On General Purpose Model Credibility

Evin Stump
Senior Systems Engineer
Galorath Incorporated

Wendy Lee
Systems Engineer / Cost Analyst
Galorath Incorporated

Abstract

In the aerospace/defense community and elsewhere, cost estimation of proposed large scale hardware projects by prospective bidders or sponsors is often done using general purpose parametric models, i.e., models based on statistical analysis of a broadly based collection of historical data. Of primary interest in this paper are the parametric models that provide estimates of costs of hardware development and production for a wide variety of hardware, ranging from simple to complex, especially hardware that may be used in severe and unusual situations and environments, ranging from undersea to outer space. In this paper, we shall call these general purpose models (GPMs).

GPMs are both challenging and expensive to construct, and few organizations attempt them. The usual arrangement is that the organizations benefitting from such models will license them from companies that specialize in model building, such as Galorath Incorporated and its competitors.

Historically, much less sophisticated parametric models go back at least to the ancient Greeks and Romans, but the level of sophistication has increased exponentially in the last 60 or so years, when it was first possible to install a GPM on a mainframe computer, and then only a few years later (contrary to many predictions) placed on a personal computer.

In the early years of GPMs, some cost analysts rejected them on grounds that they were “black boxes” of unknown content that could output anything at all just by “spinning the dials”. That fear gradually dissipated when 1) it was realized that *any* estimating process can be corrupted by mindless “dial spinning”, and the phrase “garbage in, garbage out” was born, and 2) parametric models were seen as providing less subjective, more consistently accurate results than competing processes, especially in early project phases when the most significant decisions are being made.

Today, many project sponsors and contractors have cost analysts who are highly proficient in use of these models. They and their managements tend to put a high level of trust in the model outputs, although they may (and should) want to do careful calibrations and run cross-checks using other methods wherever possible. Still, some general purpose model users, especially new users, are concerned about use of parametric models, and they often direct questions to model builders, seeking one kind of assurance or another.

In this paper, we examine the nature of a GPM, and discuss the process, part science, part art, of building a credible one. In the course of this discussion we propose useful questions that a concerned model user should ask the developer in order to get the most confidence in the model results. We also look briefly at a few questions that we have been asked that are not particularly helpful, and we explain why.

The Nature of General Purpose Models (GPMs) and How They Get Built

Today in aerospace/defense and other companies as well, and also in government agencies, there are hardware development projects that have exceptionally demanding goals; goals simultaneously related to cost, schedule, and high levels of product performance. The trends for these goals are generally in the direction of ever higher performance, ever lower cost, and ever faster schedules, all preferably without ever increasing risk – it's quite challenging!

To keep these trends moving in the right direction, they are supported by a constant drumbeat of research into new technologies, sponsored by both government and industry. There is also a sometimes slightly less visible drumbeat of efforts to reduce waste and increase efficiency in projects by improving the methods used to perform them. These include six sigma, lean manufacturing, design to cost, risk management, and computer-aided engineering analysis, drafting, manufacturing, and communications.

Modern, sophisticated hardware concurrently uses a mix of new and older technologies and methods. Still older technologies and methods are dropped when they are replaced by newer, better approaches. Because of this continuing state of technological flux, there is a constant need for new cost information, both for the new technologies and the interfaces of the new technologies with the older, more stable ones. Stakeholders meet these needs using a variety of cost estimating methods. Traditionally the methods used come under these headings:

- Analogies
- Bottom up (aka "grass roots")
- Parametric models

These headings are a bit misleading because at root virtually all cost estimates are analogies[1]. We think we have a notion of what something new will cost only when we can compare it with what something extant and "similar" is known to cost.[2] We recognize that the less the similarity, the greater the likely error in estimating. As a very senior cost engineer once told one of the authors, "In order to estimate the cost of something you first have to know what it is, then what it's like."

The differences between the three above listed estimating methods arise from how we create the analogies. Methods bearing the label "analogy" without further qualification are most commonly comparisons between known costs of a previous set of project activities and partly similar new project activities. Obviously analogies depend on the existence of at least one suitably analogous project. Analogies are rarely if ever exact, usually requiring that substantial adjustments be made using other methods, most commonly bottom up.

---

[1] Analogy: A similarity between like features of two things that are otherwise dissimilar, or a comparison based on such similarities.

[2] This all but universal truth about costs does not necessarily apply to prices. Prices are affected by novelty, time to market, competition, patents, and many other factors.

Bottom up estimates are typically done by the people who will ultimately be doing the work, or their immediate task leaders. Based on their expertise and the data available these estimates can range from carefully drawn estimates requiring extensive analysis and calculations to pure guesses. If the subject matter is unique with few or vague precedents, pure guesses are more likely. Whether or not the subject matter is well understood, bottom up methods are notorious for being protective of the estimator. (Cox and Schleier, 2010) describe a common problem with bottom up estimates that are schedule based, as most development phase activities are.

> *"Since task times follow a skewed distribution…and have unique properties, completion times cannot be estimated with precision. Nevertheless, an estimated time must be provided. Resources operating in traditional project environments, therefore, are forced to protect their careers by providing times with appropriate safety that will permit them to survive management 'adjustments' and deliver on their promises."*

Another fault of bottom up estimates where the work is traditional is that they are often too high due to financial safety concerns. On the other hand, if the work is rich in non-traditional content then they are often too low, because of engineering hubris and concerns about being underbid.

Parametric models are based on historical cost data and use of statistical methods to interpret it. They range from rough "rules of thumb" to the most sophisticated GPMs. A typical rough rule of thumb is "It costs $10,000 to put a pound into earth orbit." One problem with rules of thumb is that they sometimes have a short shelf life. Currently, we are hearing that it will soon be possible to put a pound into orbit for only $1,000.

Three recognizable intermediate steps between rough rules of thumb and GPMs are 1) mathematical rules of thumb, 2) industrial engineering models, and 3) simple parametric models.

A mathematical rule of thumb goes a bit beyond simple relations such as dollars per pound. An example is a rule that says that if you know that an electric motor of ¼ horsepower costs $100, and you need to know the cost of a ¾ horsepower and don't have time to research it, you can get pretty close by using:

$$\$100 * \text{sqrt}((3/4) / (1/4)) = \$173.$$

Such simple rules often apply to relatively simple hardware having a single major cost driver.

A special kind of parametric model known as an industrial engineering model is sometimes used for labor intensive production estimating, and occasionally for estimating costs of drafting and other rote engineering tasks. These models require a database of average times for a worker to complete particular simple tasks. Acquisition of such a database is normally accomplished by "time and motion" studies of workers actually doing the work. To complete an industrial engineering estimate, the work is first broken down into a number of elementary actions for which "standard times" have been recorded. These times are summed to determine a raw figure for total hours required. This raw value may then be corrected using various multipliers, such as for break times, or use of less skilled labor.

Simple parametric models are mostly those arrived at using ordinary least squares (OLS) methods or some variation of them. OLS is a statistical method that transforms a set of historical data into a polynomial cost estimating relation (CER), generally of order either 1 (linear) or 2 (quadratic). Under certain logarithmic transformations, nonlinear power law cost expressions can result, for example:

$$\text{cost} = ax^b$$

where a and b are constants and x is a cost driving parameter.

OLS (also known as regression analysis) is a powerful tool that can create a multivariate CER. Because of inherent design complexities, many hardware costs are dependent on several variables. For example, experts in the subject have found jet engine costs to be related to at least the following parameters (Younossi, *et al*, 2002):

- Maximum Thrust
- Engine Dry Weight
- Shaft Power
- Specific Fuel Consumption
- Thrust-to-Weight Ratio
- Overall Pressure Ratio
- Rotor Inlet Temperature
- Design Life
- Year Entering Production
- Afterburner (Y/N)

This list of ten parameters is by no means exhaustive. Additional parameters may be needed to make cost distinctions between various usages and types of jet engines, for example fighter aircraft versus helicopters versus large transport aircraft. Jet engine designs differ significantly for these three aircraft types. Other adjustments need to be made when the engine is a derivative of an existing engine versus being a totally new design.

While many useful OLS models have been built in both industry and government, there are definite limits to their capabilities. Here are some important ones:

- The number of data points acquired must be equal or (preferably) exceed to the number of cost driving parameters employed for an OLS mathematical solution to be possible. For example, for a jet engine estimate employing the above ten cost driving parameters, the model builder must find at least ten jet engine projects for which the appropriate values or design options can be identified. This may not be a problem for a government agency that buys dozens of different models of jet engines and keeps good records, but it would be a problem for a jet engine manufacturer that builds only five engine models and has no access to the larger government database. For a GPM builder who builds nothing except cost models, it can be daunting, because for competitive reasons much cost data is held to be confidential and will not be shared.

- The selected cost driving parameters must not be significantly correlated. If they are, model behavior will be erratic and accuracy will suffer. While tests for some types of correlation are available, selecting an uncorrelated set of parameters that explain costs well is often difficult.
- The mathematical sign of each parameter in the CER must be consistent with physical realities, i.e., plus if an increase in the parameter should increase cost, negative otherwise.
- The p-values and other commonly used statistics of regression must be consistent with the values typical of a good data set. See for example (Wonnacott and Wonnacot, 1970), (Lewis-Beck, 1980), or any good college level text on applied statistical methods.

We have now looked briefly at the following techniques of estimation:

- Analogy
- Bottom Up
- Parametric
  - Rough Rules of Thumb
  - Mathematical Rules of Thumb
  - Industrial Engineering Models
  - Simple Parametric Models

We are now ready to look at GPMs, which by convention fall under the parametric heading. Before examining what makes a GPM different from all of the above models, it is useful to ask to what extent it can be the same. The answer is: To the extent needed, *a GPM may benefit from use of any or all of the above estimating methods!* Why? Because of the extreme diversity of the types of hardware for which estimates will be provided, and the amount, type, and quality of the data available, no single estimating approach can deal with all of the possibilities encountered in practice. Here are some of the estimating challenges that a GPM must typically meet:

- **Many types of hardware --** A GPM will not estimate just one type of hardware such as a jet engine. It may be need to estimate dozens of types of hardware, hardware that may all be part of the same system. Here are just a few examples of the variety of hardware types that might need to be estimated by a GPM for a single project:
  - Integrated Circuits
  - Antennas
  - Jet Engines
  - Electrical Cables
  - Electronic Power Supplies
  - Fuel Pumps
  - Airframes
  - Instruments
  - Servomotors
  - Pressure Vessels
  - Cockpit Furnishings

- **Integrated subassemblies --** A GPM estimates multiple types of hardware not only as standalones, but as parts of an integrated subassembly. In complex projects tegration costs are seldom negligible. They may include costs of transportation, mounting or attaching, wiring for electrical power, wiring for signal input and output, connections for flow of fluids, or mechanical connections to transfer motion or forces.[3] Some kinds of integration require time consuming precision alignments or calibrations, as for example in optical assemblies.
- **System level integration** -- Often there are additional costs for integration at the full system level, and test and evaluation of the complete system. This must be accommodated when it is appropriate.
- **Total ownership cost** -- Often there is interest in total ownership cost (TOC) or life cycle cost (LCC), which includes development and production, deployment, and cost of operations and support (O&S) over a number of years. Among other things, O&S requires estimates of rates of production, extent of deployments, operational tempo, maintenance staffing and labor rates, product reliability, product maintainability, and product service life.
- **Allocations and factors** – Allocations and factors are a special kind of estimate that is generally of the bottom up type, that is, based on expert opinion. GPM developers soon learn that while licensees like to have their costs reported at high levels, such as total development cost, total production cost, etc., they frequently also like to have them sliced and diced different ways at lower levels as well. This presents a real challenge to GPM builders. One choice is to do the basic estimating at these lower elements and then sum the results to get the higher level costs. The other choice is to estimate at the higher levels and then allocate down to the lower levels. The latter choice is almost always the better one because more and better data is usually available at higher levels. A considerable problem is that allocations and factors that are good for one GPM user might be terrible for another user, because of different accounting or labor labeling practices.[4] In creating reasonable allocations and factors in a GPM, there is no substitute for wide industry experience and / or dialogue with GPM users.
- **Charts & reports --** Users of GPMs generally want a variety of charts and reports, some simplified and some detailed. The number of such reports and charts is usually more than fifty and could reasonably be as much as one hundred. These add considerably to the cost of building a GPM.
- **Platforms –** Some types of hardware are used on more than one "platform." A platform is a vehicle or a location where the item is installed. Different platforms typically impose different stresses on the equipment, and therefore may require different levels of reliability.

---

[3] For example, a major airframe manufacturer may have airframe subassemblies built all over the world and shipped to its main plant for assembly.

[4] A frequent problem is differentiating between labor and material costs. Many contractors label as material everything that they buy, regardless of its labor content, and as labor everything that their own employees do. A problem can arise when there is a major subcontract that has significant labor content. The subcontract can be estimated properly with respect to labor versus material only when the prime contractor treats it as if his own people would do the work. Another problem that commonly arises is due to treatment of overhead cost. Contractors have many different ways of handling overhead costs. A GPM must have the flexibility to deal with all of them.

- **Reliability –** Reliability is tied to platform, certainly, but it is also tied to mission duration, availability, survival probability, and perhaps other factors. Generally the higher the reliability, the higher the cost. Attributes related to reliability that must often be considered are maintainability, ruggedness, and useful life.
- **Universal Drivers –** Recall that in the discussion of OLS models as applied to jet engines, the parameters used were physical features or requirements such as maximum thrust, engine dry weight, engine life, etc. A GPM may beneficially use this approach for some hardware if the data is available, but keep in mind that for OLS the data points available must at least equal the number of cost driving parameters selected. Because of data collection problems this approach is sometimes not feasible. What is often more feasible is a "universal drivers" approach. In this approach, the question is, can we name a set of universal parameters that apply to a very large number of hardware items without a need for a large number of data points? The answer is yes, often we can. There are many such methods. See the Appendix for an example that uses universal cost drivers based on only three data points.
- **Keeping up –** A GPM is generally designed to estimate hardware that now exists and also as much as possible currently emerging hardware. Otherwise the model could be obsolete the day it is released. But will it still be adequately accurate in a year? Five years? That's a question that always needs to be asked. It is made necessary by the rapid rate of change of some technologies. In the integrated circuits industry, one of the most rapidly changing industries, the industry trade association issues a roadmap of expected changes, usually for at least two years into the future. The roadmap can be used to make estimates of cost effects for two, sometimes more, years ahead. From (Younossi, *et al*, *ibid*), it appears that an update about once every five years or so is appropriate for jet engines. The point is that GPM builders must stay current with what is going on or their models will become obsolete and possibly useless. Each technology has its own peculiar (and usually variable) rate of change.
- **Calibrations –** A GPM as delivered may consistently give good results for Company A, low results for Company B, and high results for Company C. This is because companies have different ways of doing business, and a single model cannot capture all of the nuances. To keep model users happy in spite if these systemic errors in estimation, model builders must provide means for users to "calibrate" their models, which boils down to correcting them to account for differences from the norm that is assumed by the model. Calibration is usually enabled by providing means for inserting multipliers at various places in models, and at various levels of cost. Users invoke these multipliers by comparing a cost produced by the model to what they believe is an equivalent cost that occurred in a recent project or an average over several recent projects.
- **User Interfaces –** Several user interface issues are important for GPMs. Leading ones are:
    - **Efficient screens –** Users want screens that are easy to read and that minimize mouse clicks and movements needed to get the job done. Having to frequently change screens or windows is both a nuisance and an error prone process.
    - **Clear Help –** Model Help is unclear when:

- You can't easily find out what will happen when you issue a command, and especially how you can back out of a command.
- You can't quickly find the help that you want.
- Different help sections are in conflict.
- Help fails to give valid and current specific examples.
- Help language is vague, garbled or prolix, or has unexplained acronyms.

- **Flexible input/output** -- The model should readily accommodate various ways that the user is likely to want to make inputs or outputs. Examples are importing data from a spreadsheet, and exporting results to a spreadsheet. Custom reports are another example.
- **Backward compatibility** – One of the most difficult issues that a GPM builder must deal with is backward compatibility. Users naturally expect a fair degree of forward compatibility, and GPM builders struggle to provide it without going overboard and attempting to predict the far future. Backward compatibility is also an issue. For some types of software such as operating systems model builders may feel free to abandon backward compatibility after a few years on grounds that users value new features more than consistency of process or result. This is harder to do with GPMs because users often like to be able to get consistent answers to both old and new projects. Yet changes in technology or market conditions are constantly pushing for changes in the estimation basis. For example, the first personal computers to come on the market (the IBM 610 in 1957) cost $55,000 apiece, and only 180 were sold. Today, a respectable personal computer can be had for under $1,000. Because rigorous observation of backward compatibility under all circumstances would result in complete stasis of a model, with ever decreasing accuracy of estimating, GPM builders have to make accommodations that are as comfortable as possible to users. One approach is to periodically create a new database that enables a revised estimation methodology. A family of these databases is shipped with the model, enabling the user to quickly retrieve an old estimate. If the cost change is small, sometimes the user is simply advised of it when a new release is made.

Gaining Confidence in Your GPM

You have heard by word of mouth or otherwise that it may be to your benefit to acquire a GPM and use it on your future proposals. Your IT department makes one available to you. What next?

New users of unfamiliar software frequently jump in and start using it, relying on intuition and perhaps occasional quick glances at help files. They gradually acquire enough skill to justify use of the software for their often limited purposes. Many or even most of the features of the software remain unknown to them and they could care less. For software that is not overly complex, or user functionality requirements that are not overly diverse, this is often a satisfactory outcome.

This approach is definitely not recommended for GPMs because the almost certain outcome is unrealistic cost results. Usually a minimum of four to ten days of training is needed for new users to gain the competence needed to get reasonable cost results. The formal training supplied by GPM builders is the most thorough, but adequate on the job training may be available within a cost analysis team if it already has members fluent in the software.

If cost analysis team members are all new to the software there may well be questions as to the validity of the cost results. Sometimes adequately warm and fuzzy comfort with regard to this issue may be available from friendly cost analysis teams that use the GPM and are willing to share information. The GPM builder will also provide some limited assurances on request, but there may still be some doubts because what the GPM builder can disclose may be limited by internal confidentiality policies and / or external agreements. Cost information can be valuable!

The single best thing you can do to quell continuing doubts is to select a few historical (completed) projects that your team has estimated, projects that you won in competition, and projects for which good records were kept.[5] Assuming that you are sufficiently well trained to use the GPM competently, use it to estimate these historical projects. After making an initial estimate, use the model's calibration features to align the cost output with the project results.

While doing this, here are some points to keep in mind:

- Use actuals with no uncertainties applied.
- Keep in mind and correct for inflation effects, especially effects on labor rates.
- Keep in mind the skill mix that actually was used on the project and its effect on average labor rates and project team productivity.
- Look for significant changes in the technology since the historical project was completed. For certain types of hardware, such as integrated circuits, there can be significant changes in as little as a year. For sand castings, which were made in China earlier than 600 B.C., significant technology changes may take many years.[6]
- Look carefully at what subcontractors did. Did you give them significant change orders? Did they keep their promises? Keep in mind that it is not unknown for subcontractors to hide major technology problems initially and that they only become visible sometime in preliminary design or even later, especially for sophisticated new technologies.
- Look carefully at sponsor directed changes and their effects on cost and schedule.
- Look for significant internal finance department adjustments in overhead rates part way through the project.
- Don't just look at top level costs. Go through the work breakdown structure and check each work element as a standalone with and without subsystem integration costs applied.
- Solicit team member anecdotes about major unexpected problems that occurred, especially significant redesigns and retests. Consider the likely effect that these had on costs and schedule.

---

[5] It is an unfortunate fact of life that most major projects have significant information gaps in the permanent records maintained for them. To get a clear picture of what actually happened, it is usually necessary to interview a few of the leading project players.

[6] And then again they may not. Considerable sand casting work is now going to China because of their low labor rates. Many changes in costs are due to global competitive and other conditions, not just changes in technology.

- Look for significant planning failures, especially manpower planning failures resulting in always having to play catch-up due to labor shortages, especially highly qualified labor. Consider also other significant failures such as unavailability of needed facilities, equipment, access, or services.
- Look for instances of sponsor provided facilities, equipment, labor, or access not being provided as promised with respect to timing or suitability.
- Look for major instances of the "standing army effect." This is enforced idling while being paid of part of the project team because another part hasn't done its work according to schedule. It can be very expensive. There can also be a "standing equipment effect" or a "standing facilities effect".
- GPM builders are reluctant to include a Quality of Management parameter in their models because it is an impolitic thing to do. Nevertheless, keep in mind that poor management decisions are often a root cause of cost and schedule overruns.

Aside from adequate training and re-estimating your old projects, what else might you do to gain confidence in you GPM? Here are a few things:

- Directly compare your GPM results to results from your older estimating methods using simple hypothetical examples. Keep in mind that comparisons can be odious for various reasons, but the results should not be excessively different. And your GPM estimating labor hours should be much lower than the hours for your traditional methods.
- Have all more or less equally qualified members of your estimating team independently estimate the same project elements of known cost using both your GPM and your existing methods. There usually will be less scatter in the GPM estimates, and the GPM results should be more accurate.
- If your experience with your GPM is still modest, invite a consultant experienced with the GPM to oversee an important upcoming estimating task. Usually, this will more than pay for itself.

Some Less than Useful Questions that We Have Been Asked

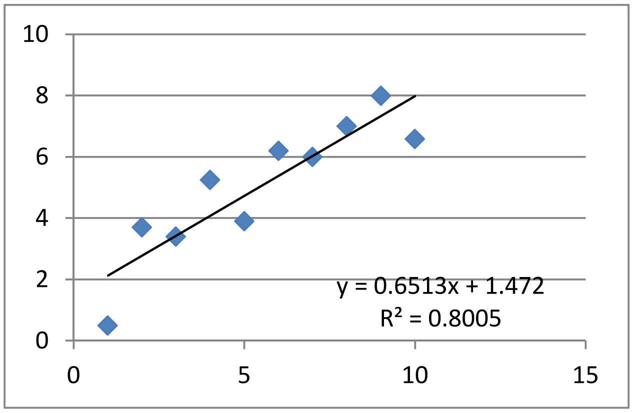
Even if you have done everything suggested in the previous section there may be someone on your estimating team who remembers at least a bit of Stat 101 and wants to ask the GPM builder a question based on that remembrance. Sometimes these questions are quite reasonable, and sometimes they are not. The "number one" less than useful question that we are asked about our models is, "From how many projects did you collect data for use in building your model?"

Before giving our answer to that question, we quote here the answer given by a very astute statistician from the 18th century, Thomas Bayes, founder of that increasingly important branch of statistics known today as Bayesian statistics:

> *"Now that the errors arising from the imperfection of the instruments & the organs of sense shou'd be reduced to nothing or next to nothing only by multiplying the number of observations seems to me extremely incredible. On the contrary the more observations you make with an imperfect instrument the more certain it seems to be that the error in*

> *your conclusion will be proportional to the imperfection of the instrument made use of. For were it otherwise there would be little or no advantage in making your observations with a very accurate instrument rather than with a more ordinary one, in those cases where the observation cou'd be very often repeated: & yet this I think is what no one will pretend to say."*

While this sounds a bit stilted to modern ears, a little study will reveal his meaning. Bayes inveighs against the notion that you can correct for instrument error and errors of the "organs of sense" by collecting more data. Rather than collect more data, one should collect data with a better "instrument."



When the experiment is a simple one, such as trying to ascertain the effect of various treatments of fertilizer on potatoes by weighing the potatoes, it is better to use an accurate scale rather than one not so accurate. But potatoes have only one metric: what they weigh (there may be others but they can most likely be neglected). Hardware projects on the other hand can have dozens of metrics, many of which interact with other metrics. This confuses the picture and makes it more difficult to understand and analyze, but adding more projects to the list of observations in no way improves accuracy.

About word R^2 scores: (R^2 is also called the coefficient of determination): Cost analysts accustomed to building OLS models often gauge the quality of a model by its R^2 score, or possibly more astutely by its "adjusted" R^2 score. When asked how high this score should be to assure a "good" model, a common answer is 0.8. Sometimes 0.7 is given, and analysts have been observed using models with scores as low as 0.5. Plotted nearby is an arbitrary OLS model that has an R^2 value of 0.8 approximately. The largest single error is 325%. The smallest is 0.52%; the average is 44.2%. *Half of the errors are over 20%.*

Yes, you can sometimes get a nice regression line or multi-dimensional surface, but typically many points will not lie close to it, as revealed by that ubiquitous R^2 score. So what you are doing is estimating the projects that lie close to the regression line, leaving not adequately estimated the many that lie distant.

The key is not how many projects are in your database, but how many parameters and which ones it takes to get a low estimating error across a particular project family. In principle, with the right parameters and only two or three good exemplar projects, you can build a good model for a project family, if you don't make the family too inclusive.[7] To assure the happiness of its customers, that is where a GPM builder must focus. And that explains why a good GPM requires not a large sample size but a large number of parameters.

[7] A too inclusive family is one that includes known "problem" projects, or projects basically intended to be research efforts, where either the technical outcome is fixed regardless of the cost, or the cost is fixed regardless of the technical outcome.

One more point with regard to the question about how many projects did GPM builders use to build their models. A GPM may be designed to estimate a hundred types of hardware. The best data for a given type came from one group of projects, while the best data for a different type came from a different group of projects. Moreover, the methods of analysis may have been different because of the type of data that was available in each group. For one type of hardware there may have been 50 data points, and OLS was used. For a different type of hardware (e.g., the Hubble telescope), there may have been only one data point, requiring an altogether different estimating approach, an approach that included deep inquiries into what happened on the project and what has happened on the most closely similar projects.

The "number two" less than useful question we are sometimes asked about our estimating methods is: "From which specific projects did you get data for your model." It would be impossible for us to give a truly meaningful answer to that question. We have on occasion for benefit of users named a few projects from which we have culled information. But generally speaking, confidentiality prevents us from saying specifically which projects provided which information.

At root of these and other less useful questions is the desire to gain confidence in the GPM and to understand its applicability to your estimating environment. That of course is both understandable and admirable, but perhaps this paper can give readers insights into better questions to ask.

Useful Questions a GPM User Should Ask

Experienced and confident users of GPMs seldom have need to ask a question of the model developer other than questions of this nature: "Why doesn't the model do X?" where X is some pet input, output or calculation that the user favors for his or her cost analysis work. These questions are always welcomed by GPM builders and in the past have been the basis for many helpful model changes. GPM model builders can anticipate much useful functionality, but hardly all useful functionality.

There are also occasions when a user thinks that a "bug" has been found in the model. Model builders always are happy to receive bug reports, because if the bug is real it provides an opportunity to improve the model. If the bug is merely perceived and not real, a user reports is an opportunity to help the user better understand the model.[8]

Many less experienced GPM users, on the other hand, are most often concerned about the "validity" of the model. They want to know what makes a GPM "valid", and in particular, is the model "valid" for the project they are currently estimating?

Certainly the logical answer always must be: a valid model consistently supplies "credible" cost estimates for all types of hardware it claims to estimate, under all conditions that it claims to represent. It would appear that the proof that a GPM does this can ultimately come only from its users, users being defined generally as contractors who are bidding for projects. Given the confidentiality rules that usually prevail, only a project winning user can compare what the model predicted with what the project actually cost.

---

[8] Contrary to some popular notions, perception is often NOT reality.

But in practice, how can one test this? Most major projects for which GPMs are used unavoidably contain uncertainties, uncertainties that are not fully resolved until the project is completed. GPMs customarily allow their users some latitude in expressing uncertainties (also called "risks") in their inputs. These uncertain inputs inevitably result in ranges (as opposed to point estimates) in the model outputs. Unfortunately bids must be fixed amounts, not ranges, so the model user must pick a particular point in the output range to bid. The point picked may be the statistical average, the statistical mode, or the statistical median of the output range. Or it could be some other point altogether, such as the 40^{th} percentile, based perhaps on a calculation or an intuition about win probability.

Generally, the outcomes of a bid can only be winning or losing. If the result is a loss, the GPM user may gain little or no insight into the "credibility" of his GPM estimate. The estimate used in the bid is not necessarily wrong. It may well be that the winning competitor has a technical approach that the project sponsor liked better.

If the outcome is a win, there usually is no immediate indicator of model "credibility". That indicator does not arrive until the project ends, if then. It may not arrive at all if the sponsor directs many changes over the course of the project. It may become virtually impossible to accurately sort out the merits of the original bid.

Model builders often get "911" calls from users, calls usually occasioned by the urgent need to produce a bid amount coupled with uncertainty about whether the model is being used properly. This is especially true when the model's cost result is higher than what the proposal manager thinks the project should cost. In a real life 911 call before any answers are given the caller is usually asked to answer some questions that will help in supplying recommendations or prescribing actions. Among the questions a model builder might ask:

- **Are your GPM analysts trained in the use of the model?** If the answer is no, then the user may have a serious emergency situation. Modern GPMs are necessarily quite complex, and even though GPM builders exercise considerable care in making the models "user friendly" they are not all that friendly to untrained users, even if those users are already widely experienced in putting bids together using analogy and bottom up methods.
- **Are your technical people working with your cost analysts to supply the most accurate answers possible to model technical inputs?** GPMs that are used to estimate hardware costs typically contain many inputs of a technical nature, and correct values for these inputs are important to cost accuracy. Unless the technical people provide close support to the cost analysts these inputs are likely to be uninformed guesses.
- **Are your financial people providing current and accurate information about labor and overhead rates, and is this information being appropriately applied in the model?**
- **Are your subcontracts management people providing current and accurate information about subcontract costs, and is this information being appropriately applied in the model?**
- **Are all make versus buy issues resolved and does the model accurately reflect these resolutions?**

- **Have you tried using the model to estimate some of your past projects so that you can more accurately calibrate it?**

Based on these and other answers the GPM builder usually can be helpful in getting your bid satisfactorily completed but can truthfully answer the "validity" question only in this manner:

> *"Your model output will be a range of possible cost outcomes, some more likely than others. The width of this range will depend entirely on the amount of risk you have input to the model. Should you be awarded the contract and be allowed to perform it essentially in accordance with the original project plan, it is approximately 95% probable that the final cost will lie between the 30th and the 70th percentiles as predicted by the model. If, however, it lies outside that range, before questioning the model's validity you should first check all of your risk assignments and compare them to what actually happened. It is very likely that the risks were understated. We would be happy to supply you with a list of ten major projects that will validate these claims of cost accuracy. Because of confidentiality we cannot disclose some details about the projects, but the data we do present will have been certified accurate by a leading firm of public accountants."*

Appendix

We demonstrate here a non-OLS method for estimating costs with the number of data points less than the number of cost driving parameters. There are eight cost drivers and three data points. There are many possible variations on this method.

	Parameter Range		Raw Values			Adjusted Values			Weight
Cost Driver	**Min**	**Max**	**Engine 1**	**Engine 2**	**Engine 3**	**Engine 1**	**Engine 2**	**Engine 3**	
Weight of Product (lbs)	0	5,000	3560	4058	4673	0.712	0.8116	0.9346	0.62671
Weight fraction steel	0	1	0.35	0.4	0.3	0.35	0.4	0.3	0.325746
Weight fraction aluminum	0	1	0.45	0.4	0.45	0.45	0.4	0.45	0.016423
Weight fraction exotic alloy	0	1	0.2	0.2	0.25	0.2	0.2	0.25	0.034562
Factory Automation	0	10	4	6	5	0.4	0.6	0.5	0.674247
Quality Level	0	10	10	9	9	1	0.9	0.9	0.088518
Integration Complexity	0	10	8	8	9	0.8	0.8	0.9	0.251947
Production Team Capability	0	10	8	7	10	0.8	0.7	1	0.085211
	Unit Cost ($M)		3.674	4.188	4.688	1.202475	1.397838	1.428231	
	Cost ratios		1	1.139902	1.275993	1	1.162467	1.187743	
		Error	0	0.019412	0.074301	0.093712636			

Because we have used eight parameters, we would need at least eight data points to use the OLS method, but we don't have eight – we have only 3. But let's assume that the three that we have are quality data points, and we know that the projects were well run and that they are representative of the kind of programs we want to run in the future.

The columns headed Parameter Range express the minimum and maximum values of each parameter that we expect to encounter. The ranges for the last four parameters are arbitrary. Note: for simplicity we have ignored the fact that the three weight fractions must sum to unity.

The columns headed Raw Values are the data that we have for the three engines. The adjusted values in the columns with that heading are fractions of the design ranges. In the column headed Weight we first elicit from subject matter experts a suitable numeric weight in the range (0,1) for each parameter. Then we randomize these values 0.25 in each direction, or to the end of the probability range.

We next run a Monte Carlo simulation on the weights until the cost ratios estimated closely match the actual cost ratios between projects. In the simulation shown the weights selected result in a total absolute error of 9.3%. If the simulation is allowed to run a long time, an error less than 1% can be achieved.

There are many possible variations on this method, not all of which require Monte Carlo simulation. Such methods can often be the basis of useful models using universal cost drivers and as few as two well selected projects. In principle, the number of cost driving parameters is unlimited.

Note: The above model is not used in any Galorath product. It is merely an example of possibilities for multi-parameter models that do not use OLS related methods and that can use smaller sample sizes to arrive at credible estimates. There are many such possibilities.

Bibliography

- Cox and Schleier, *Theory of Constraints Handbook*, McGraw-Hill, 2010
- Lewis-Bech, *Applied Regression: An Introduction*, Sage University, 1980
- Wonnacot and Wonnacott, *Econometrics*, Wiley, 1970
- Younossi, *et al*, *Military Jet Engine Acquisition*, RAND, 2002