

## The Princess and the Pea

**Lawrence H. Putnam and Ware Myers**

Once upon a time there was a fairy-tale princess who was very VERY sensitive. She could feel a pea through her mattress and it kept her awake at night. In the version of the tale that we saw, the princess was played by the comedienne, Carole Burnett, and you can imagine how sensitive she could be! Her attendants piled up the mattresses on the princess's new bed until they were seven feet from the floor. On top of the pile, Carole screamed, "I can still feel the pea!"

That is the mental picture of the software developer that we would like to carry in our head, "I can still feel the *risk!*"

You all know the actual picture. The modern software developer and his cohorts, the stakeholders, are insensitive to risk. They lie directly upon a bed of nails and proclaim proudly, "I feel not so much as a pea!"

Studies show that almost one third of projects are cancelled before completion. At the time of cancellation, they average about 100 percent over schedule and budget. Another third are completed, but at the cost of overrunning schedule and budget by up to one-third. Most of the remaining third exceed the original plan, though by lesser amounts. Only a few projects are completed in less than the originally planned schedule and cost.

Of course, there are numerous reasons for this sorry record, but one reason, surely, is the inability of the participants to feel risk. They need the acute sensitivity of a Carole Burnett! Well, there is only one Carole Burnett and she is busy elsewhere. We, somewhat boldly, step forward. You, too, can learn to be sensitive. There are three steps:

- Feel the *critical* risks (grapefruit size) before you even agree to undertake the project.
- Feel the *significant* risks (lemon size) before you commit to schedule and cost.
- Feel the *ongoing* risks (pea size) all during the project.

If you don't get sensitive in time to feel the risks, don't despair. There are two further steps:

- If you lack time initially to feel the risks, buffer your original bid for risk.
- If you first feel risks late in the project, re-plan your schedule and budget to provide additional time to resolve them.

***Feel the grapefruit!***

When you first encounter that lumpy mattress, do you just look at it from a distance and tell yourself proudly, "It looks like a mattress; it's on a bed frame; I can sleep on anything?"

Or do you actually lie down on it and feel the grapefruit?

What are grapefruit? Sorry, we meant to say "critical risks." They are the risks that, if they materialize, are *critical* to carrying out the project successfully. In other words, if critical risks are not mitigated, the project joins the long list of failed projects. You lose stature in your organization. Your organization loses money.

*Risk* is something that might happen or might not happen. The possibility of your home burning down is a risk. It might burn, or it might not. You can take various steps to minimize the possibility of fire, such as fire-proof shingles and shielded electrical wiring. You can support the local fire department. It often arrives in time to save most of the house.

If your house does burn down, if the risk does occur, the risk turns into a *problem*. Your task then becomes solving the problem. In this case, the solution is building a new house. That takes money. The money can come from fire insurance.

Similarly, a software risk is something that might happen or might not happen. At the very beginning of a software development, when all that exists is a broad vision of what the resulting system might be, all that is feasible is to look for the grapefruit-size risks. Even to do that much, however, you have to know where to look.

- You have to delimit the project. Some activities are within its limits and some are outside. (You don't have to look there.)
- You have to capture the key requirements, or at least consider what they are likely to be.
- You have to sketch out the central features of an architecture that can accommodate these key requirements.

Then, within the scope of these key requirements and the corresponding architectural concept, you have to look for the lumps--the grapefruit-size risks. In general, these big risks fall into two categories: technical and business.

**Technical risks** are those at the edge of the state of the art for your organization. First, of course, you have to identify them (feel the grapefruit) and explore them far enough to understand what you are up against. Then you have a choice of several strategies:

- No bid. Recognize right up front that some jobs are not for you. Trying to sleep on that grapefruit will surely ruin your back!
- Push the critical risks outside the scope of the project, that is, define the project limits to exclude these risks (let somebody else try to do the impossible).

- Consider the probability that a critical risk will actually materialize on this project. If the probability is zero, you can proceed happily.
- If the risk might materialize into an actual problem, can you visualize an approach for coping with it?
- Is there a consultant or subcontractor capable of coping with a risk beyond the capacity of your own organization?

**Business risks** are all the nontechnical happenings that might derail your project--loss of a key person, lack of staff on schedule, and so on. Perhaps the most important business risk at this initial phase is the client organization. Can the key players in your organization work successfully with their counterparts in the client organization? Working through the critical risks provides an opportunity to gauge this risk.

### ***Feel the lemons***

Let's say you see your way past the critical risks. In many applications, of course, there are no critical risks. You have worked in the application area before. You are acquainted with the client organization. You are confident that you feel no grapefruit! Nevertheless, there may still be *significant* risks to take into account.

These are the risks that, if they materialize, may interfere with planning the project. They may throw your estimates of schedule, effort, cost, and defect rate well off the mark. They are risks that you are sure you can surmount, but which you are not yet in a position to estimate the time and effort overcoming them will take.

In other words, a certain amount of exploration of the proposed system should precede the firm bid--the bid in which you agree to schedule, staffing level, cost, and defects at delivery. The Department of Defense has called this phase, high-level design. The Unified Process labels it Elaboration.

In this phase you flesh out the requirements some more. You turn the architectural concept into a baseline architecture. You may carry this architecture all the way to a working prototype of the unfamiliar features. Not only are you then sure your concept will work, but also the client can see that the prototype meets his needs, not just his formally stated requirements.

If the pre-bid phase is carried this far, actual construction of the system should proceed pretty much according to plan, as well as within bid parameters. Well, surprise, surprise! Clients still expect software organizations to bid long before they have enough information (and have reduced the risks to known levels). So, no surprise, clients and software organizations still get into trouble in the construction phase.

### ***Feel the peas***

If, and it's a big IF, the critical and significant risks have been dealt with up front, you will have only pea-size risks to contend with during construction. We're going to assume

that qualified software developers will feel the peas and deal successfully with the pea-size problems that materialize. Moreover, they will do so within the limits of the plan, the schedule, and the budget. Everybody will be happy.

Well, it's a nice dream. In present-day practice, unfortunately, citrus-size risks tend to materialize during construction, because they were not mitigated in the grapefruit and lemon phases. That leads us to two further issues:

- Is there any way to build into the project plan (and its reflection in a bid) an allowance for risks that were not adequately identified or mitigated up front?
- Is there any way to provide additional time and effort in the construction phase after the risks have turned up as serious problems?

In both situations the issues become how to provide the time and effort to resolve the problems to which the materializing risks give rise. (The added issue of actually solving the innumerable problems that arise is beyond the range of this article.)

### ***Buffer your bid***

How do you estimate a project? You probably employ five core metrics:

1. Amount of work to be done, or functionality of the proposed system, probably expressed in some measure of size, such as source lines of code or function points.
2. Productivity, or the rate at which work is accomplished in your organization, probably based on the productivity of the project organization, rather than that of individuals--therefore, *process* productivity.
3. Schedule, that is, the calendar months planned for accomplishment of the project.
4. Effort, that is, staff months planned for the project.
5. Defect rate, or defects remaining at delivery.

First, you have to estimate the amount of work to be done, or the size of the final product. Second, you have to estimate the process productivity you expect to achieve on the project. Fundamentally, there are three approaches to estimating process productivity:

- The method we employ derives process productivity from actual measures of work accomplished on past projects, such as source lines of code divided by effort (person-months) influenced by the time allowed (calendar months).
- The second method arrives at a dozen or more "effort adjustment factors" through management judgment.
- Arm waving--preferably with a cigar at the end of the arm!

However, all of these estimates are just that--estimates. The very name, estimate, implies uncertainty. Their exact value is *uncertain*. A project estimate, therefore, is uncertain:

- The amount of work to be done is uncertain. The size of the eventual product cannot be predicted with certainty. Both are even more uncertain if you have not employed "due diligence," that is, explored the critical and significant risks.
- The productivity level that you will achieve in attacking the work is also uncertain. If you base your process productivity number on managerial judgment, that can easily be inaccurate by 20 percent. If you compute it from the core metrics of completed projects, it will still be uncertain, but by a lesser amount.
- The new project will not be identical to the baseline projects--different staff, managers, and problems. Its process productivity is likely to be different, that is, for the purpose of the next estimate, uncertain.

Yet, the time, effort, and resulting price you have to set forth in a bid is precise. Going from *uncertain* estimates to a *certain* bid is a risk. The question before us is how to buffer that *certain* bid against this risk.

You produce your estimate of time and effort using a formula involving product size and process productivity. Since product size and process productivity are both uncertain, it follows that the time and effort estimates are also uncertain. For instance, if you took your smallest estimate of size and your largest estimate of productivity, you would get your smallest estimates of time and effort. Contrariwise, if you took your largest estimate of size and your smallest estimate of productivity, you would get your largest estimate of time and effort. Thanks to the invention of the computer, you can run this computation a thousand times, taking input estimates of size and productivity from all along their ranges.

The mean of the thousand time-and-effort estimates would then be the most likely outcome. Still, half the estimates would be greater than the mean, and half would be less. In other words, if you were to bid the mean, the probability would be 50 percent that you would complete the project within the bid figures. Over a series of projects, you would break even.

If you did not have time to explore the project for critical and significant risks, you might not feel comfortable with this 50-percent probability. You might want to move out to the larger time and effort figures corresponding to, say, 90 percent probability of success. That would give you a time-and-effort buffer with which to resolve the problems that arise if critical and significant risks materialize.

Of course, we also operate in a competitive system. A competitor might choose to bid at a low level and take the job away from you. He might lose his shirt, too. Just what level at which to enter your bid becomes a matter of business judgment. All this methodology can do is give you a way to buffer the risks you may encounter. Competitors, unfamiliar with risk buffering, may still bid low. That is too bad, but low bids lead to losses, and it is better for you if they have the losses, not you.

Amidst all these uncertainties there is no statistical hocus-pocus that will assure you of making out on each project, even if your clients are willing to accept bids in the 60 or 70

percent probability-of-success range. There will inevitably be risks on some projects that exceed whatever margin of probability you can obtain. Cheer up! On a series of projects the risks should average out; you should make out. Moreover, the series of experiences give you the opportunity to fine-tune your core metrics and your estimating methods. Fine tuning somewhat reduces the inherent uncertainties.

### ***Re-plan the project***

You're in the midst of the construction phase. You're bogged down in some citrus-size problems that you did not originally anticipate, plan for, or budget for. It is becoming evident, at least to you, if not to all your stakeholders, that you are not going to complete the originally specified system on schedule and budget to the intended level of reliability. You are going to have to re-plan, reschedule, and re-budget the rest of the construction phase.

At the time that re-planning the project becomes necessary, you realize that the remaining schedule time and effort will not be sufficient to complete the system as originally specified. It follows then that one or more of the three core metrics on which you based your original bid must be responsible:

- The amount of work to be done is greater than you originally estimated. That is probably the case, because you have now run into citrus-size risks for which you did not originally make allowance.
- Your process productivity is probably lower than you originally planned because you have encountered serious risks (that have now become problems). Solving problems reduces productivity, compared to smooth sailing.
- Your defect rate during construction is probably running higher than you expected.

Of course, you could negotiate with the client to reduce the amount of work and the system reliability (defects remaining) and complete the project within the original schedule and budget. Let us assume, however, that the specifications of the system are firm. Your task is to revise the schedule and effort budget to accomplish those specifications.

You have to re-estimate the first core metric, amount of work, the hard way, that is, think through the additional work resolving the problems will take and express that additional work in the core metric you are using, such as source lines of code.

Your process productivity, however, can be recomputed from the work already accomplished on the project. Since you have accomplished less work in a given amount of time and effort than you have on previous projects, the revised process productivity will be lower.

Using these two revised core metrics, you re-estimate the time and effort to completion. Clients are not pleased, of course, to encounter the need for more time and more money, but you, at least, have some good arguments on your side:

- Some real risks have materialized that haste in getting started prevented making allowance for in the original bid. The time and money now being added should have been there all along.
- The re-planning methodology supports the need for more time and money in an impartial way.

Besides, if a princess can feel a pea through seven mattresses, you should be able to feel a risk when it pricks you good!