

# Core Estimating Concepts

William Roetzheim, Level 4 Ventures, Inc.

**Abstract.** Understanding the core estimating concepts will help you understand any of the currently available estimating tools and provide you with the framework you need when building new models for your particular problem domains. This article strips off the domain specific layers to get at the basic skeleton that underlies estimation in general.

## Estimating Concepts

Most estimating articles, and tools, focus on domain specific models, benchmark data, and approaches. But for all labor related activities there are some generic concepts that underlie estimates for any of the types of work to be performed. These fundamental concepts apply whether you are using commercial parametric estimating tools or home-built Excel based models. User configurable cost estimating tools can be configured using these core concepts to support estimates for any labor driven work, or even for projects consisting of fundamentally different types of activities, even if the tool originally ships pre-initialized for a given domain.

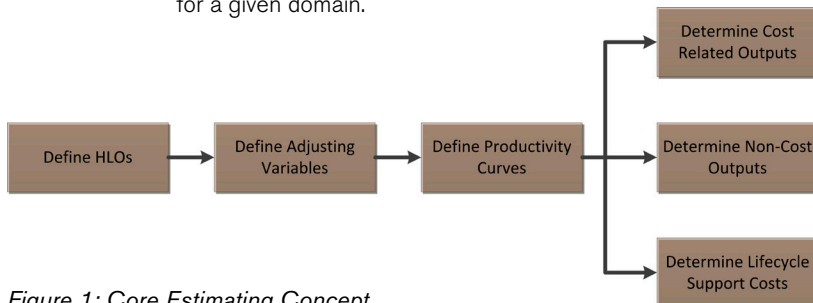


Figure 1: Core Estimating Concept

Figure 1: Core Estimating Concept, provides an overview of the estimating process at a sufficiently high level to ensure that it applies to estimating within any labor driven problem domain.

Step one in the process is to identify one or more High-level Objects (HLOs) that have a direct correlation with effort. The HLOs that are appropriate are domain specific, although there is sometimes an overlap. Examples of HLOs include yards of carpet to lay, reports to create, help desk calls to field, or claims to process. In activity-based costing, these would be the cost drivers. HLOs are often assigned a value based on their relative implementation difficulty, thereby allowing them to be totaled into a single numeric value. An example is function points, which are a total of the values for the function point HLOs.

HLOs may have an assigned complexity or other defining characteristics that cause an adjustment in effort (e.g., simple report versus average report). It is also typically necessary to have a technique for managing work that involves new development, modifications or extensions of existing components, or

testing/validation only of existing components. Various formulas or simplifying assumptions may be used for this purpose. For example, in the case of reuse the original Constructive Cost Model (COCOMO) I model reduced the HLO size to:

$$HLO = HLO * (.4DM + .3CM + .3IT)$$

Where DM is the percent design modification (1% to 100%); CM is the percent code modification (1% to 100%); and IT is the percent integration and test effort (1% to 100%).

Step two is to define adjusting variables that impact either on productivity, or on economies (or diseconomies) of scale. The productivity variables tend to be things like the characteristics of the labor who will be performing the work or the tools they will be working with; characteristics of the products to be created (e.g., quality tolerance) or the project used to create them; and characteristics of the environment in which the work will be performed. The variables that impact on economies or diseconomies of scale are typically things that drive the necessity for communication/coordination, and the efficiency of those activities. These adjusting variables are important both to improve the accuracy of any given estimate, and also to normalize data to support benchmarking across companies or between application areas.

Step three involves defining productivity curves. These are curves that allow a conversion between adjusted HLO sizing counts and resultant effort. They are typically curves (versus lines) because of the economies or diseconomies of scale that are present. Curves may be determined empirically or approximated using industry standard data for similar domains. Curves may also be adjusted based on the degree to which the project is rushed. In any event, procedures are put in place to collect the necessary data to support periodic adjustment of the curves to match observed results, a process called calibration.

The outputs of the process are driven by the needs of the organization. These outputs can be broken down into three major categories:

### 1. Cost (or effort, which is equivalent for this purpose):

In addition to the obvious total value, most organizations are interested in some form of breakdown. Typical breakdowns include breakdowns by organizational unit for budgetary or resource planning purposes; breakdowns by type of money from a generally accepted accounting principles perspective (e.g., opex versus capex); or breakdown by work breakdown structure elements in a project plan. These outputs will also typically include labor needed over time, broken down by labor category. These outputs are generated using a top down allocation.

**2. Non-cost Outputs:** Non-cost outputs are quantitative predictions of either intermediate work product size, or non-cost deliverable components. Examples include the number of test cases (perhaps broken down by type), the engineering documents created with page counts, the number of use-case scenarios to be created, or the estimated help desk calls broken down by category. These outputs are typically created using curves similar to the productivity curves, operating either on the HLOs or on the total project effort.

**3. Lifecycle Costs:** If the estimate is for a product to be created, delivered, and accepted then the cost and non-cost items above would typically cover the period through acceptance. In most cases there would then be an on-going cost to support and maintain the delivered product throughout its lifecycle. These support costs are relatively predictable both in terms of the support activities that are required and the curves that define the effort involved. For many of them, the effort will be high immediately following acceptance, drop off over the course of one to three years to a low plateau, then climb again as the product nears the end of its design life.

Understanding these basic concepts, it is clear that for a given system there may be many different estimates that need to be prepared and combined. Each aspect of the work that involves different HLOs, different adjusting variables, or different productivity curves is really a different model. But all of the models rest within a consistent framework, and in fact, can run within the same tool. There is another dimension of the estimate we need to consider: project lifecycle, or time.

For most projects, it is impossible to completely and accurately define the end product to be delivered. In fact, I would argue that the only way to completely avoid uncertainty in the end product is to have an exact model of the desired results before you start, and it is unusual to have such a model available. In fact, most of the effort spent on projects is on a progressive elaboration of the baseline description of what is to be ultimately delivered. As shown in Figure 2: Progressively Elaborated Baseline," the baseline of what will ultimately be delivered is progressively elaborated throughout the life of the project. Using software as an example, the requirement specification elaborates the functional baseline; the design elaborates the requirement specification; and the code elaborates the design.

As a project moves through this process of progressive elaboration, the estimation models also progress forward (see Figure 3: Estimating Lifecycle). At the most obvious level, as you understand the problem more you can more accurately decompose the work to be performed and prepare an estimate. However, there is another phenomenon at work. The actual estimation model components will change as you move through the process. For example, the HLOs that are used to define the product(s) will change, becoming more and more granular as you move forward. At the high-level estimate stage you might think in terms of a new screen including supporting back-end processing and middleware communication components; at the scope estimate you might be looking at a screen, a table, and a new service; and at the validation estimate stage you might be talking in terms of stored procedures to be written. They are all different perspectives of the same functionality that will ultimately be delivered, but with different levels of granularity. However, the core components of Figure 1 are the same for all of these estimates.

Not only are better estimates possible as you move through the project life, but the primary reason for doing the estimate will change over time. Take a look at Figure 4: Estimating Purposes. Early lifecycle high-level estimates are often used for demand management. Projects are examined for feasibility and selected based on ROI or other financial measures that require estimates to perform the calculations. Scarce resources are allocated to

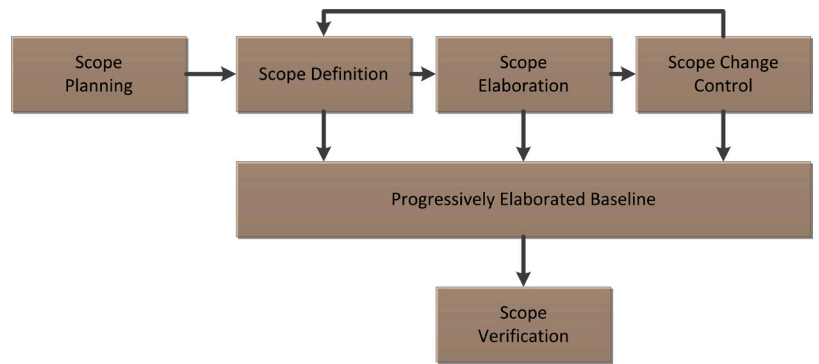


Figure 2: Progressively Elaborated Baseline

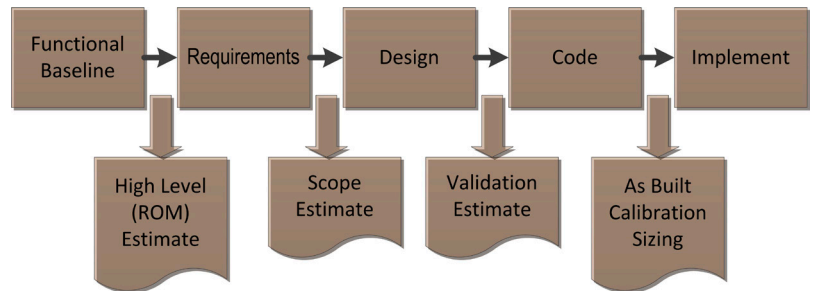


Figure 3: Estimating Lifecycle

support planned projects based on these demand estimates. One characteristic of high level estimates is that a significant percentage of the projects that are estimated (as high as 90% in some cases) are never started. Once a project is at least partially funded and the requirements are better understood and defined (i.e., the baseline has been progressively elaborated one level), then a scope level estimate can be prepared. In many organizations, this is called a "commit" estimate because this will be the estimate used as a basis for measuring project success going forward. The scope level estimate defines the project baseline estimate. Changes in scope are then estimated and, if approved, those estimates are used to modify the baseline. When the project is complete, an as-built sizing is performed to update the organization historical database and for calibration.

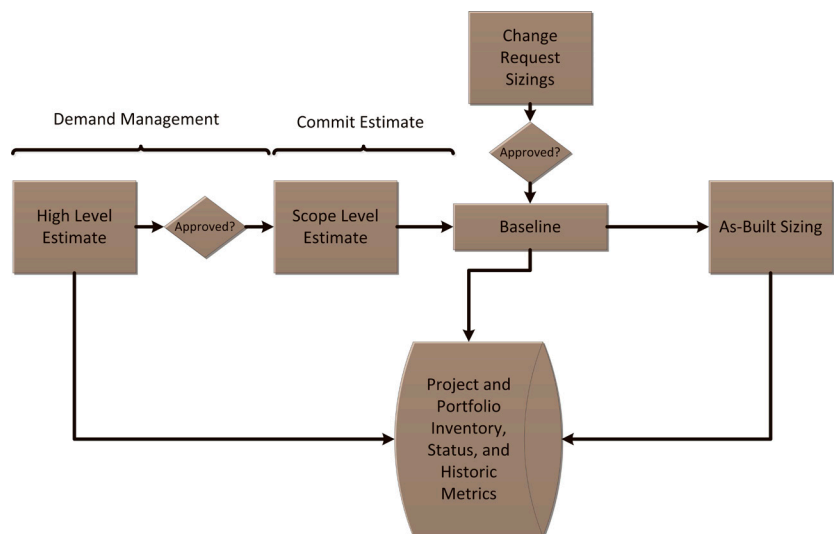


Figure 4: Estimating Purposes

One final core concept of cost estimating is worth discussing: The difference between an estimate and a budget. An estimate is defined as the most likely outcome of a probabilistic event, taking into consideration everything that is currently known about the project. However, the estimate does not include risk, an important component of the project budget. As shown in Figure 5: Estimating versus Budgeting, the estimate defines a starting baseline. Your risk management process (shown at the top of the figure) will then determine the necessary funds for contingency funds and risk response funds. Risk response funds are planned expenditures designed to reduce negative risk or enhance positive risk (oppor-

tunities). Risk Response Funds will always be a cost to the project. Contingency funds are monies set aside to deal with risks that are known but uncertain. Generally, these will be a net cost to a project, although in some situations where risk management has identified some significant positive risks, they may actually reduce the project budget. Finally, the organization will normally want to include a management reserve to allow for unknown-unknowns, or risks that are not discovered until later in the project life.

**Putting it All Together**

Let us take a look at how all of this fits, starting with a slide prepared by the Naval Center for Cost Analysis and presented by Mr. Bryan Flynn at the 43rd Annual DODCAS [1]. As shown in Figure 6: DON Cost Estimating Standard, the DON standard approach aligns well with the approach just described. We will look at it step-by-step, using some examples to explain the process.

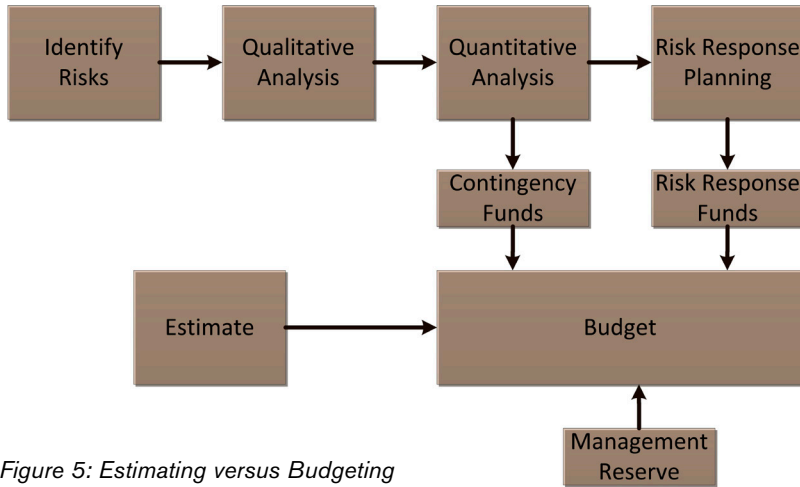


Figure 5: Estimating versus Budgeting

**Step 1: Establish Needs With Customer**

While not directly addressed in this article, this project initiation step is actually the most critical, yet the most often overlooked. I often say that good software cost analysis is 90% stakeholder management, and 10% math. And the key to stakeholder management is understanding the needs of the stakeholders.

**Step 2: Establish a Program Baseline**

Here we are reviewing the business requirements and acquisition strategy (perhaps captured in a cost analysis requirements description), identifying cost drivers or HLOs of this article; and identifying risk areas (the start of risk analysis). For example, in

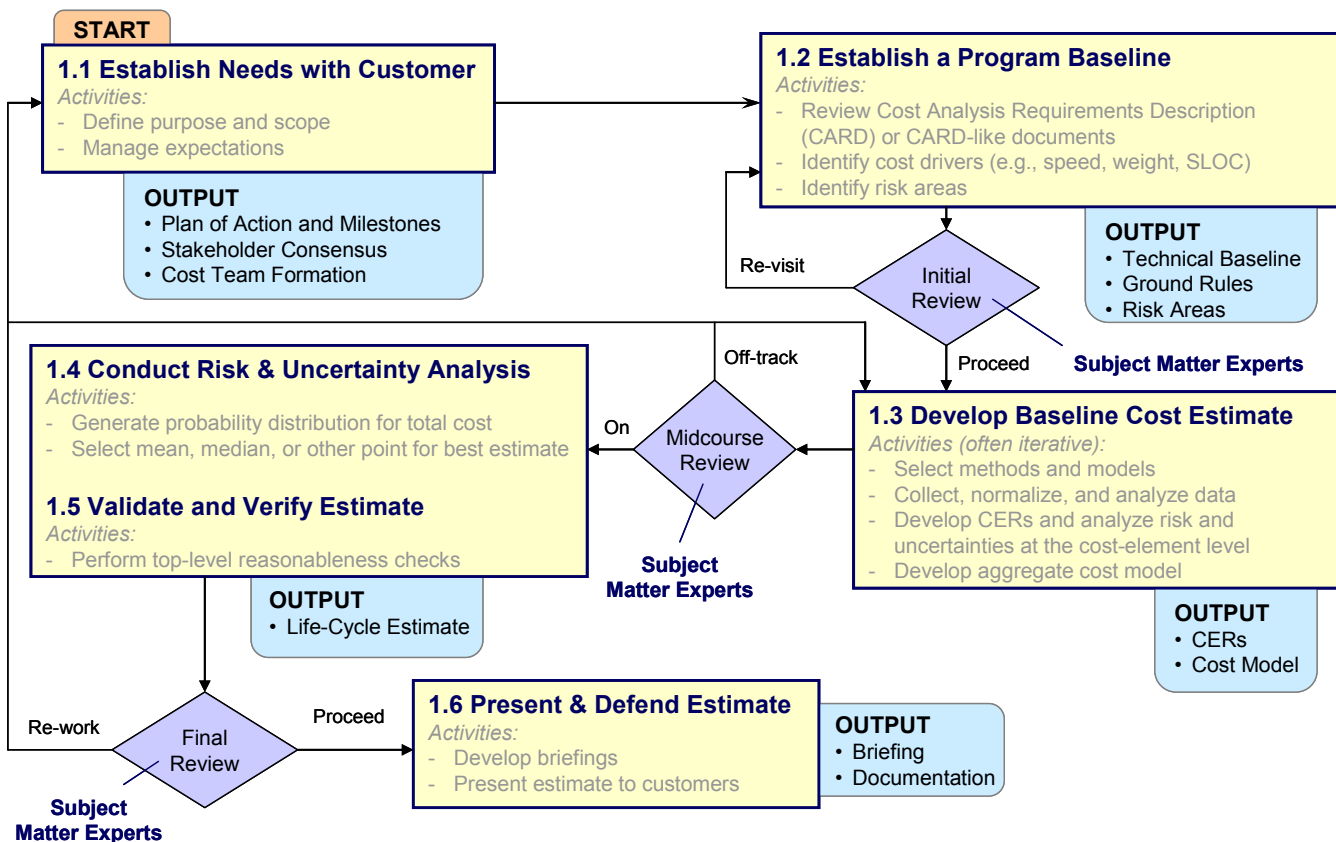


Figure 6: DON Cost Estimating Standard

conducting an analysis of a large DoD ERP implementation, we looked at the available requirement document and determined that the most logical HLOs would be Reports, Interfaces, Conversions, Enhancements, and Workflows. We not only collected together the count of each, but also assigned a complexity value to each (very low, low, average, high, very high) and differentiated between those that were new versus those that were modified. For the modified objects, we estimated the extent of the modification (low, medium, high). In this case, we had historic information that allowed us to estimate both the relative effort for each type of HLO, plus the spread between very low to very high complexity for each type of HLO.

### Step 3: Develop Baseline Cost Estimate

The methods and models that are mentioned here are our productivity curves. What we want is models or methods that will allow us to convert between HLOs and effort. Or more broadly, we might say that we are looking at cost curves to convert between HLOs and cost, assuming that we can develop models encompassing non-labor cost driver equations.

The activity of normalizing data discussed here actually happens at multiple points in the process. First, HLO types are normalized relative to each other through some form of relative weighting in terms of effort (or cost). Second, the cost curves are normalized through project specific adjustments, our adjusting variables.

The cost estimating relationships from the figure are at the heart of the allocation process used to generate our cost and non-cost related outputs.

For the ERP estimate that we are using as our example, we first want to estimate the total effort. For this we start with a suitable productivity model based on the lifecycle being used and the historic data set used for the analysis. The resultant equation is of the form:

$$\text{Effort} = \alpha * \text{Size}^{\beta}$$

Where  $\alpha$  and  $\beta$  are the constants of the model and Size is the normalized total of the HLO values. We then look at project and organizational specific adjustments to  $\alpha$  and  $\beta$ . What we are really interested in here are differences between this project/organization and the historic projects that we used. A couple of good sources to look for potential changes and their likely impact on the variables are the COCOMO II environmental variables and the IFPUG General System Characteristics, although those are by no means the only valid sources.

### Step 4: Conduct Risk and Uncertainty Analysis

The activities described here deal with probabilistic variances in the cost estimate based on uncertainty in the estimation process itself. While these are certainly one source of risk, they are not the only source of risk. It is probably more generically correct to follow the PM-BOK approach described in this article, in which an allowance is added to the estimate to allow for risk mitigation activities, risk contingency funds based on the expected value of the risk factors at work, plus some form of management reserve based on the risk tolerance of the organization and the nature of the project.

### Step 5: Validate and Verify Estimate

A key mistake many novice estimators make is to bury their head in their spreadsheets and end up with results that go against common sense. In the Naval Aviation field, we would have talked about the necessity for a pilot to, "Get their head out of the cockpit"

Of course, just because an estimate goes against common sense does not mean it is wrong. I have seen many situations where the models were right and common sense was wrong. But it does mean that you should take another look to make sure you are not making an error of some kind.

And of course, the validation of an estimate may go beyond a gut check. It is often possible (and useful) to attack the problem using two or more different approaches and to then see if the results converge. For example, you might compare a parametric estimate with a bottom up estimate, or you might prepare two estimates using different HLOs as the sizing input. An estimate by analogy is often a good validation approach. This basically involves finding one or more other projects that is similar to this project, adjusting for any differences, and comparing the adjusted historical values to the current estimate. Another approach that is sometimes used is to compare the results from two or more commercial estimating tools.

### Step 6: Present and Defend Estimate

Yes, of course this is necessary. But what is also necessary is the step of updating the estimate as additional information becomes available throughout the life of the project.

## Conclusions

My goal in writing this article was to define estimating in terms of the fundamental concepts that would pertain no-matter what type of estimate you were creating and no matter what tool you were employing. This understanding of the big picture is useful both in understanding how estimating models and tools work, and also in developing new models or tools for domains where existing models do not exist. ♦

## ABOUT THE AUTHOR



William Roetzheim is founder and CEO of Level 4 Ventures, Inc. He has written 27 published books, more than 100 articles, and three columns. He has been a frequent lecturer and instructor at multiple technology conferences and two California universities. Mr. Roetzheim has an MBA, is an IFPUG certified function point counter, is a Certified Cost Estimation Analyst (CCEA), and has both a PMP and RMP designation by the Project Management Institute.

## REFERENCES

1. Flynn, Bryan: "DoD/DON Acquisition Instructions and DON Cost Estimating Standard," presented at the Department of Defense Cost Analysis Symposium, The Lodge at Williamsburg, Virginia, 19 February 2010.