a mapping from *drivetrain* to *transmission-differential subsystem* seems reasonable. On the other hand, a mapping from *drivetrain* to *exhaust system* would not be a fair and accurate mapping. The chart that follows illustrates the mapping from Garf reports to PRICE S parameters.

Mapping From Source to PRICE S

Source Report Data Item	→	PRICE S Parameter
Schedule		
Start Date	→	SDR of Configuration Item
End Date	→	FCA of Configuration Item
Software Type		
data	→	Non-executable SLOC (FRAC)
data, exec., c_dir.	→	Application (APPL)
Number of SLOC		
New + Reused	→	Total Size (SLOC)
New	→	% New Design & Code (NEWD/NEWC)
All Other Factors Nominal	→	Tools, Experience, Interface Machine, Product Knowledge, Integration, & Processor Resource Utilization

At this point, PRICE S is exercised on the above data to produce the calibration information to be incorporated into the database. Recall that there are three categories of calibration items for recording: productivity factor (or PROFAC), configuration item core development allocation parameters (Globals), and project level allocation parameters (other Globals). Each calibration data item category is determined in the order listed. Specifically, the process is:

1. For each configuration item, determine the PRICE S Productivity Factor - PROFAC
 - a) Based on core development phases of:
 - i) Requirements Analysis (2 PRICE S Phases)
 - ii) Preliminary Design
 - iii) Detail Design
 - iv) Code & Unit Test (A portion of PRICE S Code/Test)
 - v) CSC Test (a portion of PRICE S Code/Test)
 - vi) CSCI Test
 - b) For functions of Design, Programming, & Data Engineering only.
2. For each configuration item, determine effort allocation parameters by computing ratio of recorded labor month expenditure to PRICE S estimates (with PROFAC from 1, above) by functional category (1.b).
3. For each project, determine effort allocation parameters for Systems Engineering/Program Management, Quality Assurance, & Configuration Management by computing ratio of

respective project level recorded labor months expended to PRICE S estimates under control of 1 and 2, above.

When these tasks are accomplished, the results are recorded in the database, completing its initial creation. A formatted listing is displayed in the appendix to this paper. At this point, the database is ready to serve as the basis of estimate for any new software project related to those of the database. It is also ready to grow and to be enhanced, since there are limitations to its future utility. Among the limitations of this database construction are :

- It represents a logical and rational approach, but there are other approaches that are probably just as, if not more logical and rational.
- The database most likely contains more information than needed for existing projects, but not enough information for all future projects.
- Candidates for data item growth are:
 - Complicating & mitigating factors parameters (these were set to nominal in building this parametric estimating system).
 - Development tools (set to nominal in example).
 - Development team language & machine knowledge (set to nominal in example).
 - Development team product experience (set to nominal in example).
 - Number of development sites to coordinate (set to 1 in example).
 - Software application (essentially the same for each of the three projects of the example).
 - Component, configuration item, & system integration requirements & volatility (presumed to be the same for each project of the example).

Parametric Database Use

The primary purpose of the parametric database is to serve as a repository of information to guide preparation of credible estimates. At any time, it contains the rationale for estimating projects related to those that fed it. With a properly built and operating parametric estimating system, the rationale exists before the estimating task is undertaken. Too often with traditional legacy estimating systems, the rationale is created during or after the estimating exercise. Rationale created this way often lacks objectivity. Below is the rationale for our example. It is nothing more than a summary of the calibration data for the database as constructed. It contains the performance parameters and allocation parameters for estimating a software system related to those of the database.

Garf Corporation Parametric Database Summary									
<i>Project</i>	<i>PROFAC</i>	<i>Reqmt. Analysis</i>	<i>Prelim. Design</i>	<i>Detailed Design</i>	<i>Code & Unit Test</i>	<i>CSCI Test</i>	<i>Sys. Eng./ Pgm. Mgt.</i>	<i>Qual. Assur.</i>	<i>Config. Mgt.</i>
AN/SIM-13 (Avg.)	8.70	1.17	1.00	0.72	1.23	0.94	1.00	0.33	0.07
AN/SIM-13 (Std. Dev.)	0.09	0.06	0.00	0.02	0.03	0.02			
DI-MIS (Avg.)	8.01	1.32	0.94	1.10	1.46	0.43	0.97	0.34	0.07
DI-MIS (Std. Dev.)	0.03	0.02	0.01	0.01	0.02	0.01			
CRL-MIS (Avg.)	7.96	1.36	1.10	1.10	1.53	0.26			
CRL-MIS (Std. Dev.)	0.02	0.26	0.13	0.07	0.06	0.03			
Database (Avg.)		1.28	1.01	0.97	1.41	0.55	0.99	0.34	0.07
Database (Std. Dev.)		0.10	0.08	0.22	0.16	0.36	0.02	0.01	0.00

The performance parameter (PROFAC) is not averaged since the data suggests that this value differs for military customers (AN/SIM-13) and civilian customers (DI-MIS and CRL-MIS). The former results in a higher productivity capability index (8.7) than the latter (8.01 and 7.96) due to familiarity of the military customer with the 2167a development standard. More liaison with non-military customers is required to ensure understanding of deliverables under the somewhat foreign standard for

them. This added effort diminishes the developers productivity capability somewhat. Thus, the selection rationale for the PROFAC to use in a new estimating setting will be based on the customer.

Most of the allocation parameters cluster within a standard deviation of .1 or less, indicative of reasonable correlation (the criterion of significant statistical relationship). The exceptions to this, Detailed Design, Code & Unit Test, and CSCI Test, display tendencies to the military/civilian groupings characterized for PROFAC. However, with only three data points (projects), this conclusion cannot be adequately tested. Thus, the rationale for allocation parameter selection is to use the average values of the database.

A simple, yet meaningful test of the accurate prediction criterion is to apply the database rationale individually to each project of the database. In this way, a prediction can be compared to the actual effort. The table below summarizes that exercise, showing less than 2% variance between the actual effort and that predicted with the database rationale - very impressive.

	<i>Actual</i>	<i>Predicted</i>	<i>Var.</i>
SIM-13	1826	1783	-2%
DI-MIS	1048	1047	0%
C-MIS*	469	464	-1%
* Includes completed CSCIs only and excludes SE/PM, QA, and CM			

Application Example

The parametric estimating system constructed in this example will now be used to estimate the effort to implement the incomplete fifth configuration item of the CRL-MIS project. At the time of parametric estimating system construction, this item was 10 months into a planned 16 month development cycle. It was expected to yield a software item of 50740 executable SLOC and 12580 non-executable SLOC, all of which required new design and code generation. Both the Requirements Analysis and Preliminary Design phases of the development had ended. Approximately 80% of the SLOC had been accounted for under detailed design, and an estimated 45% of the code had been implemented and tested at the module level. Configuration item testing effort had yet to start, and project level effort was that expended to date.

At month 10, defect rates uncovered during module testing were significantly higher than specified. This raised concern about the overall quality of the completed project, which cast doubt on the viability of the budgeted effort and time required to complete development and deliver the CRL-MIS system to specified requirements. The table below summarizes the effort situation of the project. It shows an incurred expenditure to date (ITD) that is critically close to the amount budgeted for the complete development. It also displays the results of a recently completed Project Management re-evaluation of the effort at completion (EAC). None of the table values were produced with the parametric estimating system. That system will now be applied to provide another opinion to compare to the EAC.

CRL-MIS 5th CSCI Status at Month 10

	Labor Months of Effort
Budget	1085
ITD	919
EAC	1131

The first thing done in the parametric analysis is to use the tool (PRICE S) with the calibration data from the database (PROFAC and allocation parameters) and the configuration item quantitative data (SLOC, New Design, etc.) to estimate the core effort to develop the 5th configuration item. A summary of those results, compared to the breakout of ITD is shown below.

Effort Comparison for 5th Configuration Item of CRL-MIS

	Parametric Estimate	ITD
Requirements Analysis	54.4	48.1
Preliminary Design	55.0	55.1
Detailed Design	87.4	72.4 (80%)
Code/Unit Test	153.6	66.5 (45%)
Configuration Item Test	66.5	0 (0%)
Subtotal	416.9	242.1

There is a conclusion suggested by this comparison that may help explain the anticipated over-run and hopefully, a way to minimize it. It is that the Requirements Analysis phase ended prematurely. Notice that approximately 10% less effort has been expended than that suggested by calibration to actuals for similar items. The result of incomplete requirements definition is often programming and testing of defective software. This seems to be what is happening here since both the Detailed Design and Code/Unit Test assessments of completeness (80% and 45% respectively) closely match the ratio of ITD to Parametric Estimate for both phases (83% and 43%). So, it would not be prudent to continue with detailed design, code, and test of this configuration item until requirements are reviewed and incomplete definitions are identified and corrected.

With this problem identified, the analysis now expands to encompass the entire CRL-MIS project. Using the parametric estimating system calibration data as input to the tool (PRICE S), an estimate, to include all project level tasks is made and compared to the ITD, EAC, and Budget. The table below is a summary of that comparison.

Comparison of Parametric Estimate, EAC, ITD, and Budget for CRL-MIS Project

	Parametric Est.	ITD	EAC	Budget
Requirements Analysis	117.2	118.9	118.9	
Preliminary Design	119.8	126.8	127.8	
Detailed Design	187.3	180.6	189.3	
Code/Unit Test	324.4	251.2	332.9	
Configuration Item Test	132.3	33.5	56.5	
Sys. Eng./Project Mgmt.	290.1	160.5	241.3	
Quality Assurance	42.3	40.1	54.3	
Configuration Mgmt.	9	7	10	
Total	1222.4	918.6	1131	1085

The parametric estimate is approximately 10% higher than the estimate to complete, due mainly to a higher projection for configuration item testing and System Engineering/Project Management effort. For the parametric estimate, these values are based on calibration of the PRICE S tool to a comparable data set, resulting in less than 2% error between estimate and actual. This is the rationale for the parametric estimate. What is the rationale for the Project Management EAC? It is unknown, though possibly influenced by a natural human tendency to minimize negative impact (in this case, the difference between EAC and Budget).

Observations and Conclusion

A carefully planned and constructed parametric estimating system can be an effective, impartial, objective, and consistent aid to decision making and business operation. A parametric estimating database can usually be constructed from data already being collected within an organization. The key is to survey all existing databases to identify the items useful for parametric estimating. In fitting the parametric tool(s) to the database, variations in calibrated parameters should be expected. Searching for explanations for such variations will lead to a better understanding of business processes and products, thereby helping the parametric estimating capability to grow in accuracy and scope of use. In constructing the mapping from traditional data sources to the parametric database, it is more important to be concerned with consistency than rigorous precision. In the end, a consistent and well documented parametric estimating system, calibrated to past performance, is more audit worthy than a very detailed level estimating system driven by few data points and much subjective judgment.

APPENDIX

Garf Corporation Report Replicas Garf Corporation Software Experience Parametric Database