

Estimating With Objects - Part XI

Contents

[The data](#)

[Resource estimating accuracy](#)

[Some industrial data](#)

[Further project data](#)

[Interpreting the data](#)

[Some other considerations](#)

[Conclusions](#)

This column is the last in a series about estimating. If you have read the prior 10 columns, you now know how object-oriented methods can help you make good estimates and plans. This column describes some data on how the PROBE method that is described in these articles has helped engineers make better estimates and do better work.

If you are new to this series of columns on Estimating with Objects, the first was in the July 1996 Object Currents issue. The prior columns in this series gave an overview of estimating and defined the steps needed to make size and resource estimates. If you have not read these earlier columns, you should look at them to understand the context for this discussion and to see how these various estimating topics relate. To repeat what I have said in previous columns, the estimating method described here is called PROBE. If you want to learn the PROBE method, you should read my book [A Discipline for Software Engineering](#), from Addison Wesley. This book introduces the Personal Software Process (PSP)SM, which is an orderly and defined way for software engineers to do their work.

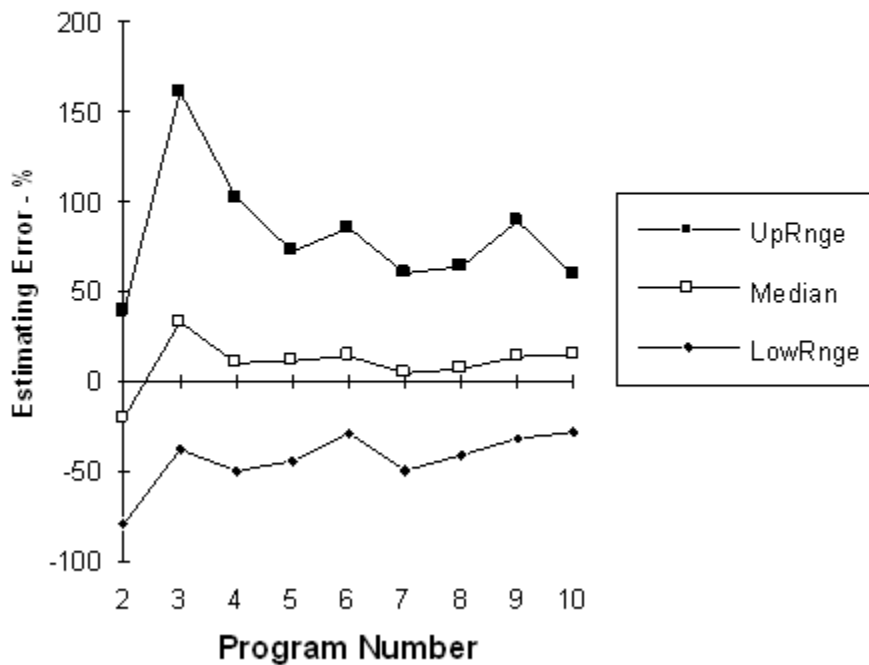
This month's column concludes the discussion of how object-oriented techniques can help you estimate and plan your work. To make a project plan, you need a resource estimate and, to estimate resources, you need to estimate the size of the product you plan to build. Also, to make good estimates, you need historical data on the sizes and development times for the programs you have previously written. The previous columns described how to gather these data and how to use them to make size and resource estimates, judge the accuracy of these estimates, produce a project schedule, and track the work against the plan. This column concludes with data on how these methods work in practice.

The data

The data shown here are two kinds. The first example shows the experiences of students who have learned estimating methods in the PSP course. Following this is some data on engineers who have used the PSP on the job. We have a substantial amount of student data, so I am able to show trends and averages. For working engineers, however, the data are far less. This is both because the PSP is new, and because few industrial software groups have the kind of before-and-after data needed to make comparisons.

The data in Figure 1 show the improvement in size estimating accuracy for 104 engineers from program 2 at the beginning of the PSP course to program 10 at the end. The size estimating data start with program 2 because the students do not make a size estimate for PSP program 1. These data are for both university students and for working engineers who took the PSP course on the job. About two thirds of these 104 engineers either worked in industry when they took the PSP course or had had industrial experience. The results are roughly the same for industrial and university courses.

Figure 1. Size Estimate Error - 104 Engineers



In the PSP course, estimating methods are discussed starting with program 1. The PROBE method is fully introduced with program 4, and the engineers then use it for the balance of the course. Learning to make good estimates has a substantial learning time, and there is a good chance that the learning is not complete by the end of the course. Although we have no data to demonstrate this, engineers will likely continue to improve their estimating ability as they continue to use the PROBE method and to gather and use estimating data.

The estimating data in Figure 1 are plotted with the middle line being the median, or the data for that engineer who was in the middle. There are as many engineers with larger (or more positive estimating errors) as there are engineers with smaller (or more negative estimating errors) than the median.

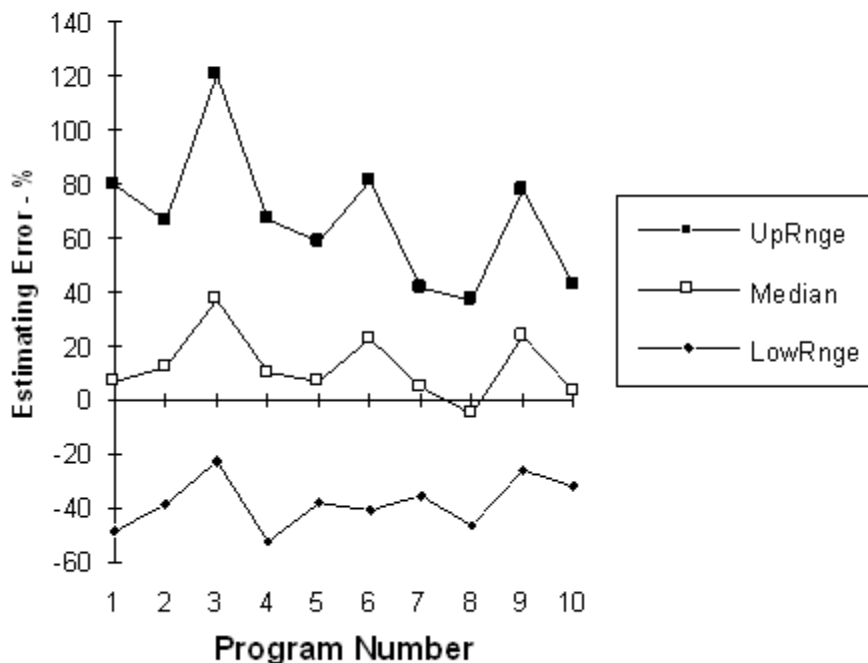
The upper line in the figure is the upper quartile of the range. This is the median or middle of that half of the engineers with the largest or most positive estimating errors. The lower line is the lower quartile, and this is the median of that half of the engineers with the smallest or most negative estimating errors.

In the PSP course, the PROBE method is introduced starting with program 4. As you can see, the size estimating error improves dramatically after program 3. The range of size estimating errors from program 3 to program 10 is reduced by about 50%.

Resource estimating accuracy

Figure 2 shows the resource (or time) estimating errors for the same 104 engineers. Here again, PROBE was introduced starting with program 4. The improvement from program 3 to program 10 of the PSP programming exercises is again about a factor of 2.

Figure 2. Time Estimating Error - 104 Engineers



The general PSP experience is that time estimating accuracy improves more than size estimating. The opinion of the engineers is that this is because they have more control over the amount of time they spend than over the size of the program they write. The simple fact of making an

estimate and tracking the work against the estimated seems to motivate engineers to meet their plan.

Some industrial data

The first industrial case is for a four-engineer team who started a project before they were PSP trained. At the beginning, they estimated how long the work would take. They seriously underestimated and were several months late in developing the first 3 components. Their average error in estimating the number of weeks to do the work was 394%.

Management then decided to train the engineers in the PSP and to renegotiate the contract with the customer. When the engineers estimated the next six components, their average error was -10.4%. That is, on average, they finished 10.4% early.

These results are shown in Figure 3 and Figure 4. In Figure 3, the number of estimated and actual weeks are shown by component. Figure 4 shows the percentage error for the estimates for these same components.

Figure 3. Estimate vs. Actual Weeks

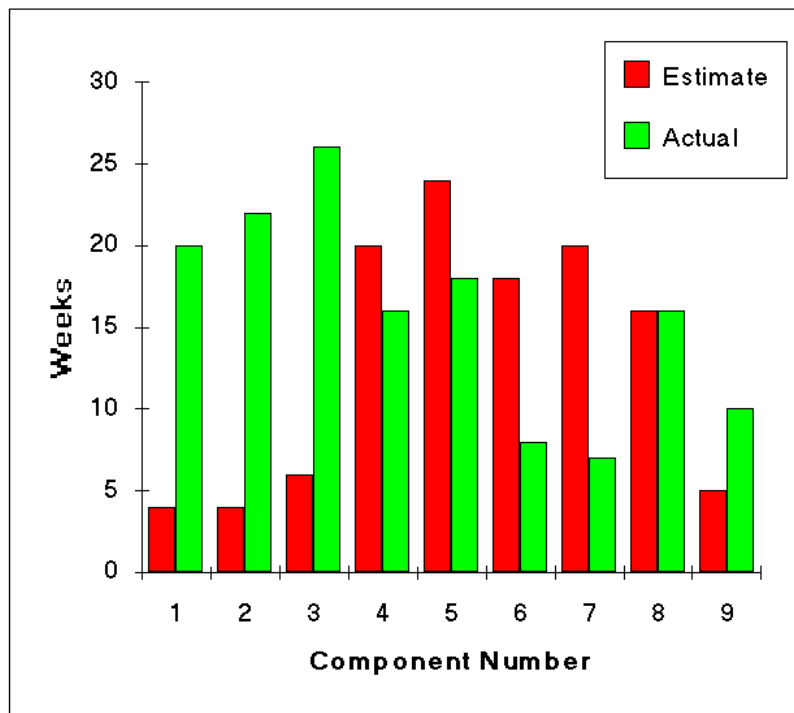
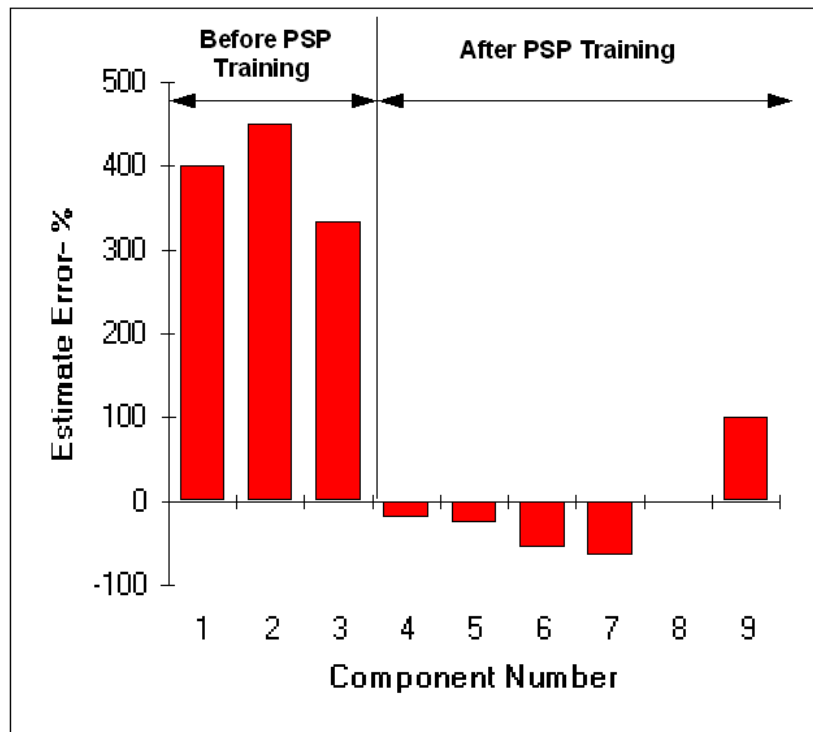


Figure 4. Schedule Estimating Error



Further project data

Table 1 shows data on a group of six industrial projects. Here, the first two projects were completed with engineers who had not had PSP training. As you can see, both projects were late, and they both took nearly twice as long to develop as planned. The next four projects were estimated and developed by PSP-trained engineers. Here, you can see that three of the projects were on schedule and one was delivered 2 months early.

Table 1. Summary of Project Data

	PSP	Size	Estimate	Actual	Late %
	No	19 Req.	2 months	5 months	+ 125%
	No	30 Req.	10 months	19 months	+ 90%
	Yes	24 Req.	7 months	5 months	- 29%

	Yes	2255 LOC	6 months	6 months	0 %
	Yes	1400 LOC	2 months	2 months	0 %
	Yes	6196 LOC	2 months	2 months	0 %

Interpreting the data

These results look so good, some people find them hard to believe. Part of the reason for this reaction is that the industrial data look so much better than the student data. This, however, is not magic. In fact the results are quite logical when you consider the two cases more carefully. In the PSP course, the improvement is from the first estimates before the students had personal data, before they used the PROBE method, and before they had much estimating experience. They were, however, making estimates of every program, even the first.

In the industrial cases, the improvement is from first cases which were not estimates. You might even call them guesses. The second cases, however, are estimates made by the engineers who did the work. These engineers had all been PSP trained. They had used size estimating for 9 previous programs and also estimated development times for 10 programs. They also had their PSP data to help them make the estimates. Thus, for the initial industrial case, the comparison is with engineers who did not know how to estimate. So the industrial improvement is from not estimating to estimating. In the PSP course, we see the improvement in estimating ability from a first estimate to a last one.

Some other considerations

Another fact that shows up with the PSP is that the number of defects engineers inject in their products impacts their estimating accuracy. It takes a lot of compile and test time to find and fix large numbers of defects. Since this compile and test time is highly variable, when engineers have many defects, they have a large amount of unpredictable development time. This results in high estimating errors. Since the PSP course also reduces compile and test defects by 5 to 10 times, this also helps improve estimating accuracy.

A further factor is the use of the proper data. Early in the PSP course, engineers do not have personal data, so they use data from the textbook. While this is better than nothing, when they have personal size and development time data, and when they adjust their prior estimates to see what their estimating errors would have been had they used personal data, the errors are much less.

There is so much variation among individual engineers that, when they use their personal data, they can make much better estimates. They also build an increasing store of data to help make better estimates in the future. The PSP shows engineers how to do this.

Conclusions

A growing number of engineers are now using the PSP and their experiences are roughly similar. With the prior projects, they were generally late, but after PSP training, they usually meet their schedules. Note, however, that few organizations can make publishable comparisons of before and after cases. The reason is that without PSP-trained engineers, few organizations have data on estimating performance. In addition to better planning and estimating, the PSP also helps engineers do better work. PSP-trained engineers routinely produce defect-free programs and their average productivity improvement is better than 20%. But that is a different story.

This concludes the series on Estimating With Objects. If you are interested in finding out more about this subject, I suggest you read my prior columns in this series, or get a copy of my book: [A Discipline for Software Engineering](#), from Addison Wesley.

This is my final column in *Object Currents*. Thanks for your attention and happy estimating.

Watts S. Humphrey
watts@sei.cmu.edu