

Estimate Before, During, and After the Software Project

A common misperception about software development is that the estimator's job is complete after a project's parameters are set. On the contrary, proper estimation should continue throughout the entire project lifecycle, as it is essential for PMOs to [accurately estimate data](#) to increase the value of their PPM tools.

Here are the two key reasons continuous estimation is so important:

First, one must account for the inevitable changes to “conditions on the ground.” As a software project moves forward, more information is gathered (i.e., requirement changes, unexpected obstacles). By re-evaluating the data at every stage, we can make better, more accurate forecasts.

Second, the data collected on one project can help to validate or discredit estimates on another — perhaps in real-time. The more accurate the data collected from your estimation, the better data you'll have for your project management tools to answer questions such as, “Should we give ourselves 6 months or 7 months to complete this project?” and “Should we commit 5 or 6 of our software developers to this project team to get it done on time?”

Does all this sound daunting? It doesn't have to be. Here are three basic rules to help you and your PMO get the most out of estimating efforts — before the project begins, during implementation and after the project has been completed:

1. Conduct feasibility estimates before the project begins

Feasibility estimates are an essential first step to any project, helping to reduce and clarify risk, as well as improve planning and [resource allocation](#). More importantly, these initial estimates will tell us if project success is truly achievable. (With the wrong parameters, a project may be doomed before it even begins.)

Here's how it works: feasibility estimates are performed by taking a high-level overview of project parameters (including size, schedule and cost) and comparing those parameters to historical project averages. If the data suggests the project is not feasible, we can report back to planning committees to set more realistic expectations.

Ideally, a Project Management Office should make this process so easy and intuitive that anyone — from the sales desk to the C-suite — can perform a quick, reliable analysis on a smartphone or tablet. (A two-minute feasibility estimate might prevent two years of project overrun.)

The following (View A) is a sample feasibility estimate — conducted using mobile, web-based estimation software — that flagged a risky project. The second chart (View B) illustrates precisely why those project parameters were deemed risky, based on historical averages.

View A:

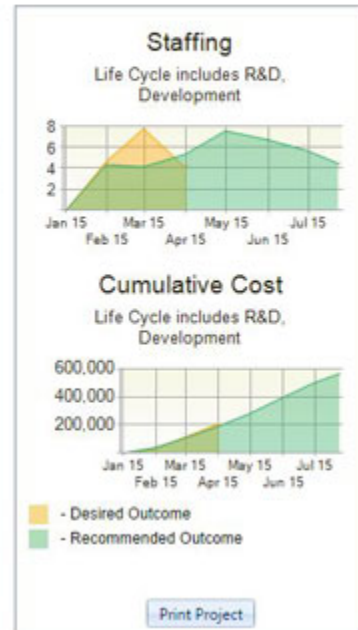
- ① Name the Project
- ② Modify the Template (Optional)
- ③ Define Desired Outcome
- ④ View Assessment

Desired Outcome: Risky

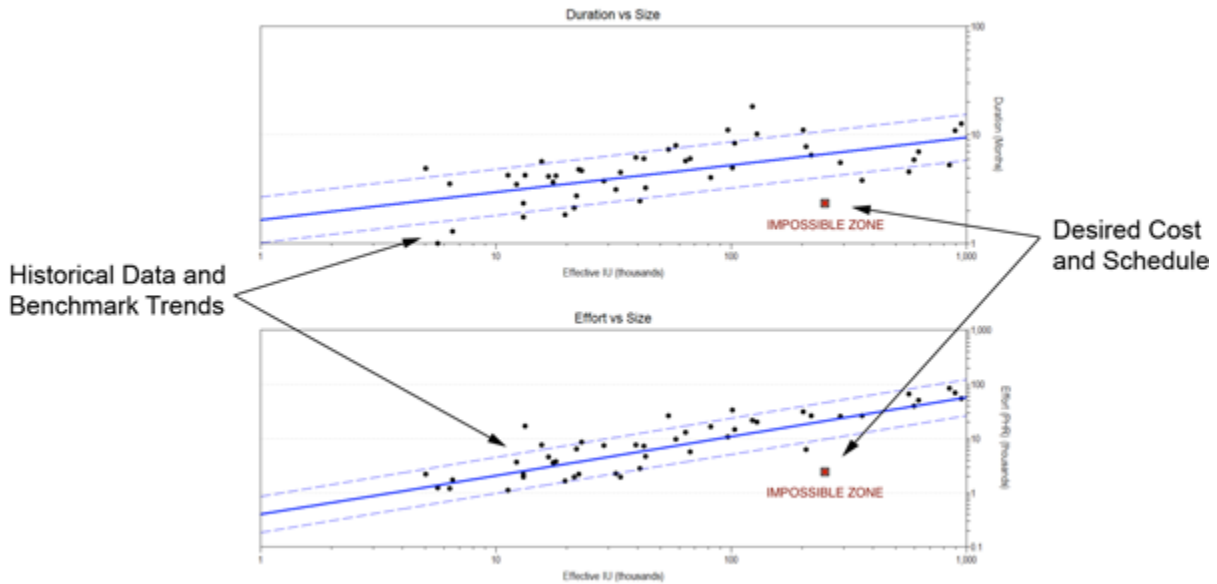
Conservative
Risky
▼

	Desired Outcome	Recommended Estimate
R&D Start	<input type="text" value="1/1/2015"/>	1/1/2015
Effective BusRe	<input type="text" value="19"/>	19
Schedule (Months)	<input type="text" value="3.00"/>	6.7 Months
Cost (\$)	<input type="text" value="250,000"/>	582,941 USD
Average Staff (Full Time Equivalent)	4.8 FTE	5 FTE
Effort (Hours)	2,500 Phrs	5,829 Phrs

Previous: Define Desired Outcome
Calculate
Save Project



View B:



2. Run continuous forecasts during the project

After a project begins, [continuous forecasting](#) is essential for keeping it on track. At this stage, we're no longer working with purely abstract concepts or prognostications, but rather concrete facts about the implementation. As the project progresses and we learn more about its internal dynamics, we can make more accurate estimates that reduce our risks even further.

Continuous estimation here can also help PMOs control stakeholder expectations — especially when PMOs are brought in as contractors mid-project to assess and track an implementation's progress. Take, for example, a

software quality manager who's asked to track the progress of an 8-9 month old software project. The customer — a major medical equipment supplier — has complained of receiving low-quality deliverables. With continuous estimation, our software quality manager is quickly able to determine the problem: schedule. Because an initial feasibility estimate was never conducted, the developers had committed to an unrealistic schedule, which fell further behind with each new requirement added from on high. (The project, originally estimated at 250K lines of code, has ballooned to more than three times that size.)

Estimation further reveals that software reliability is declining even faster than schedule; forecasted reliability is now only 55% of the team's initial quality targets.

By comparing the current data with [historical benchmarks](#), our software quality manager demonstrates the continuing impact of requirements growth on both quality and schedule, and she convinces her bosses to take the appropriate corrective actions.

3. Create benchmarks after the project is completed

In the example above, our software quality manager uses historical data from completed projects in her portfolio to analyze a current one. This goes to show the importance of recording data and establishing benchmarks after a project has been completed. Historical data translates into better feasibility estimates in the future; it's a closed-loop process.

Here's how it works: a major integrator wants to improve estimation for one of its accounts — a large multinational company. The integrator recommends the company spend 60% of its improvement efforts on calibration activities, which include assembling a historical data sample of more than 50 projects to create a better benchmark. After implementing these new calibrations, the company performs 35 estimates and compares the results to actual project performance. Of the 35 calibrated estimates, 29 are within +/-10% of the actual, 32 are within +/-20%, and only 3 are off by more than +/- 20%.

	Number of Projects	% of Projects
Number of Projects within +/-10% variance	29	83%
Number of Projects within +/- 20% variance	32	91%
Number of Projects above +/- 20% variance	3	9%

Such results dramatically illustrate the benefits of calibrating a company's estimates by using its own [historical data](#). In fact, the bigger the project, the more these project estimates seemed to benefit from calibration.

What this specific example doesn't demonstrate, however, is how the various projects in a PMO's current portfolio can affect one another in real time. In such cases, the impact of estimation is even more pronounced, since a PMO is attempting to optimize the distribution of a finite number of resources.

In Conclusion

With accurate estimation and the right process, you and your PMO can reduce risk, maximize efficiencies, and capture essential data for use with PPM tools that will lead to future successes. The key is estimating continuously throughout the project lifecycle — evaluating and analyzing data before the project begins, during construction, and after the project is complete. The more accurate the data, the more useful it will be for your project and the entire portfolio.