31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 85

6

The PROBE Estimating Method

The principal objective of the estimating process is to produce accurate estimates. To do this, you must start with the best requirements you can get, obtain good historical data, and use a sound estimating method. Because your early estimates will likely have large errors, you also need a way to judge and improve your estimating accuracy. The PROBE method shows you how to obtain estimating data, how to use these data to make estimates, and how to measure and improve the accuracy of your estimates.

6.1 Estimating from Data

The PROBE method guides you in using historical data to make estimates. For example, in estimating the work to develop a database query system, you would first produce the conceptual design and then divide it into parts. Then you would estimate the number of elements in each part. For example, if you estimate a total of 80 elements and you know that each element has taken you, on average, 1.5 hours to develop, the total development-time estimate would be 120 hours, or 80 (the number of elements) times 1.5 (the time per element).

To make accurate estimates, however, you need a statistically sound way to determine the average time required to develop a part. The PROBE method shows you how to do this. Figure 6.1 shows data on 21 database projects. These are the same data shown in Figure 3.1 (see p. 39). The trend line for these data is calculated with a method called **linear regression**. It produces a line that accurately fits the data. The **trend line**, or **regression line**, is represented by the following equation:

$$\text{Development Time} = \beta_0 + \beta_1 * \text{EstimatedSize}$$
$$= 0.17 + 1.53 * 80 = 122.57$$

Here, $\beta_0 = 0.17$ and $\beta_1 = 1.53$. These β values are calculated by the PROBE method described in this chapter and they are determined mathematically to best represent the trend of the data. The β_1 value is the average development time required per database element, or about 1.5 hours per element, and β_0 is the overhead time. In other words, this equation indicates that, for a project with no elements, the development time would be 0.17 hour, or about 10 minutes. For larger projects, you would then add 1.5 hours per element.

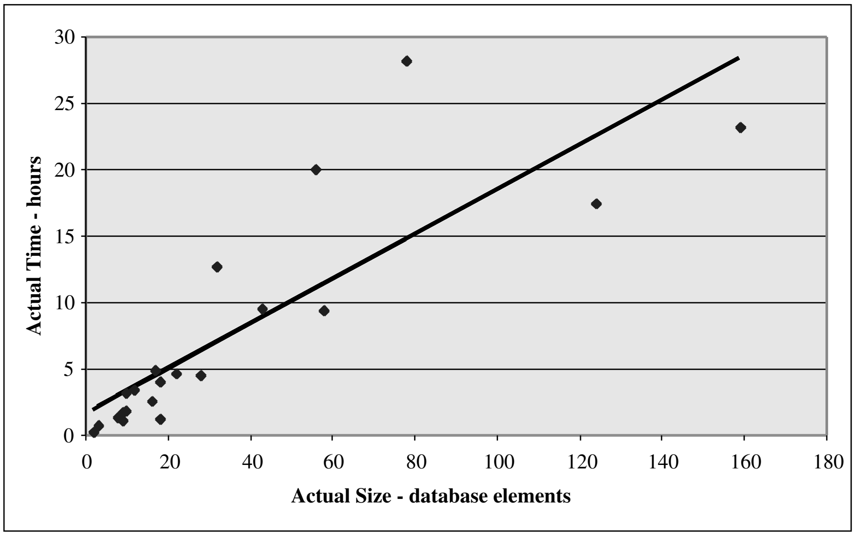

FIGURE 6.1 EXAMPLE DATABASE ESTIMATING DATA

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 87

6.2 Proxy-Based Estimating

The *PRO*xy-*B*ased *E*stimating (PROBE) method enables you to use any item you choose as the proxy, as long as it meets the criteria for a good proxy that are described in Chapter 5 (see pp. 71–74). Using one developer's data, the basic steps of the size estimating procedure are shown in Figure 6.2 and described in the following paragraphs. Because, at least initially, you will not have enough data to use the full PROBE method, several exception cases are described later in the chapter.

PROBE Step 1: The Conceptual Design

After you complete the conceptual design, you identify the parts of the product you plan to develop. You are now ready to make the size estimate.

PROBE Step 2: Estimate Parts Size

Following the methods described in Chapter 5, estimate the sizes of the conceptual design's parts. Do this by determining the part types and then judging the number of items (methods) for each part and the relative sizes of the items. Once you know whether an item is very small (VS), small (S), medium (M), large (L), or very large (VL), look up the item size in a table like Table 5.1 (see p. 77).

For example, the size estimate made by Student 12 for Program 8 is shown in Table 6.1. She first named each added part in the conceptual design and determined its type. The first part, Matrix, was of the type *Data.* She next estimated that this part would likely contain 13 items and judged their relative size to be medium. Finally, she determined from Table 5.1 that a medium Data part would have 8.84 LOC per item. Multiplying this by the number of items resulted in 114.9 LOC. She then repeated this procedure for each added part to arrive at a total of 361 added-part LOC. After completing the project, she entered the actual sizes for these same parts.

If a part has items of different types, estimate each part-item combination as a separate part. For example, if class A has items (methods) of type Char, Calc, and Print, treat these as three parts: AChar, ACalc, and APrint.

PROBE Step 3: Estimate the Sizes of the Reused Parts and Base Additions

In PROBE step 3, estimate the size of any other program parts (base, deleted, reused, and so on). For example, if you can find available parts that provide the functions required by your conceptual design, you may be able to reuse them. Provided that

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 88

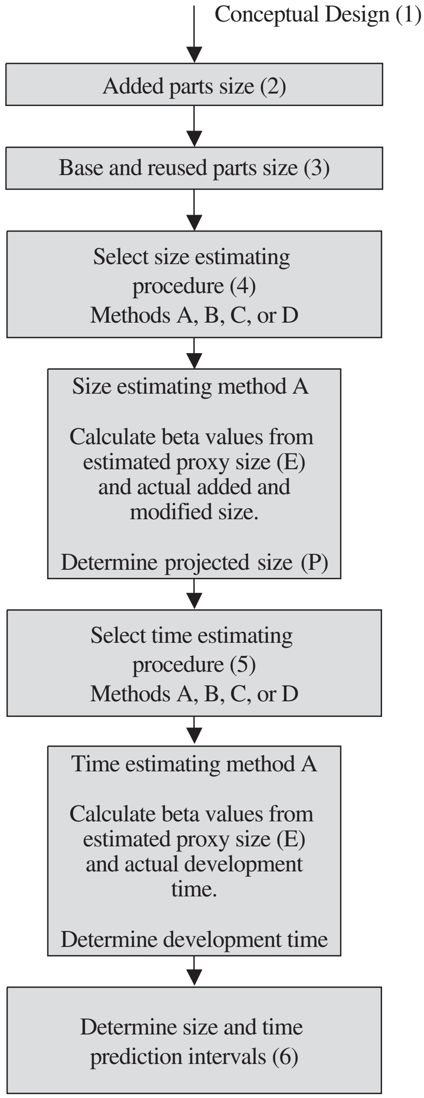

FIGURE 6.2 THE PROBE ESTIMATING METHOD

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 89

TABLE 6.1 THE PARTS ADDITION ESTIMATE

	Estimated				Actual	
Parts Additions	**Type**	**Items**	**Rel. Size**	**Size***	**Size***	**Items**
Matrix	Data	13	M	115	136	14
LinearSystem	Calc.	8	L	197	226	11
LinkedList	Data	3	L	49*	54*	3
Control	Logic				48	2
Total			**PA**	361	464	

these parts work as intended and are of suitable quality, reusing available parts can save considerable time. For any parts from the reuse library, note their names and sizes under *Reused Parts.* In the example shown in Table 6.2, Student 12 identified two reused parts with a total of 169 LOC.

PROBE considers two kinds of reused parts. First are the parts taken from a reuse library. The second, the *new-reusable* parts, are some of the added parts you have already estimated. Identify these as new-reusable parts if you plan to use them in developing other programs. It is good practice to build a library of useful parts when working in a new application environment. Then, when developing new programs, code you plan to add to the reuse library is new-reusable code. In Table 6.1, Student 12 planned to add three items to the LinkedList part, or 49 LOC of estimated new-reusable code. She noted these items with an asterisk beside their LOC estimates. Because she took the other LinkedList items from the reuse library, she included them as *Reused Parts.*

If you plan to modify an existing program, determine the size of the base program to be enhanced and the sizes of any changes you expect to make. Table 6.3

TABLE 6.2 REUSED PARTS ESTIMATE

Reused Parts		**Estimated Size**	**Actual Size**
LinkedList		73	73
DataEntry		96	96
Total	**R**	169	169

TABLE 6.3 SIZE ESTIMATING TEMPLATE

Student	Student 12	Date	5/1
Program	Multiple Regression	Program #	8
Instructor	Humphrey	Language	C++
Size Measure	LOC		

	Estimated			
Base Parts	**Base**	**Deleted**	**Modified**	**Added**
PSP Program 3	695	0	5	0
Total	**B** 695	**D** 0	**M** 5	**BA** 0

	Actual			
Base Parts	**Base**	**Deleted**	**Modified**	**Added**
PSP Program 3	695	0	18	0
Total	695	0	18	0

	Estimated				Actual	
Parts Additions	**Type**	**Items**	**Rel. Size**	**Size***	**Size***	**Items**
Matrix	Data	13	M	115	136	14
LinearSystem	Calc.	8	L	197	226	11
LinkedList	Data	3	L	49*	54*	3
Control	Logic				48	2
Total			**PA**	361	464	

TABLE 6.3 (continued)

Student Student 12 Program 8

Reused Parts		**Estimated Size**	**Actual Size**
LinkedList		73	73
DataEntry		96	96
Total	**R**	169	169

PROBE Calculation Worksheet (Added and Modified)		**Size**	**Time**
Added size (A):	A = BA + PA	361	
Estimated proxy size (E):	E = BA + PA + M	366	
PROBE estimating basis used:	(A, B, C, or D)	A	A
Correlation:	(R^2)	0.62	0.58
Regression Parameters:	β_0 Size and Time	62	108
Regression Parameters:	β_1 Size and Time	1.3	2.95
Projected Added and Modified Size (P):	$P = \beta_{0size} + \beta_{1size}$*E	538	
Estimated Total Size (T):	T = P + B – D – M + R	1397	
Estimated Total New Reusable (NR):	sum of * items	49	
Estimated Total Development Time:	Time = $\beta_{0time} + \beta_{1time}$ * E		1186
Prediction Range:	Range	235	431
Upper Prediction Interval:	UPI = P + Range	773	1617
Lower Prediction Interval:	LPI = P – Range	303	755
Prediction Interval Percent:		90	90

shows the completed PSP Size Estimating Template with the added and reused parts, the base parts, and the planned modifications. Student 12 identified 695 LOC of base code, 5 of which she planned to modify. She actually did modify 18 LOC.

Although developers occasionally get to develop entirely new programs, most of our work is enhancing existing programs. Here, base program size is the total size of the unmodified program before development. Base additions are for enhancements you make to the base program. If you know enough about the base product to do so, estimate base additions the same way you estimate added parts.

The reused category is only for unmodified parts. When modifying existing programs, the unmodified program is the base, and you estimate its additions, changes, and deletions. Even if the base program is unmodified, it is not considered reused unless it was specifically intended for reuse. The reused category is only for parts that come directly from the reuse library without *any* modification.

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 92

The final part of step 3 is to add all of the parts estimates to obtain the estimated proxy size (E), to be entered in the Size Estimating Template in Table 6.3. E is the number used by PROBE to make the size and time projections. In the example, Student 12 had a total of 361 LOC of parts additions, 5 LOC of modifications, and 0 LOC of base additions, resulting in a value of 361 + 5 + 0 = 366 LOC for E. This number is entered in the Estimated proxy size (E) space on line 2 of the PROBE Calculation Worksheet.

PROBE Step 4: The Size Estimating Procedure

In PROBE step 4, check your data to determine whether you can use PROBE method A. After you have used the PSP for a while, you will normally use method A to make size and time estimates. Later in the chapter, after examining the basic PROBE method A, you will learn how to use PROBE methods B, C, and D to make estimates with limited data.

With estimated proxy size E, you can calculate the projected program size P and total estimated development time. If, for example, historical data showed that your finished programs were generally about 25% larger than estimated, you might add a 25% fudge factor to each estimate. The PROBE method essentially does this but in a statistically sound way. As noted before, the method is called linear regression. Although these calculations are a little more complex than taking simple averages, they use historical data to produce a statistically sound estimate. The parameters β_0 and β_1 are used in the following equation to calculate projected added and modified size:

$$\text{Projected Added and Modified Size (P)} = \beta_0 + \beta_1 * \text{E}$$

When two sets of data are strongly related, you can use the linear regression method to represent that relationship. As Figure 5.2 (see p. 73) and Figure 5.3 (see p. 75) show, estimated part size and actual added and modified size are often closely correlated. This means that linear regression is often appropriate. The parameters β_0 and β_1 are calculated from your historical data. These calculations are described later with an example.

Finished programs usually contain more than just the parts specified in the conceptual design. For example, they will likely contain declaration and header code that is not included in the parts estimates. To account for this and for any other such code, you must use a factor that is based on your historical experience. Fortunately, the PROBE method accounts for this factor when it calculates the β_0 and β_1 parameters.

In the example shown in Table 6.3, the fact that β_1 = 1.3 indicates that Student 12's finished programs have historically been 30% bigger than the total of the estimated parts and modifications. In addition, the value $\beta_0 = 62$ indicates that, on average, she underestimated by 62 LOC. As shown in Table 6.3, the regression calculations result in a total of 538 projected added and modified LOC. In this case, because added and modified code were used in making the size estimate, total program size is calculated by adding the 695 LOC of base code and the 169 LOC of reused code, and subtracting the 5 LOC of modified code. The total estimate is then 1,397 LOC for the finished program. The modified LOC are subtracted because they would otherwise be counted twice: once in the 695 LOC of base code and again in the estimated proxy size (E). As noted in size accounting in Chapter 3 (see p. 42), a line of modified code is treated as both an added line and a deleted line.

PROBE Step 5: The Time Estimating Procedure

In PROBE step 5, again check your data to determine whether you can use PROBE method A for the time estimates. Although method A is also the preferred method for time estimates, you can also use alternative methods B, C, and D, which are described later in the chapter. Once you have obtained the β values for time, use them and the estimated proxy size (E) to calculate the estimated development time. Then enter the size and time estimates in the plan spaces in the Project Plan Summary shown in Table 6.4.

PROBE Step 6: The Prediction Interval

The final calculations on the PROBE Size Estimating Template are for the **prediction interval**. The prediction interval is a statistically determined range around your size or time estimate within which the actual value is likely to fall. For a 70% prediction interval, you would expect the actual time and size values to fall outside of this range about 30% of the time. The prediction interval is described in Chapter 7.

TABLE 6.4 PSP1 PROJECT PLAN SUMMARY

Student	Student 12	Date	5/1
Program	Multiple Regression	Program #	8
Instructor	Humphrey	Language	C++

Summary	***Plan***	***Actual***	***To Date***
Size/Hour	27.4	30.9	23.4

Program Size	***Plan***	**Actual**	**To Date**
Base (B)	695	695	
	(Measured)	(Measured)	
Deleted (D)	0	0	
	(Estimated)	(Counted)	
Modified (M)	5	18	
	(Estimated)	(Counted)	
Added (A)	533	464	
	(A+M – M)	(T – B + D – R)	
Reused (R)	169	169	
	(Estimated)	(Counted)	
Added and Modified (A+M)	538	482	2081
	(Projected)	(A + M)	
Total Size (T)	1397	1328	5161
	(A+M + B – M – D + R)	(Measured)	
Total New Reusable	49	54	294
Estimated Proxy Size (E)	366		

Time in Phase (min.)	**Plan**	**Actual**	**To Date**	**To Date %**
Planning	146	166	710	13.3
Design	425	332	1220	22.8
Code	390	333	1639	30.6
Compile	30	16	288	5.4
Test	105	31	749	14.1
Postmortem	90	58	737	13.8
Total	1186	936	5343	100

Defects Injected	**Actual**	**To Date**	**To Date %**
Planning	2	6	3.5
Design	11	38	22.0
Code	26	129	74.5
Compile	0	0	0
Test	0	0	0
Total Development	39	173	100

Defects Removed	**Actual**	**To Date**	**To Date %**
Planning	0	0	0
Design	10	10	5.8
Code	19	72	41.6
Compile	10	61	35.3
Test	0	30	17.3
Total Development	39	173	100
After Development			

6.3 Estimating with Limited Data

Table 6.5 shows the four PROBE methods, the conditions for using them, and how they are used. These methods are also described in the PROBE Estimating Script in Table 6.6, the Size Estimating Template Instructions in Table 6.7, and the PROBE Calculation Worksheet Instructions in Table 6.8.

By choosing one of the four PROBE size estimating procedures, you decide how to calculate the β parameters. Base this choice on the quality of your data. Method A should be your first choice but it requires at least three, and preferably four, data points of *estimated proxy size* (E) and *actual added and modified size* that correlate with an $r >= 0.7$. If you can't use method A, try to use method B. This method uses *plan added and modified size* and *actual added and modified size.* Again, you must have at least three, and preferably four, data points that correlate with an $r >= 0.7$. If the data are not adequate for methods A and B, then use method C if you have at least some data on plan and actual added and modified size. If you don't have any data, you must use method D. With method D, you are not actually making a projection but merely guessing at a value to enter as

TABLE 6.5 THE FOUR ALTERNATE PROBE CALCULATION METHODS

Method	Data Used for Beta Values		Data Requirements
	Size	**Time**	
A	Estimated proxy size and actual program size	Estimated proxy size and actual development time	The data must correlate with $r >= 0.7$
B	Planned program size and actual program size	Planned program size and actual development time	The data must correlate with $r >= 0.7$
C	Planned program size if available and actual program size. Set β_0 = 0 and β_1 = to-date actual size/to-date planned size. If planned data not available, set β_1 = 1.0	Planned program size if available and actual development time. If planned size data not available, use actual size. Set β_0 = 0 and β_1 = to-date actual time/to-date planned size. If planned data not available, set β_1 = to-date actual time/to-date actual size.	Some actual size and time data
D			No data

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 96

TABLE 6.6 PROBE ESTIMATING SCRIPT

Purpose		To guide the size and time estimating process using the PROBE method
Entry Criteria		• Requirements statement • Size Estimating template and instructions • Size per item data for part types • Time Recording log • Historical size and time data
General		• This script assumes that you are using added and modified size data as the size-accounting types for making size and time estimates. • If you choose some other size-accounting types, replace every “added and modified” in this script with the size-accounting types of your choice.
Step	**Activities**	**Description**
1	Conceptual Design	Review the requirements and produce a conceptual design.
2	Parts Additions	Follow the Size Estimating template instructions to estimate the parts additions and the new reusable parts sizes.
3	Base Parts and Reused Parts	• For the base program, estimate the size of the base, deleted, modified, and added code. • Measure and/or estimate the side of the parts to be reused.
4	Size Estimating Procedure	• If you have sufficient estimated proxy size and actual added and modified size data (three or more points that correlate), use procedure 4A. • If you do not have sufficient estimated data but have sufficient plan added and modified and actual added and modified size data (three or more points that correlate), use procedure 4B. • If you have insufficient data or they do not correlate, use procedure 4C. • If you have no historical data, use procedure 4D.
4A	Size Estimating Procedure 4A	• Using the linear-regression method, calculate the β_0 and β_1 parameters from the estimated proxy size and actual added and modified size data. • If the absolute value of β_0 is not near 0 (less than about 25% of the expected size of the new program), or β_1 is not near 1.0 (between about 0.5 and 2.0), use procedure 4B.
4B	Size Estimating Procedure 4B	• Using the linear-regression method, calculate the β_0 and β_1 parameters from the plan added and modified size and actual added and modified size data. • If the absolute value of β_0 is not near 0 (less than about 25% of the expected size of the new program), or β_1 is not near 1.0 (between about 0.5 and 2.0), use procedure 4C.
4C	Size Estimating Procedure 4C	If you have any data on plan added and modified size and actual added and modified size, set β_0 = 0 and β_1 = (actual total added and modified size to date/plan total added and modified size to date).
4D	Size Estimating Procedure 4D	If you have no historical data, use your judgment to estimate added and modified size.

TABLE 6.6 (continued)

Step	Activities	Description
5	Time Estimating Procedure	• If you have sufficient estimated proxy size and actual development time data (three or more points that correlate), use procedure 5A. • If you do not have sufficient estimated size data but have sufficient plan added and modified size and actual development time data (three or more points that correlate), use procedure 5B. • If you have insufficient data or they do not correlate, use procedure 5C. • If you have no historical data, use procedure 5D.
5A	Time Estimating Procedure 5A	• Using the linear-regression method, calculate the β_0 and β_1 parameters from the estimated proxy size and actual total development time data. • If β_0 is not near 0 (substantially smaller than the expected development time for the new program), or β_1 is not within 50% of 1/(historical productivity), use procedure 5B.
5B	Time Estimating Procedure 5B	• Using the linear-regression method, calculate the β_0 and β_1 regression parameters from the plan added and modified size and actual total development time data. • If β_0 is not near 0 (substantially smaller than the expected development time for the new program), or β_1 is not within 50% of 1/(historical productivity), use procedure 5C.
5C	Time Estimating Procedure 5C	• If you have data on estimated—added and modified size and actual development time, set β_0 = 0 and β_1 = (actual total development time to date/estimated – total added and modified size to date). • If you have data on plan – added and modified size and actual development time, set β_0 = 0 and β_1 = (actual total development time to date/plan total added and modified size to date). • If you only have actual time and size data, set β_0 = 0 and β_1 = (actual total development time to date/actual total added and modified size to date).
5D	Time Estimating Procedure 5D	If you have no historical data, use your judgment to estimate the development time from the estimated added and modified size.
6	Time and Size Prediction Intervals	• If you used regression method A or B, calculate the 70% prediction intervals for the time and size estimates. • If you did not use the regression method or do not know how to calculate the prediction interval, calculate the minimum and maximum development time estimate limits from your historical maximum and minimum productivity for the programs written to date.
Exit Criteria		• Completed estimated and actual entries for all pertinent size categories • Completed PROBE Calculation Worksheet with size and time entries • Plan and actual values entered on the Project Plan Summary

the plan added and modified size or the development time on the Project Plan Summary.

After selecting an estimating method, calculate the β_0 and β_1 values and verify that they are reasonable. If β_0 for size is larger than about 25% of the planned program's expected size, or β_1 is not between about 0.5 and 2.0, use method B; if

TABLE 6.7 SIZE ESTIMATING TEMPLATE INSTRUCTIONS

Purpose	Use this form with the PROBE method to make size estimates.
General	• A part could be a module, component, product, or system. • Where parts have a substructure of methods, procedures, functions, or similar elements, these lowest-level elements are called items. • Size values are assumed to be in the unit specified in size measure. • Avoid confusing base size with reuse size. • Reuse parts must be used without modification. • Use base size if additions, modifications, or deletions are planned. • If a part is estimated but not produced, enter its actual values as zero. • If a part is produced that was not estimated, enter it using zero for its planned values.
Header	• Enter your name and the date. • Enter the program name and number. • Enter the instructor's name and the programming language you are using. • Enter the size measure you are using.
Base Parts	If this is a modification or enhancement of an existing product • measure and enter the base size (more than one product may be entered as base) • estimate and enter the size of the deleted, modified, and added size to the base program After development, measure and enter the actual size of the base program and any deletions, modifications, or additions.
Parts Additions	If you plan to add newly developed parts • enter the part name, type, number of items (or methods), and relative size • for each part, get the size per item from the appropriate relative size table, multiply this value by the number of items, and enter in estimated size • put an asterisk next to the estimated size of any new-reusable additions After development, measure and enter • the actual size of each new part or new part items • the number of items for each new part
Reused Parts	If you plan to include reused parts, enter the • name of each unmodified reused part • size of each unmodified reused part After development, enter the actual size of each unmodified reused part.

the β values are still not within the desired ranges, use method C. For method C, with some size data, total the plan added and modified sizes for the programs for which you have data and total the actual added and modified sizes for the same programs. Then set β_1 equal to to-date total actual size divided by the to-date plan size and set β_0 to 0. If you don't have planned size data, set $\beta_1 = 1.0$. For method D, make your best guess.

For the time estimate, the guidelines are much the same. For example, if the time β_0 is larger than about 25% of the planned program's expected development

TABLE 6.8 PROBE CALCULATION WORKSHEET INSTRUCTIONS

Purpose	Use this form with the PROBE method to make size and resource estimate calculations.
General	The PROBE method can be used for many kinds of estimates. Where development time correlates with added and modified size • use the Added and Modified Calculation Worksheet • enter the resulting estimates in the Project Plan Summary • enter the projected added and modified value (P) in the added and modified plan space in the Project Plan Summary If development time correlates with some other combination of size-accounting types • define and use a new PROBE Calculation Worksheet • enter the resulting estimates in the Project Plan Summary • use the selected combination of size accounting types to calculate the projected size value (P) • enter this P value in the Project Plan Summary for the appropriate plan size for the size-accounting types being used
PROBE Calculations: Size (Added and Modified)	• Added Size (A): Total the added base code (BA) and Parts Additions (PA) to get Added Size (A). • Estimated Proxy Size (E): Total the added (A) and modified (M) sizes and enter as (E). • PROBE Estimating Basis Used: Analyze the available historical data and select the appropriate PROBE estimating basis (A, B, C, or D). • Correlation: If PROBE estimating basis A or B is selected, enter the correlation value (R^2) for both size and time. • Regression Parameters: Follow the procedure in the PROBE script to calculate the size and time regression parameters (β_0 and β_1), and enter them in the indicated fields. • Projected Added and Modified Size (P): Using the size regression parameters and estimated proxy size (E), calculate the projected added and modified size (P) as $P = \beta_{0Size} + \beta_{1Size} * E$. • Estimated Total Size (T): Calculate the estimated total size as T = P+B-D-M+R. • Estimated Total New Reusable (NR): Total and enter the new reusable items marked with *.
PROBE Calculations: Time (Added and Modified)	• PROBE Estimating Basis Used: Analyze the available historical data and select the appropriate PROBE estimating basis (A, B, C, or D). • Estimated Total Development Time: Using the time regression parameters and estimated proxy size (E), calculate the estimated development time as $Time = \beta_{0Time} + \beta_{1Time} * E$.
PROBE Calculations: Prediction Range	• Calculate and enter the prediction range for both the size and time estimates. • Calculate the upper (UPI) and lower (LPI) prediction intervals for both the size and time estimates. • Prediction Interval Percent: List the probability percent used to calculate the prediction intervals (70% or 90%).
After Development (Added and Modified)	Enter the actual sizes for base (B), deleted (D), modified (M), and added base code (BA), parts additions (PA), and reused parts (R)

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 100

time, or β_1 is not between about 0.5 and 2.0 times your historical hours/LOC, use method B; and if the β values are still not within the desired ranges, use method C. For method C, assuming that you have at least some size data, total the plan added and modified sizes for the programs for which you have data and total the actual development times for the same programs. Then set β_1 equal to the total actual development times divided by the total planned sizes and set β_0 to 0. For method D, make your best guess.

The final step in the PROBE method is to calculate the value of P, the projected program size, and the estimated development time and enter them on the Project Plan Summary. Enter the projected size for Added and Modified Size (A+M) in the *Plan* column; and the development time under Time in Phase, Total, in the *Plan* column.

For many types of software work, good correlations are likely for development time and added and modified size. If this is the case for your data, use the Calculation Worksheet in the PROBE Size Estimating Template (see Table 6.3, p. 91) to produce the projected value, P, as the Projected Added and Modified Size for the program. Then enter this value in the *Plan* column of the Project Plan Summary. However, if the best correlation was for some other combination of the size accounting types, you could use that. Then you would have to produce an appropriately modified PROBE Calculation Worksheet to calculate projected program size for the size accounting types you plan to use.

6.4 An Estimating Example

The following example shows the time-estimation calculations for the data in Table 6.9.

TABLE 6.9 EXAMPLE HISTORICAL DATA

Program Number	Estimated Proxy Size	Plan Added and Modified Size	Actual Total Hours
1		83	11.2
2		116	9.3
3		186	21.6
4	97	81	6.9
5	81	114	10.2
Totals	178	580	59.2

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 101

1. You have the estimated proxy size of E = 126 LOC.
2. As shown in Table 6.9, you have estimated proxy size data for only two programs, so you cannot use method A. Next, check the correlation between *Plan Added and Modified Size* and *Actual Total Hours* to determine whether you can use method B. Using the data values in Table 6.10 and the formulas shown in Box 3.1 (see p. 36), calculate the value of the correlation r as follows:

$$r = \frac{n * Sum(x * y) - Sum(x) * Sum(y)}{\sqrt{\left(n * Sum(x^2) - Sum(x)^2\right) * \left(n * Sum(y^2) - Sum(y)^2\right)}} =$$

$$r = \frac{5 * 7747.7 - 580 * 59.2}{\sqrt{(5 * 74498 - 580 * 580) * (5 * 830.14 - 59.2 * 59.2)}} = \frac{4402.5}{4828.7} = 0.9117$$

Because $r >= 0.7$, you can use method B.

Even though most PSP and TSP support tools make these calculations, you should understand how they are made. Box 6.1 gives the calculation method, and one of the PSP exercises is a program to calculate the regression parameters. The assignment kit for that program describes the calculations with an example.

3. Using the values in Table 6.10 with the formulas in Box 6.1, the regression parameters are as follows:

$$\beta_1 = \frac{Sum(x * y) - n * x_{avg} * y_{avg}}{Sum(x^2) - n * (x_{avg})^2} = \frac{7747.7 - 5 * 116 * 11.84}{74498 - 5 * 116^2} = 0.121987$$

$$\beta_0 = y_{avg} - \beta_1 * x_{avg} = 11.84 - 0.121987 * 116 = -2.31046$$

TABLE 6.10 EXAMPLE REGRESSION CALCULATIONS

Program Number	Estimated Proxy Size – *x*	Actual Hours – *y*	Size*Size *x*x*	Size*Hours *x*y*	Hours*Hours *y*y*
1	83	11.2	6,889	929.6	125.44
2	116	9.3	13,456	1,078.8	86.49
3	186	21.6	34,596	4,017.6	466.56
4	81	6.9	6,561	558.9	47.61
5	114	10.2	12,498	1,162.8	104.04
Total	580	59.2	74,498	7,747.7	830.14
Average	116	11.84			

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 102

Using these β values, calculate the estimated development time as follows:

Time = $\beta_0 + \beta_1$ * E = –2.31046 + 126 * 0.121987 = 13.060 hours

BOX 6.1 CALCULATING THE SIZE OR TIME PARAMETERS β_0 AND β_1

Start with *n* values of *x* and *y*, where *x* is the estimated proxy size (E) and *y* is the actual added and modified size or actual development time for the *n* programs:

$$\beta_1 = \frac{\sum_{i=1}^{n} x_i y_i - n x_{avg} y_{avg}}{\sum_{i=1}^{n} x_i^2 - n x_{avg}^2}$$

$$\beta_0 = y_{avg} - \beta_1 x_{avg}$$

6.5 Estimating Nonprogramming Tasks

Although most of the assignments in this book involve planning and developing small programs, two are for planning and producing reports. When working on almost any kind of project, the PSP methods can help you to plan, manage, and track your work. To estimate the time needed for the project's tasks, follow the decision tree shown in Figure 6.3. It illustrates the logic for selecting an estimating method. The following steps describe how to estimate the time needed to write a report that is estimated to have seven single-spaced pages. The developer's historical data are shown in Table 6.11 and the steps are numbered as shown in Figure 6.3.

1. Start with as clear an understanding of the task requirements as you can get.
2. Do you have a defined process for the task? It is difficult to make an intelligent plan with an ill-defined process for doing the work. It is also difficult to measure the work, track progress, or judge the quality of the result.
3. If the answer to step 2 is no, produce a simple process definition. This definition should divide the work into parts that represent between about 10% and 25% of the total job, and they must include the planning and postmortem steps. It is usually best to start with only four to eight steps. With experience and historical data, you can further refine this process.

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 103

TABLE 6.11 EXAMPLE REPORT-WRITING DATA

Job	Estimated Size *x*	Estimated Time	Actual Size	Actual Time *y*	*x*x*	*x*y*	*y*y*
494	6	900.00	4.00	556.00	36.00	3,336.00	309,136.00
498	6	600.00	5.00	575.00	36.00	3,450.00	330,625.00
501	6	830.00	5.00	1,003.00	36.00	6,018.00	1,006,009.00
502	3	415.00	5.00	431.00	9.00	1,293.00	185,761.00
507	5	600.00	3.50	431.00	25.00	2,155.00	185,761.00
Total	26	3,345.00	22.50	2,996.00	142.00	16,252.00	2,017,292.00

4. Do you have defined process measures? You need measures to plan your work, track your progress, and learn to make better plans in the future.
5. If your answer to step 4 is no, define at least size and time measures. For program development, include defect measures. For PSP planning purposes, you will need data for the time taken by each process step and the sizes of the products produced.
6. Do you have historical data? Historical data will help you to judge the size of the job and the effort required. The following example uses the data in Table 6.11 for writing documents.
7. Even if your answer to step 6 is no, you can usually identify some similar jobs and judge how long they took. If you have never measured any of your previous work, your estimates for new work could be off by three or more times. When estimates are this badly in error, they are generally too low. Under these conditions, using almost any relevant data will reduce your planning errors.
8. Estimate the size of the planned new product. In the example, the developer estimated that the report would require seven single-spaced pages.
9. To use PROBE, the estimated size data must correlate with the actual development time data with an $r >= 0.7$. In this case,

$$r = \frac{n * Sum(x * y) - Sum(x) * Sum(y)}{\sqrt{\left(n * Sum(x^2) - Sum(x)^2\right) * \left(n * Sum(y^2) - Sum(y)^2\right)}} =$$

$$r = \frac{5 * 16252 - 26 * 2996}{\sqrt{(5 * 142 - 26 * 26) * (5 * 2017292 - 2996 * 2996)}} = \frac{3364}{6144.5} = 0.5475$$

10. Because the answer to the question in step 9 is no, you must use PROBE method C and make an average estimate. With the data in Table 6.11, divide

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 104

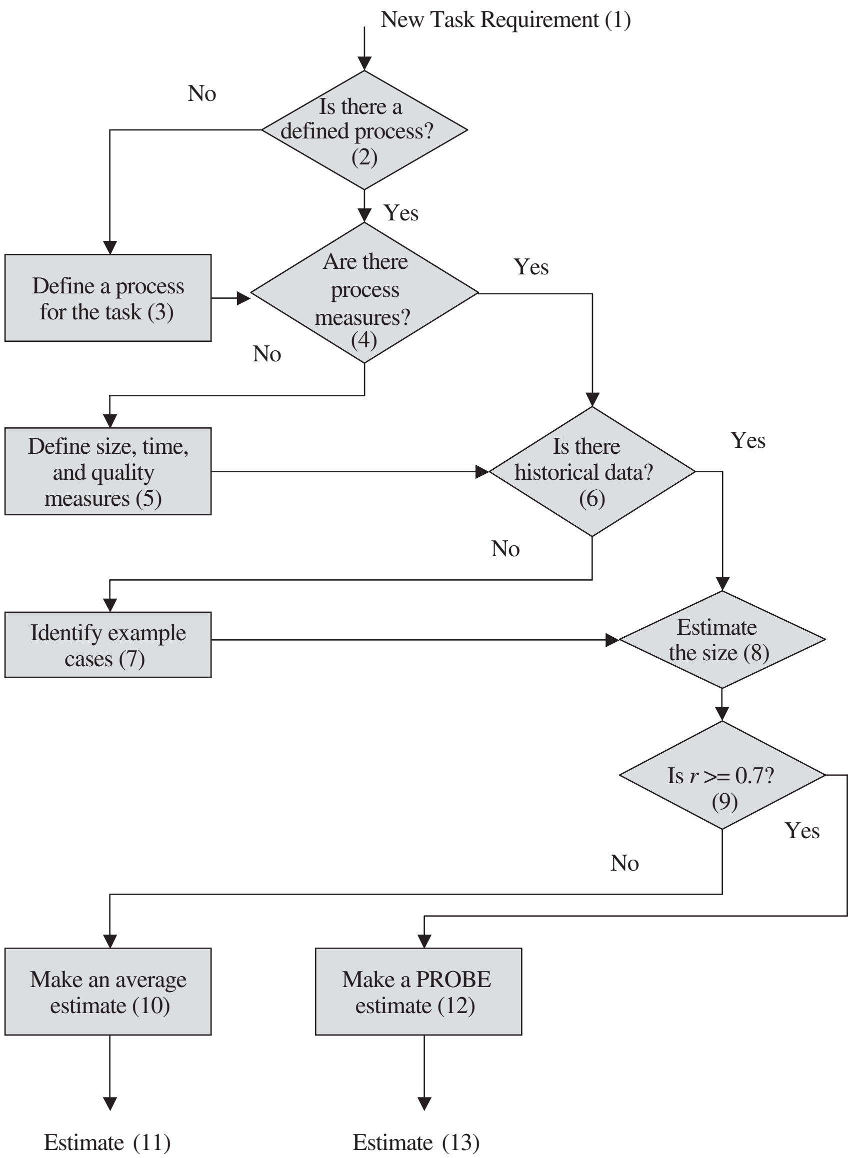

FIGURE 6.3 ESTIMATING NONPROGRAMMING TASKS

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 105

the total actual development time, 2,996 minutes, by the total estimated size of 26 pages to get a rate of 115.2 actual minutes per estimated page. For a job estimated to have seven pages, the development time would be 7 * 115.2 = 806.4 minutes or 13.4 hours.

11. Record the estimate.

12. If the answer to step 9 had been yes, you could have used PROBE method A or B. For method A, however, you would need a size proxy such as topic paragraphs with historical size data for various topic types and their sizes.

6.6 Considerations in Using PROBE

With experience and a growing volume of personal data, the PROBE method will help you to make progressively more accurate estimates and plans. The following discussion covers some of the more common questions about using PROBE.

Estimating Practice

While you are learning to make consistent estimates, your estimating accuracy will likely fluctuate widely, and your β_0 and β_1 parameters will change significantly from one program to the next. Over time, however, your estimates will tend to stabilize. Once the values of the regression parameters are reasonably stable, you can stop recalculating them for every program. Keep track of your estimating data, however, and periodically update the β parameters.

As you gain experience, your process will evolve and the data on some prior programs will no longer represent your current practice. When that occurs, drop the older data points and include only the newer and more representative ones. Always retain the data, of course, but only use data that represent how you currently work.

Estimating Judgment

In using any statistical method, you could get nonsensical results. For example, the β_0 and β_1 parameters represent a straight-line fit to the data. Because total program size is generally somewhat larger than estimated proxy size, you would expect β_1 to be somewhat larger than 1.0. A β_1 of 1.25, for example, would mean that on average, your total programs have an added and modified content that is about 25% larger than estimated part size.

If the data contain some wildly inaccurate estimates, however, it is possible to get almost anything for the β_0 and β_1 values. In fact, I have seen β_1 values as low as 0.05 and as high as 4.0 or more. A negative β_1 is even possible. When the value of β_1 is this far from 1.0, the linear regression method is likely to give misleading results. This would also be the case if the value of β_0 were a large positive or negative number. A large β_0 value would be one with an absolute value that was larger than the sizes of some small programs or a significant fraction of the size of the program you are currently estimating. When the linear regression parameters are obviously incorrect, use method C for the estimates. That is, add up all of the proxy size estimates and all of the actual added and modified sizes for these same programs. Then use the ratios of these totals to make the estimate.

Suppose the estimated proxy size for a new program is 137 LOC. Suppose also that you found the values of β_0 and β_1 to be 168 and 0.26, respectively. Even if your data correlated, when β_0 is large and β_1 is much less than 1.0, use an averaging method instead of linear regression. For example, if the sum of the proxy LOC estimates for the previous four programs was 427 LOC and the sum of the actual added and modified LOC for these same programs was 583 LOC, the ratio of these would be 1.365. In this case, set β_0 to 0 and β_1 to 1.365. With 137 proxy LOC for the new program, the added and modified size estimate for the new program would be 137 * 1.365 = 187 LOC.

Whenever either the size or time linear-regression parameters appear unreasonable, follow method C and use average values for the estimates. In addition, occasionally check your linear-regression estimates against an estimate made with average data. If the results are significantly different, check your calculations and reconsider the suitability of using linear regression. Differences of 20% to 30% should not concern you, but much larger errors, if they persist, could indicate that your estimating data is out of date or no longer appropriate. Furthermore, if your estimating accuracy starts to deteriorate, check your data and recalculate the β parameters.

Outliers

The reason for nonsensical values of β_0 and β_1 is often that you have included one or more **outlier** points in the data. An outlier is a data point where something unusual occurred, such as the requirements changed or you made an unusual mistake when producing the conceptual design. If you just made a larger-than-normal estimating error, you cannot discard that point. For every point you discard, describe in your assignment report the nature of the nonrecurring event that justified treating this point as an outlier. The reason to discard outlier points is that they can cause the regression method to produce nonsensical results. After removing any outliers, make sure you still have at least three points that show a reasonable correlation between program size and development hours—that is, $r >= 0.7$.

Another common cause of trouble is that all of the data are bunched in a narrow range. For example, if you had data for a large number of programs that all contained 75 to 100 database elements, you wouldn't have a wide enough range of data to give a reasonable trend. Another potential problem is having a lot of clustered data with one extreme point. For example, if you have 12 data points that all fall between 75 and 100 database elements and one point for 350 elements, you will likely get a high correlation. Unless you are confident that the extreme point was truly representative of your performance, the high correlation could be misleading. In that case, you should use method C.

Using Limited Data

Another problem, particularly when you are just starting to use the PSP, is having few historical data points. To use linear regression, you must have at least three sets of historical values. Therefore, you cannot use PROBE method A until you have estimated part data for at least three programs, and you can't use method B until you have estimated added and modified data for at least three programs. Until then, use method C, the averaging method, to make the size and resource estimates.

Overcompensation and Undercompensation

In addition to the overcompensation considerations discussed in Chapter 5 (see p. 83), normal statistical variation can also affect your estimating judgment. Analyze your estimating accuracy and make appropriate adjustments, but only after thoughtfully studying the data. If you attempt to correct your process for normal statistical fluctuations, you will almost certainly get worse results. The best approach is to consciously make corrections only when you change your process or identify erroneous or ineffective practices. Then maintain these changes consistently for several estimates, even if they appear to overcorrect for the previously observed problems.

Although you should follow a consistent estimating pattern and not worry about each error, this is almost impossible. The estimation process involves so many judgments that it is easily biased. The only reliable approach is to use a defined method and historical data. When you consistently use the same methods and are guided by the data, the statistical aberrations will be reduced. Though you must still make judgments, you can check them to ensure that they are reasonably balanced and consistent with the data. The regression method is essentially a way to use personal data to make estimates. If your estimates are consistently near the average values indicated by your data, your estimates will, on average, always be accurate.

31422 06 pp. 085-108 r3ah.ps 2/9/05 5:21 PM Page 108

6.7 Summary

The principal reason to estimate the size of a software product is to help you make plans. The quality of these plans will then depend on the quality of the estimates. This accuracy, in turn, will depend on the quality of the data and the soundness of the estimating methods you use. The PROBE method provides a statistically sound way to make estimates, and the PSP process will guide you in gathering and using the data you need to make accurate plans.

When using PROBE, you gather estimated and actual size and time data for each project. Then you divide the size data into categories and size ranges. The estimating process starts with a conceptual design followed by an estimate of the sizes of the proxy parts in that design. With the total estimated proxy size, the PROBE method shows you how to estimate program size and development time.

When estimating a large job as a single unit, there are two sources of error. First, estimating accuracy will fluctuate around some mean or average. Second, the estimate will typically have some bias. If you gathered data on many estimates and calculated the biases, and if the estimating process were reasonably stable, you could make an accurate bias adjustment. This, in fact, is what the PROBE method does.

Without data and without a defined estimating method, your estimates often will be inconsistent and unpredictable. By using the PROBE method and historical data on your personal work, you can minimize this error. Then, as you gather data and gain experience, you will make progressively better estimates. Of course, because estimating is an uncertain process, you should expect some estimating error. With the PROBE method, however, you can consistently make balanced estimates and judge the accuracy of your estimates.

6.8 Exercises

The assignment for this chapter is to use PSP1 to write one or more programs. The PSP1 process and the program specifications are given in the assignment kits, which you can get from your instructor or at **www.sei.cmu.edu/tsp/psp**. Each kit contains the assignment specifications, an example of the calculations involved, and the PSP1 process scripts. In completing this assignment, faithfully follow the PSP1 process, record all required data, and produce the program report according to the specifications given in each assignment kit.