# From Estimate to Commitment

by Paul Pagel, 8th Light

# Introduction

While there is little that is more important in the software world than our concept of the future, that concept—like memory—can prove difficult, chaotic, fuzzy, and surreal. The future is an idea, an experiment, and a gesture; a guess or attempt. It can be tricky to accurately predict or explain, and despite scientific, mathematical, and organizational theories or measurements that might help us more precisely predict what happens next, creative thinkers and craftsmen are in fact often working from a sense of timing, a vision, or a shape based on experiences and information that are taken from the past. For this reason, it is even more important that we determine the strategies and methods that might best aid in narrowing down our future timelines and agendas.

The level of uncertainty will always depend on experience, so estimating, therefore, is the ability to place a value on that uncertainty to the best of one's current knowledge and assessment. Due to this subjective nature, estimating is very personal. An estimate is a human-made, fact-based fiction that we tell ourselves about the time commitment necessary to complete a task as a function of not only the project's inherent difficulties, but also our own capacity to combat them.

Because they are subjective to personal experience, estimates can be abstract and challenging to define for an entire project or team. One person's subjective estimate cannot easily or perfectly be assimilated by another. There are a number of science and math-based theories, however, that can assist us in gaining the highest accuracy possible. Some of these theories are: PERT calculations, granularity of scope, and standard deviation. All three of these tools are important to learn because they can work together to minimize error in calculating for future outcomes.

# PERT

The most common mistake in estimates is that they are often based on an "ideal day." The ideal day measures the amount of work one can accomplish if all assumptions are correct and the project does not encounter any new problems. The concept of an ideal day is derived from Plato's world of the forms. Plato's forms suggest that the ideal, uninterrupted day of productivity is a myth based on optimism that clouds our minds and causes us to estimate only the known, while ignoring the uncertain and risky circumstances that will inevitably interfere. When estimating we need to be aware of the relationship exists between problem solving and problem finding.

The practical nature of a craft is such that we slowly uncover layers of the problem by creating and solving, only to discover that we were unable to fully comprehend the entire problem from the start. In other words, new evidence—which is encountered only through action and process—inevitably points our solution in a previously unforeseen direction. The dream of an ideal day encourages us to ignore these process-based problems, since there is no risk or uncertainty built into the estimate. All of our problems are covered up and lost in the big picture. This idea suggests that we need to learn to factor unknown risks into our assessments in order to arrive at a reasonable and pragmatic estimate.

Historically, one method that has proven successful in doing this is the "Program Evaluation and Review Technique" (PERT). From its onset, PERT was designed to make sense of very complex projects. It was first used in the late 1950s by the U.S. Navy to develop nuclear submarines, and was later used to plan for the 1968 Winter Olympics in Grenoble, France. Thankfully, few of us will need to plan for the uncertainty involved in creating something as perilous as the first nuclear submarine or a multi-national sporting event, though it is comforting to know that PERT could handle it if we needed it to.

PERT focuses on time and task dependencies to predict a working schedule for a project. The process itself is a large and exhaustive network, including several PERT charts showing schedules and flow diagrams that connect task-dependent chains. For our purposes right now, we will only focus on one portion of the process: the estimating technique.

At its heart, the PERT estimation technique is quite simple. Given a task at hand, we are asked to provide three separate estimates: the optimistic, the realistic, and the pessimistic.

1. The optimistic estimate assumes that everything proceeds as planned. If everything matches expectations perfectly, then clouds part to reveal a bright and sunny day. This is your ideal time estimate, as dictated by your "ideal day" scenario.

2. The realistic estimate is your "normal" estimate. This is your gut feeling—taking your experiences and past knowledge into account. Intuition (or "educated guessing") is a difficult thing to communicate, but it is oftentimes (for an experienced craftsman) the most accurate factor in an estimate. Because of this fact, we will later emphasize the realistic estimate in calculating what we call a "weighted mean," so that neither our most optimistic nor pessimistic estimations can skew our calculations.

3. The pessimistic estimate is where Murphy's Law comes into play, which says that anything that could go wrong does go wrong. This is where all of the difficulties and complexities that we might have feared manifest themselves—along with several others that we hadn't previously considered.

Before we look at a case study of PERT, a quick note about estimation units. We are going to follow Mike Cohn's principle of estimating in "story points," which are arbitrary and metaphorical units of complexity.

For example, I know that task A is twice as difficult as task B, so it will be assigned twice as many story points. The simplest task will be a single story point by default. Later on you can equate these units to “real” time by calculating how many of your story points you can complete in a set amount of time. For our purposes, it is easier to estimate in terms of complexity than it is in terms of time.

So let’s say that I want to write a web application feature that emails a completed to-do list to everyone who is interested. This is how I would work through the PERT estimates:

First, I would record my optimistic estimate. In this scenario the web application allows me to use its clean interface to get all of the information out of my to-do list as well as all of the email addresses that are registered to receive a copy of my to-do list. I then pay to use an email service to send the list out on time. Voila! I had minimal work to do and can now spend the rest of the day at the beach. My optimistic estimate in this case would be two (2) story points.

Realistically, the to-do list’s interface may require a bit of experimentation for me to figure out how to get my data out of the website. The documents could potentially be outdated, but I also know that customer support should be able to help fill in the gaps. Working with email vendors may cost more money than I initially thought, so realistically I will have to send the emails myself. Writing a program to do that shouldn’t be difficult, but it will take time. I will have to code an error case or two to make sure everything is usable. This scenario is four (4) story points.

Finally, examining my pessimistic estimate is where I am forced to ask: What if everything goes wrong? What if the interface has bugs in it, there is no documentation, and the customer support is horrible? Then, the email program also has to worry about the recipients not losing the email to spam folders, etc., etc. This nightmarish scenario provides me with a pessimistic estimate of ten (10) story points.

With these three estimated numbers, I can use the following PERT calculation:

ESTIMATE (OR WEIGHTED) MEAN = (Optimistic + (4 * Realistic) + Pessimistic) / 6

For my to-do integration story: (2 + (4 * 4) + 10) / 6 = 4.667

So, for this example I arrive at the weighted mean of 4.667 story points as my estimate. Notice that this is one point higher than what my realistic estimate had been, which suggests that even that estimate could have been too optimistic.

PERT is an estimator's weapon against a single person's natural disposition to be optimistic or pessimistic. Either prejudice can throw off expectations for when the work will be complete. Excessive optimism can create a situation in which an individual or team is constantly having to over-achieve to meet expectations, and likely becoming over-worked. However, if there is too much pessimism in your estimates, an individual or team will not be making the most of its resources, and will likely seem under-achieving or apathetic as a result. My example shows how a PERT estimation can combat each of these against each other to produce a much more precise estimate.

## Risk Assessment

Implementing an effective risk assessment analysis allows us to determine the accuracy of our project's estimated schedule, and thus give us a clearer conception of what to expect from future progress. Risk assessments should be done on a spectrum, weighing the time and energy it takes to produce estimates against the value of the estimate to choose the appropriate risk levels for the project. In certain instances, it is more logical to make casual guesses for estimates, while others require detailed, time-based commitments. We will use the simple heuristic called estimate minimalism, which states that we want to use the least amount of energy to produce the most accurate estimates needed.

The two tools we use to control the risk of our estimates are granularity of scope and completion confidence. Scope granularity reduces the risk of the unknown by focusing on smaller, more detailed aspects of the project. Completion confidence employs the standard deviations of our three PERT estimates to calculate a confidence level.

## Granularity of Scope

Let's first examine estimate granularity. Making an estimate more granular is the process of restricting the scope, especially by isolating the unknown or risky factors. This lets us push the risks of the unknown to the future, when we will undoubtedly be more prepared to confront them.

We can envision this process as similar to how we find driving directions on Google Maps. When zoomed out, we can make approximations about the duration of the drive based on the length between two points on the map. However, after slightly increasing the granularity by zooming in halfway, we notice that the route is not as direct as we would have liked. Perhaps we will have to take two different highways that intersect at a point slightly off from the direct path. And after reaching the highest granularity by zooming in to very specific detail, we might discover that the route is much more difficult than we had presumed. The roads between and leading up to the highways are one-ways, several roads near our destination are closed for construction, and perhaps we will choose to avoid driving through heavily wooded areas at night in fear of encountering wild animals.

The closer we examine the route—or, the higher the scope of granularity—the more precisely we can plan our trip, but it also requires us to take more variables into account. The smaller the piece we examine, the more obstacles present themselves.

It works the same way when planning for a project. When examining the whole, our estimates will be very vague and inaccurate. But by increasing the scope of granularity, we can break down projects into larger coherent sections, referred to as "epics." This will give us a more clear idea of what we should encounter, though it will still not be as clear as if we take the next step to increase the scope of regularity to examining individual story points.

Granularity helps us in two ways. First, it facilitates more involved communication with stakeholders about what they can expect from a given project. It ensures that the team and the stakeholders are on the same page about how each level of the project will be implemented.

Second, it helps us create the most precise estimates possible by breaking the project into individual tasks. It is very difficult to schematize entire projects into numeral estimates; but if we ignore the context and focus on the individual components, assigning numerical values is quite simple. Therefore, a high granularity will produce more familiarity with the nuances of a project, and thus a more accurate estimate.

However, fleshing out this high granularity is an extremely expensive and time-consuming venture. In most cases, the project will change as it is being written, and sometimes the information required will not even be available yet. Because of this, it is best to leave the granularity of scope low until you need it to be more granular. This will save time, money, and frustration if the work ends up being redundant or irrelevant.

If possible, it is best to leave the estimate level at the highest level until you need it to be more granular. Story level estimates should only be a short amount of time ahead of their actual implementation. We will know the most about the details of a particular story only slightly before they need to be realized in an implementation. We also want to push off the cost in time of the most detailed estimates until we are sure the work will need to be completed. These estimates should then become commitments to the cost of completion.

## Completion Confidence

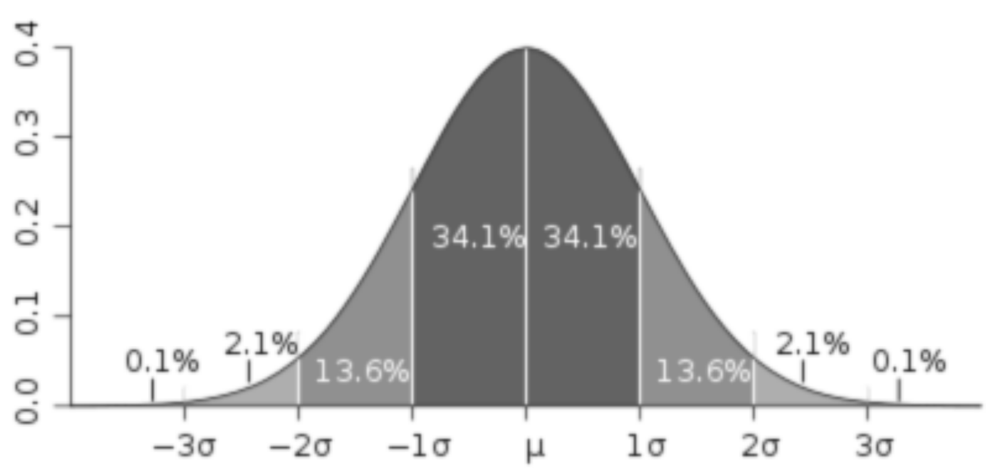
The second major method of risk assessment is to measure completion confidence by taking into account the standard deviation, which is a measure of the variance within your PERT estimates.

We measure Standard Deviation by finding the difference between our optimistic and pessimistic estimates, and then dividing it into sections. The formula looks like this:

STANDARD DEVIATION = (Pessimistic – Optimistic) / 6

By breaking the sample into sections, we can better visualize how our confidence levels wax and wane with our weighted mean. If we place the weighted mean on a bell curve with normal distribution, the standard deviation will let us assign a percentage value to our confidence levels for any estimate.

**Bell Curve w/ Standard Deviations**



Sample data rarely if ever falls along a normal distribution like this. However, standard deviation provides us with the tools necessary to make accurate predictions along more unpredictable and realistic data sets as well. It provides a skeletal structure of partitioning.

Finishing a project ahead of time is not an issue, so we can add everything to the left of the weighted mean to assume there is a 49.9 percent chance the project will be finished on the estimated schedule. However, by simply adding one standard deviation, we can increase that chance up to 84 percent. There are also instances when it is necessary to be more precise, so we would add a second standard deviation to reach a confidence level of 97.6 percent. In software development, that is an extremely high confidence level.

Determining how much confidence to include depends on both the project and the stakeholder with whom you are working. If the project is small (and thus the granularity is high), it is usually fine to have a smaller deviation, because your confidence in the estimate should already be high. However, if the project is large, you will want to include more room to account for unpredictable error.

Likewise, small startup companies learn quickly that they need to be adaptable and flexible to scheduling as they roll out their services to a limited audience. In this instance, one standard deviation will likely suffice—the projects are probably not very involved, and estimate inaccuracy could be simply a matter of hours or days. In these instances, granularity of scope is much more important, because choosing the right features to implement in the earliest stages will be to your advantage in building a minimum viable product.

Large, established companies, though, are capable of doing more market research to understand the product impact before starting development. Because of this, they might require estimates that account for the fixed public release date so a marketing campaign can launch in parallel. Thus, it is extremely important to meet deadlines, and your confidence level will be very important.

The integration of granularity of scope and standard deviation is most important when working on complex software systems for large companies. It will require estimates that can be relied upon to release, but also ones that are vague enough to allow for the inevitable unforeseen problems. Ultimately, this balancing act is what will decide the success or failure of the project.

## Presentation

The two primary factors to take into account when planning your presentation are feedback loop and the firmness of the date.

Feedback loop refers to how frequently you stay in contact with the stakeholder. If there is regular contact, then estimates can be loose. This way, you will be able to solve problems cooperatively and keep the stakeholder updated regularly on the progress of the project. However, if you are unable to hold regular contact, you will want to have a very precise estimate because whatever setbacks you encounter will have to be handled independently and cannot affect the outcome of the project.

In your presentation, it is important to use language that is appropriate for the level of commitment you are making. If your stakeholder is making plans about a big launch to the project and needs a firm commitment, the language should change. For example, you could say, “With this current scope outlined, we can have the first release done in six weeks.” This estimate should have a 95 percent or higher commitment based on the standard deviation math lined out above. If you are offering a range and have a low scope of granularity, then you would use vague language making the uncertainty and risk transparent. For example, you could say, “Our estimates show that understanding a high-level view of the project, it is likely to be completed within 3-5 months." It is extremely important that the stakeholder has a clear idea of what to expect.

As the project progresses, stakeholders should also have a regular feedback mechanism for the updated estimates. If the project is encountering a number of unexpected hurdles in the story points and falling behind schedule, it should be transparent to the stakeholders.

## Conclusion

By nature, estimating is imperfect. It is a creative process that attempts to control for unpredictable factors, surprises, and flaws. Nevertheless, using some of the techniques I have discussed can help us to estimate more confidently. And the more confident we are in our estimates, the better and more transparent the proposal we can offer stakeholders.

In order to estimate our timelines like professionals, it is important for us to be able to understand and communicate what kind of estimation methods and techniques we are employing. That way, we can set expectations truthfully and grant everyone involved with the information they need to make decisions about the project, and their own vision of the future.