Estimating With PROBE

Personal Software Process^{SM}
for Engineers: Part I

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS MATERIAL IS FURNISHED ON AN “AS-IS” BASIS WITH NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, ANY WARRANTY WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT, OR THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

Lecture Topics

Planning overview

Why estimate size?

Size estimating principles

The PROBE estimating method

Size estimating proxies

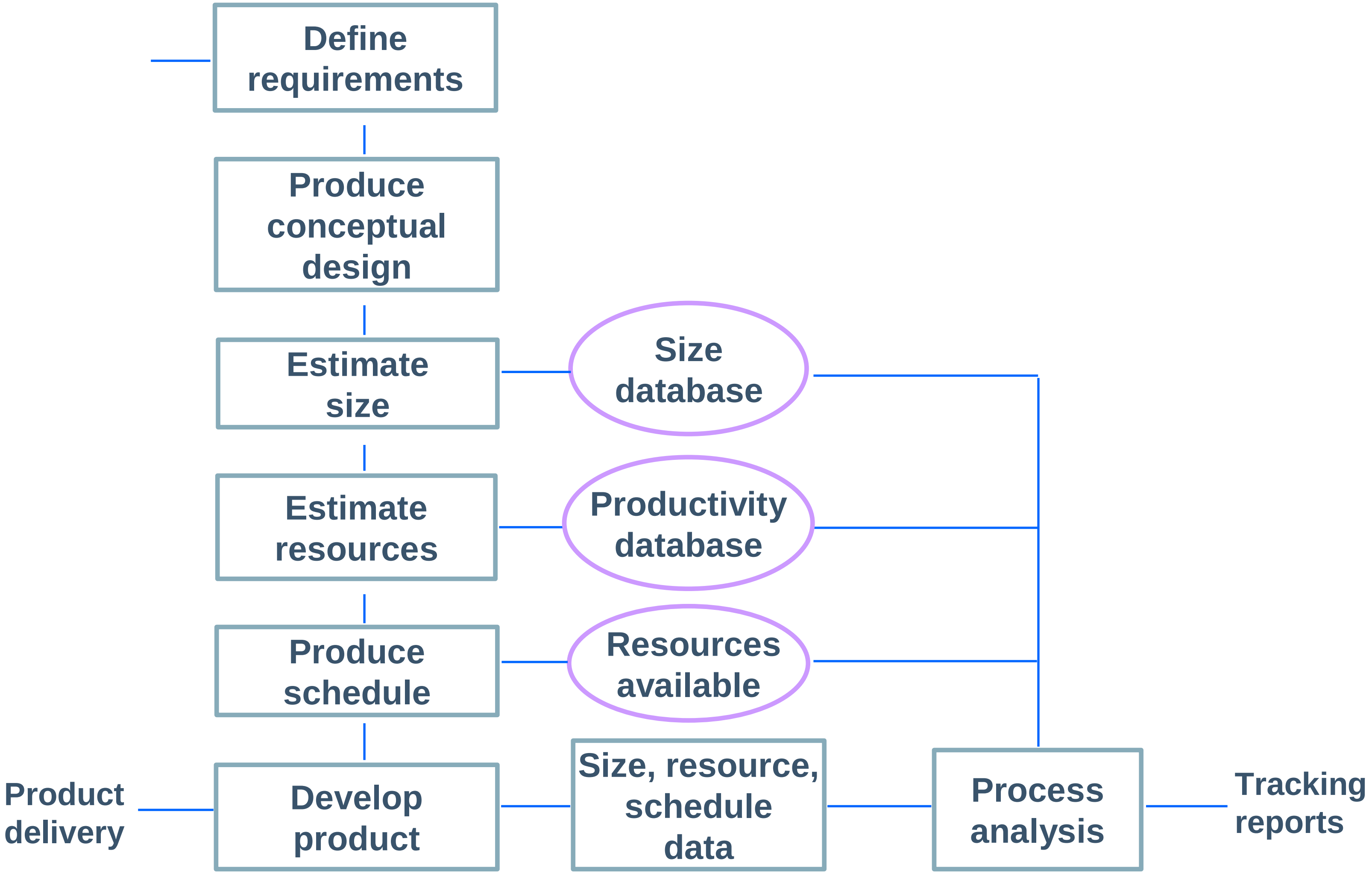
Estimating examples

Planning Overview

Plans

- allow you to make commitments that you can meet
- provide the basis for agreeing on job scope, schedule, and resources
- guide the work
- facilitate progress tracking and reporting
- help ensure that key tasks are not overlooked

The Project Planning Framework



The Planning Process

Before making the plan, you must have a requirement.

- The better the requirement, the better the plan.
- With ill-defined requirements, expect to make frequent plan updates.

Plans are most accurate when based on size estimates and historical data.

With a size estimate and historical data, you can

- identify the data on the most similar prior work
- base the resource estimate on these data

Size Estimating Principles -1

Estimating is an uncertain process.

- No one knows how big the product will be.
- The earlier the estimate, the less is known.
- Estimates can be biased by business and other pressures.

Estimating is an intuitive learning process.

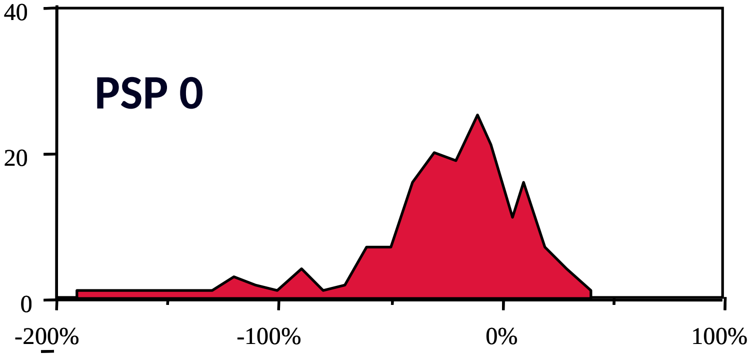
- Ability improves with experience and data.
- Some people will be better at estimating than others.

The objective is to become consistent.

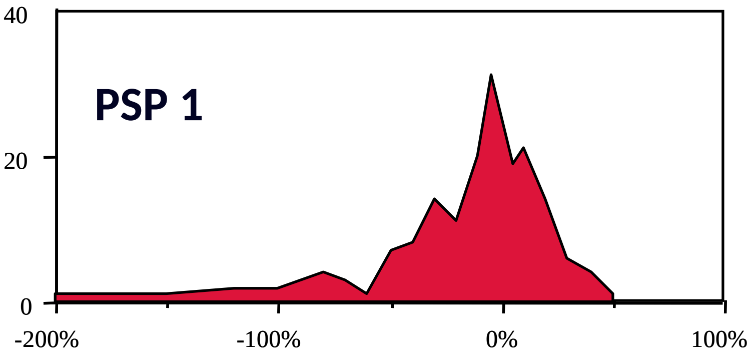
- You will then understand the variability of your estimates.
- You seek an even balance between under- and overestimates.

Balanced Estimates

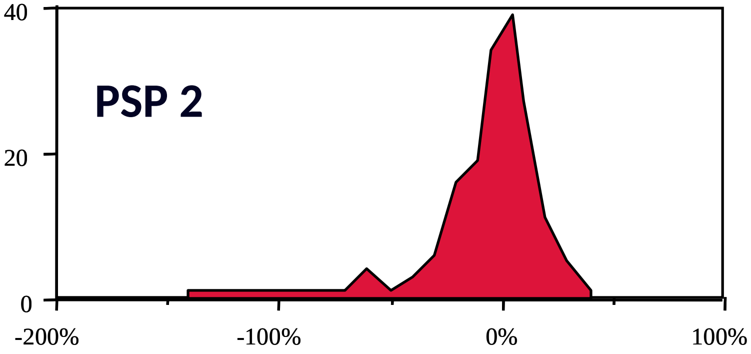
Majority are under-estimating



Balance of over- and underestimates



Much tighter balance around zero



Size Estimating Principles -3

The advantages of using defined estimating methods are that you

- have known practices that you can improve
- have a framework for gathering estimating data
- can consistently use historical data to produce balanced estimates

Estimating with PROBE

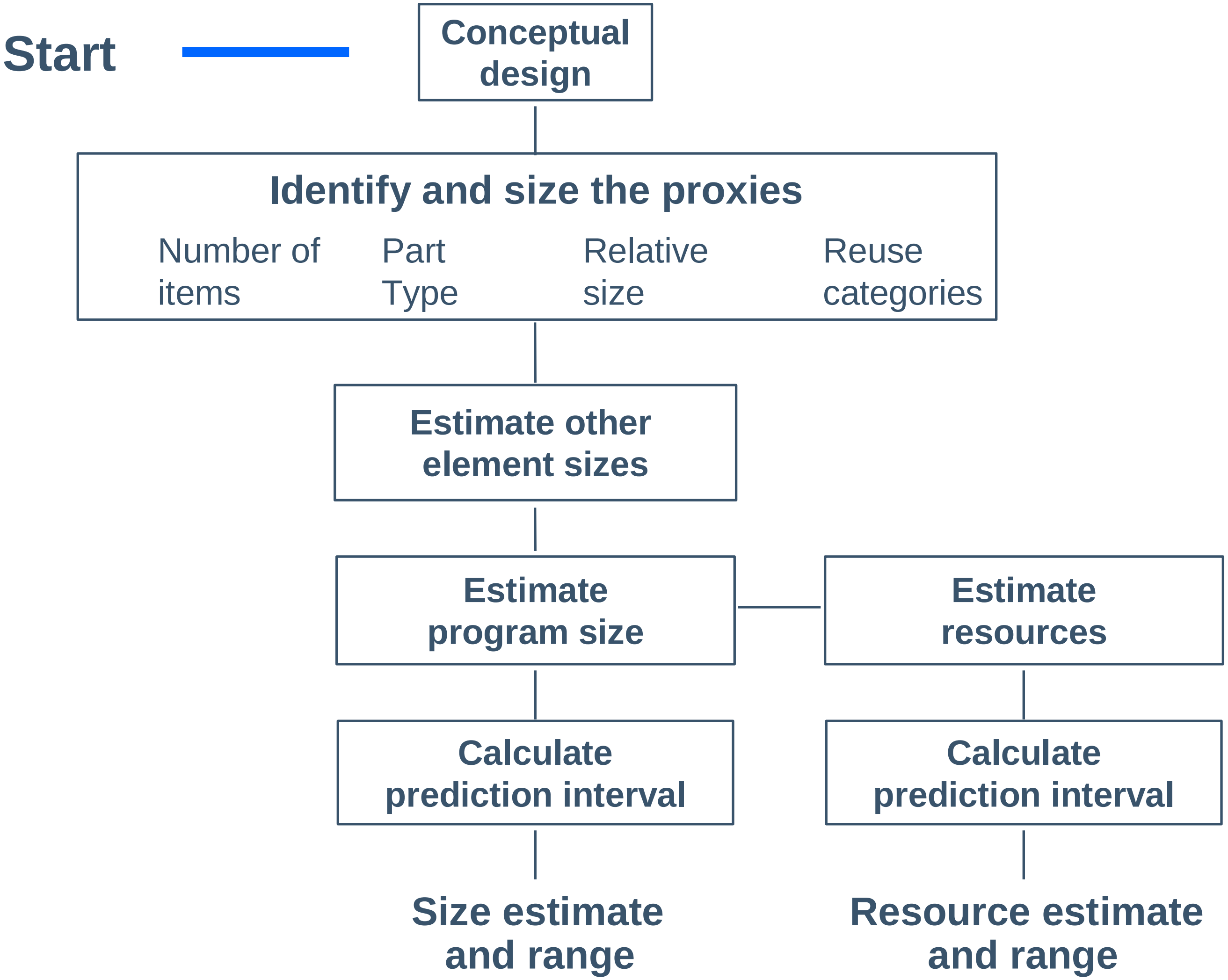
The PSP uses the PROBE method to estimate and plan projects.

PROBE stands for PROxy Based Estimating.

PROBE uses proxies to estimate program size and development time.

A good proxy will help you to make accurate estimates.

The PROBE Estimating Method


Start
Conceptual design
Identify and size the proxies
Number of items
Part Type
Relative size
Reuse categories
Estimate other element sizes
Estimate program size
Estimate resources
Calculate prediction interval
Calculate prediction interval
Size estimate and range
Resource estimate and range

Conceptual Design -1

The first estimating step is to make a conceptual design:

- relate the requirements to the product,
- define the product elements that will produce the desired functions,
- estimate the size of what you plan to build.

For most projects, the conceptual design can be produced relatively quickly.

For the PSP exercise programs, try to limit your conceptual design time to 10, or at most 20, minutes.

Conceptual Design -2

To make a conceptual design, identify the product functions and the program parts needed to produce them.

In effect, you say: “If I had the following parts, I could build this product.”

Then, compare these parts to programs you have already written and estimate their sizes.

Finally, combine the part estimates to give total size.

If you do not understand the product well enough to make a conceptual design, you do not know enough to make a plan.

Size Estimating Proxies

A good proxy should correlate closely to development costs.

A good proxy should be easy to visualize early in development.

It should also be a physical entity that can be measured.

Example: Building Costs

Problem

- The builder needs to know the floor area (in sq. ft.) to estimate the cost of construction.
- Clients normally cannot describe their needs in square feet.
- They usually can describe the type and number of rooms they want.

Solution

- Use rooms as a proxy for size.
- Use historical (typical) data to translate from rooms to square feet.

Example: Customer Requirements

Bedrooms:	1 large, 2 medium, and 1 small
Bathrooms:	1 large and 2 medium
Kitchen:	1 medium
Living room:	1 large
Dining room:	1 medium
Family room:	1 large
Utility:	1 medium

Historical Building Data

	Small	Medium	Large
Bedrooms	90	140	200
Bathrooms	25	60	120
Kitchens	100	130	160
Living rooms	150	250	400
Dining rooms	100	140	200
Family rooms	150	240	340
Utility	25	50	80

Proxy Calculation

Bedroom	1 large	=	1 x	200	=	200
Bedroom	2 medium	=	2 x	140	=	280
Bedroom	1 small	=	1 x	90	=	90
Bathroom	1 large	=	1 x	120	=	120
Bathroom	2 medium	=	2 x	60	=	120
Kitchen	1 medium	=	1 x	130	=	130
Living room	1 large	=	1 x	400	=	400
Dining room	1 medium	=	1 x	140	=	140
Family room	1 large	=	1 x	340	=	340
Utility	1 medium	=	1 x	50	=	50
			Total (sq. ft.)		=	1870

Example: The Builder's Estimate

The first estimation step provides the builder with the proxy data for room size.

However, there are many other cost elements in home construction.

Builders typically have extensive data to relate room size to the other building costs.

With agreed initial plans, builders typically require detailed architectural specifications and estimates before quoting a price.

Proxies for estimating software

Classes, functions, and procedures:

- When classes are selected as application entities, they can be visualized early in development.
- Functions and procedures can often be estimated in the same way.
- Classes, functions, procedures, and their sizes can be automatically counted.

Correlation with development hours:

- Numbers of classes correlates reasonably well.
- Class size correlates very closely.

With a good correlation, calculate program size from the relationship between class size and program size, considering historic data.

Estimating with Proxies

Once you have selected a proxy, you must:

- obtain proxy data,
- organize the data for use in estimating,
- use the data to estimate the size of the proxies in the new program,
- combine the proxy estimates into the product estimate,
- make a resource estimate,
- produce a project plan.

The PROBE method shows you how to perform these steps.

Organizing Proxy Data

A common way to estimate is to:

- break the planned product into parts,
- relate these planned parts to parts you have already built,
- use the size of the previously built parts to estimate the
- sizes of the new parts.

To do this, you need size ranges for the types of parts you typically develop.

For each product type, you also need size ranges to help judge the sizes of the new parts.

Example C++ Class Size Ranges

			LOC per item		
Type	**VS**	**S**	**M**	**L**	**VL**
Calculation	2.34	5.13	11.25	24.66	54.04
Data	2.60	4.79	8.84	16.31	30.09
I/O	9.01	12.06	16.15	21.62	28.93
Logic	7.55	10.98	15.98	23.25	33.83
Set-up	3.88	5.04	6.56	8.53	11.09
Text	3.75	8.00	17.07	36.41	77.66

Estimating Program Size

Just as homes have square feet that are not in rooms, programs have code that is not in the program parts.

- includes,
- declarations,
- other overhead functions.

The development job, however, must also produce this overhead code.

The size of this additional overhead code is usually proportional to the size of the program's parts.

Estimating Development Time

With sound estimating methods, actual program size will be closely related to estimated program size.

The differences will be due to the overhead code and the estimating error.

Actual development time is also often related to estimated program size.

Again, with sound methods, the differences will be largely due to overhead activities and estimating error.

Statistically-Based Estimates

PROBE uses historical data, linear regression, and the prediction interval to produce estimates of known accuracy.

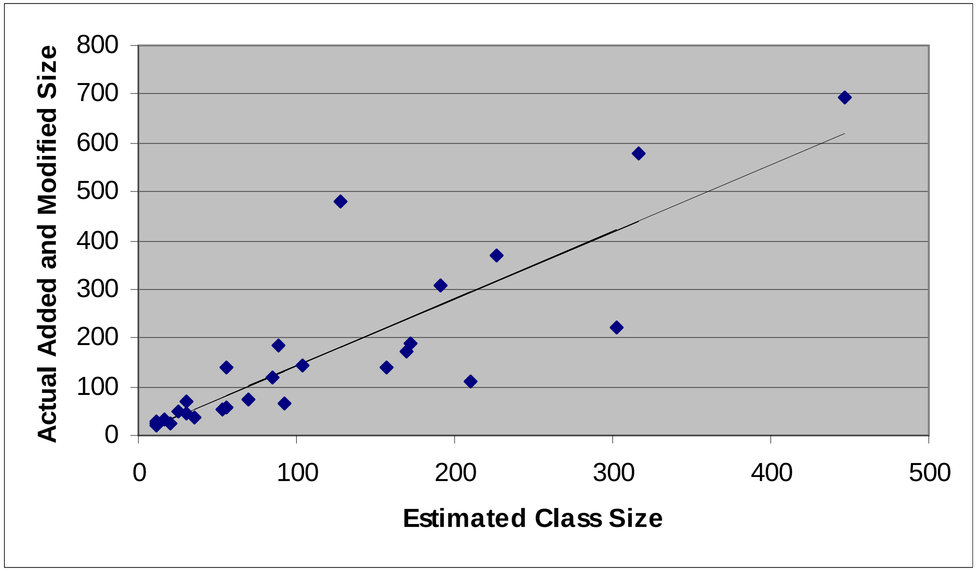
Regression provides the best fit, or minimum variance, of a line to these data.

The variance of the data is used to determine the likely estimation error.

The greater the variance, the larger the likely error.

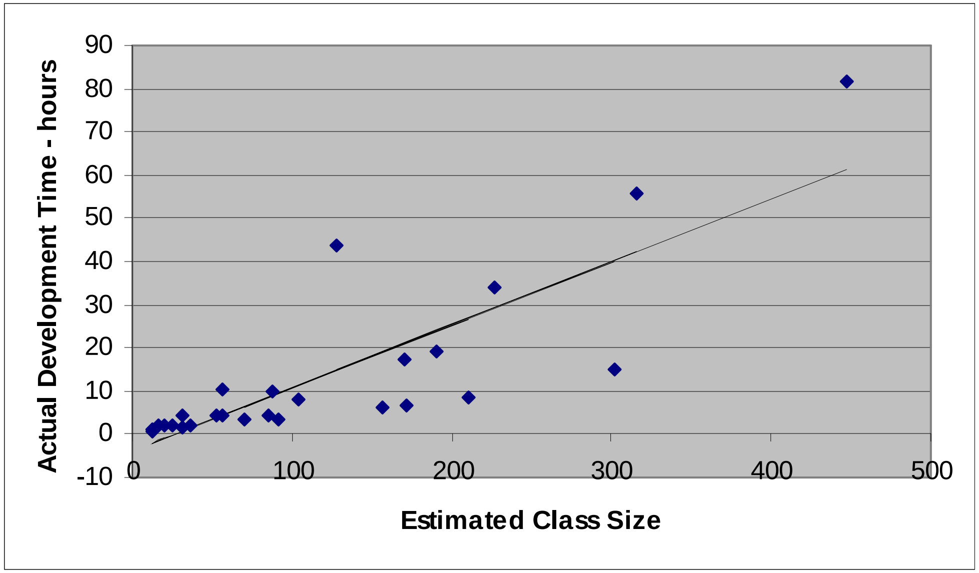
Regression Line for Program Size

27 C++ programs



Regression Line for Development Time

27 C++ programs



PROBE: Estimating methods

Depending on the quality of your data, select one of the four PROBE estimating methods.

Method	
A	**regression with estimated proxy size**
B	**regression with planned added and modified size**
C	**the averaging method**
D	**engineering judgment**

To use regression method A or B, you need:

- a reasonable amount of historical data,
- data that correlate,
- reasonable β_0 and β_1 parameter values.

Method A (Regression): Estimated Proxy Size

Method A uses the relationship between estimated proxy size (E) and actual

- added and modified size
- development time

The criteria for using this method are

- three or more data points that correlate (R2 > 0.5)
- reasonable regression parameters (table 6.6 on pg. 96)
- completion of at least three exercises with PSP1 or higher

Method B (Regression): Plan Added and Modified Size

Method B uses the relationship between plan added and modified size and

- actual added and modified size
- actual development time

The criteria for using this method are

- three or more data points that correlate (R2 >0.5)
- reasonable regression parameters (table 6.6 on pg. 96)
- completion of at least three exercises with PSP0.1 or higher

Method C: Averaging

Method C uses a ratio to adjust size or time based on historical averages.

The averaging method is easy to use and requires only one data point.

Averages assume that there is no fixed overhead.

Method D: Engineering Judgment

Use method D when you don’t have historical data.

Use judgment to

- project the added and modified size from estimated part size
- estimate development time

Use method D when you cannot use methods A, B, or C.

Method A: Estimating Size and Time -1

To project a program's size or development time, find the equation for the regression line.

Then use the regression formula to project size and time.

E is the estimated proxy size.

$$y_k = \beta_0 + \beta_1 * x_k$$

$$Added \,\&\, Modified\_Size = \beta_{0size} + \beta_{1size} * (E)$$

$$Development\_Time = \beta_{0time} + \beta_{1time} * (E)$$

Method A: Estimating Size and Time -2

Calculate the regression parameters b0 and b1 from data on previously developed programs.

For x, use estimated proxy size (E).

For y, use the actual

- added and modified size for the size estimate
- total development time for the time estimate

Calculate two sets of b0 and b1 regression parameters: one for size and the other for time.

Method A: Calculating Total Program Size

Total program size includes added, modified, deleted, base, and reused code.

When modifying an existing program, base code is the size of the unmodified existing program.

When modifying programs, include their unmodified size in base code and not in reuse.

While base code is a form of reuse, the PSP only counts unmodified code from the reuse library as reused.

Method A: Completing the Estimate

To complete the size estimate, calculate the

- projected added and modified size with the size regression parameters
- total program size, including added, modified, deleted, base, and reused code
- estimated new reusable code to be added to the reuse library

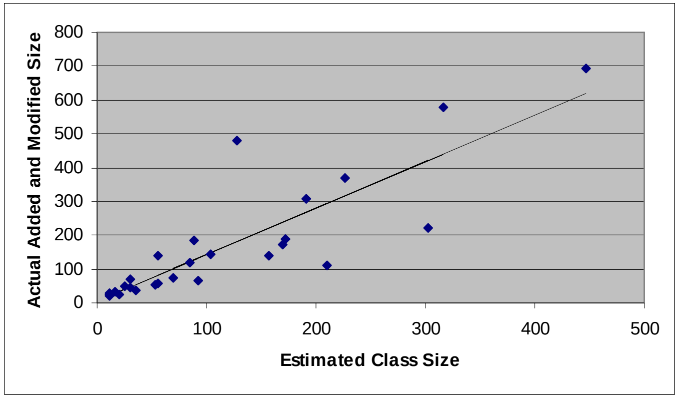
The completed estimate includes estimated development time calculated with the time regression parameters.

Method A: Completed Example -1

Base program (B)	695 LOC
Deleted (D)	0 LOC
Modified (M)	5 LOC
Base additions (BA)	0 LOC
Added parts (PA)	115+197+49 = 361LOC
Reused parts (R)	169 LOC
Estimated size E	= BA + PA + M = 366 LOC

Method A: Completed Example -2

Starting with E = 366, use the size regression parameters to calculate the projected size (P).



β_{0Size} = 62 and β_{1Size} = 1.3

Projected size P = 62 + 1.3 * 366 = 538 LOC

Total size T = 538 + 695 - 5 + 169 = 1397 LOC

Estimated total new reusable = 49 LOC

Note: Modified size is subtracted so it is not included twice.

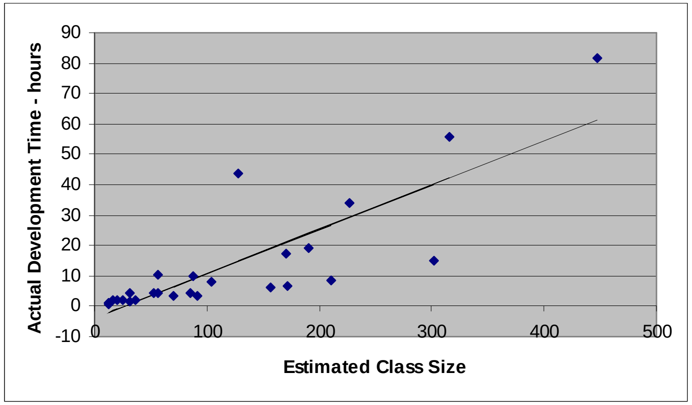
Method A: Completed Example -3

Starting with E = 366, use the time regression parameters to calculate development time.

$$Development\_Time = \beta_{0time} + \beta_{1time} * (E)$$

β_{0Time} = 108 and β_{1Time} = 2.95

Development time = 108 + 2.95 * 366 = 1186 min.



Messages to Remember

Accurate size estimates will help you to make better development plans.

Estimating skill improves with practice.

A defined and measured process provides a repeatable basis for improvement.

To make accurate estimates, you must use historical data and follow sound methods.

The PROBE method shows you how to do this.