Software Size Estimation II

Material adapted from:
Disciplined Software Engineering
Software Engineering Institute
Carnegie Mellon University

Estimating Software Size

Size estimating overview

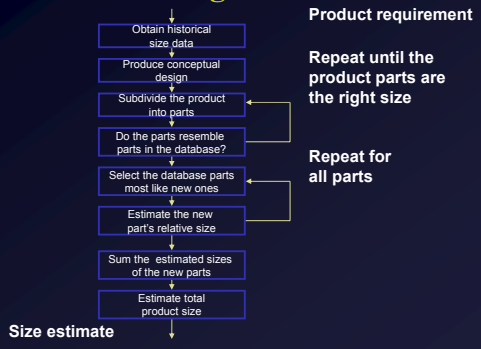
The PROBE estimating method

Categorizing object data

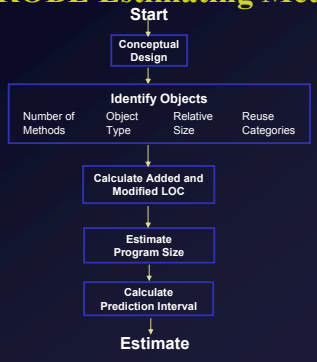
The regression method

Process additions

Size Estimating Overview



The PROBE Estimating Method



Conceptual Design

A conceptual design is needed

- to relate the requirements to the product
- to define the product elements that will produce the desired functions
- to estimate the size of what will be built

For understood designs, conceptual designs can be done quickly.

If you do not understand the design, you do not know enough to make an estimate.

Identify the Objects - 1

Where possible, select application entities.

Judge how many methods each object will likely contain.

Determine the type of the object, i.e.: data, calculation, file, control, etc.

Judge the relative size of each object: very small (VS), small (S), medium (M), large (L), very large (VL).

Identify the Objects - 2

From historical object data, determine the size in LOC/method of each object.

Multiply by the number of methods to get the estimated object LOC.

Judge which objects will be added to the reuse library and note as "New Reused."

Identify the Objects - 3

When objects do not fit an existing type, they are frequently composites.

- Ensure they are sufficiently refined
- Refine those that are not elemental objects

Watch for new object types

Estimate Program Size - 1

Total program size consists of

- newly developed code (adjusted with the regression parameters)
- reused code from the library
- base code from prior versions, less deletions

Newly developed code consists of

- base additions (BA) - additions to the base
- new objects (NO) - newly developed objects
- modified code (M) - base LOC that are changed

Estimate Program Size - 2

Calculate the new and changed LOC from the newly developed code

- BA+NO+M
- use regression to get new and changed LOC

$$New\&Changed = \beta_0 + \beta_1 * (BA + NO + M)$$

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

The regression parameters are calculated from historical data on prior estimated newly developed (object) LOC and actual new and changed LOC.

Estimate Program Size - 3

Code used from the reuse library should be counted and included in the total LOC size estimate.

Base code consists of:

- LOC from the previous version
- subtract deleted code
- subtract modified code (or it would be counted twice)

Completing the Estimate

The completed estimate consists of:

- the estimated new and changed LOC calculated with the regression parameters
- the 70% and 90% upper prediction interval (UPI) and lower prediction interval (LPI) for the new and changed LOC
- the total LOC, considering base, reused, deleted, and modified code
- the projected new reuse LOC to be added to the reuse library

Completed Example - 1

Base Program (B)	695 LOC
Deleted (D)	0 LOC
Modified (M)	5 LOC
Base Additions (BA)	0 LOC
New Objects: NO = 115+197+49 =	361 LOC
Reused Programs	169 LOC

Completed Example - 2

Use the regression parameters to calculate New and Changed LOC (N):

$$New\&Changed = \beta_0 + \beta_1 * (BA + NO + M)$$

Added code: BA + NO +M = 366 LOC

New and changed: N = 62 + 366*1.3 = 538 LOC

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

To Make Size Estimates, You Need Several Items

Data on historical objects, divided into types

Estimating factors for the relative sizes of each object type

Regression parameters for computing new and changed LOC from:

- estimated object LOC
- LOC added to the base
- modified LOC

Historical Data on Objects

Object size is highly variable
- depends on language
- influenced by design style
- helps to normalize by number of methods

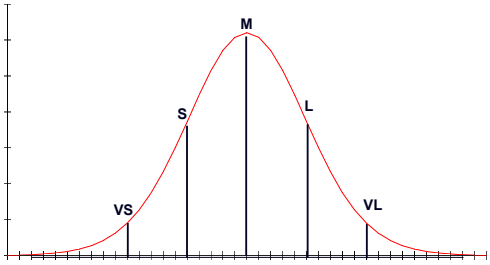
Pick basic types
- logic, control
- I/O, files, display
- data, text, calculation
- set-up, error handling

Estimating Factors for Objects

You seek size ranges for each type that will help you judge the sizes of new objects.

To calculate these size ranges
- take the mean
- take the standard deviation
- very small: VS = mean - 2*standard deviations
- small: S = mean - standard deviation
- medium: M = mean
- large: L = mean + standard deviation
- very large: VL = mean + 2*standard deviations

Normal Distribution with Siz Ranges



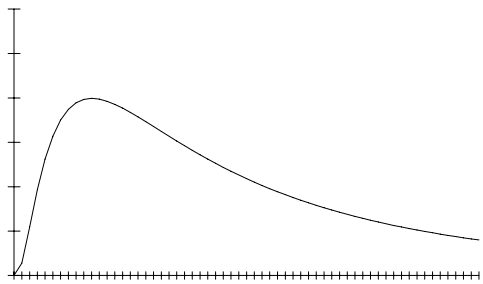
Log-Normal Distribution

These size ranges assume the object data are normally distributed.

If the data are log-normally distributed, take the log of the data before making the size range calculations.

Then, after computing the size ranges, take the antilog to get the factors in LOC

A Log-Normal Distribution



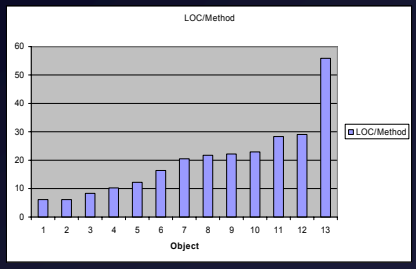
Example

Object	# of Methods	LOC	LOC/Method
A	3	31	10.3
B	3	18	6
C	4	87	21.8
D	3	87	29
E	3	25	8.3
F	3	18	6
G	4	89	22.3
H	3	85	28.3
I	3	37	12.3
J	10	558	55.8
K	4	82	20.5
L	5	82	16.4
M	10	230	23
Min	3	18	6
Max	10	558	55.8
Average	4.5	109.9	20
StD	2.5	145.6	13.4

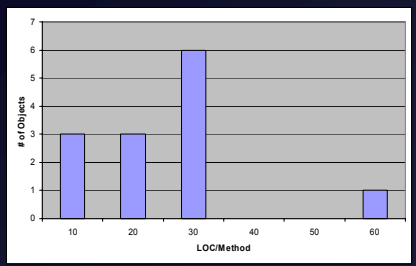
Ranges

Very small = Avg – 2*StD = -6.8
Small = Avg –StD = 6.6
Medium = Avg = 20
Large = Avg + StD = 33.4
Very large = Avg + 2*StD = 46.8

LOC/Method



LOC/Method Distribution



Estimating Factors - 1

Object	# of Methods	LOC	LOC/Method	ln(LOC/Method)
B	3	18	6	1.7918
F	3	18	6	1.7918
E	3	25	8.3	2.1163
A	3	31	10.3	2.3321
I	3	37	12.3	2.5096
L	5	82	16.4	2.7973
K	4	82	20.5	3.0204
C	4	87	21.8	3.0819
G	4	89	22.3	3.1046
M	10	230	23	3.1355
H	3	85	28.3	3.3429
D	3	87	29	3.3673
J	10	558	55.8	4.0218
Min	3	18	6	1.7918
Max	10	558	55.8	4.0218
Average	4.5	109.9	20	2.801
StD	2.5	145.6	13.4	0.6612

Estimating Factors - 2

Calculate ln(LOC/Method)

Compute average and standard deviation

Very small = Avg – 2*StD = 1.4789

Small = Avg –StD = 2.1398

Medium = Avg = 2.801

Large = Avg + StD = 3.4622

Very large = Avg + 2*StD = 4.1234

Estimating Factors - 3

From these log size ranges, the LOC ranges are obtained by taking the antilog

- very large - VL: exp(4.1234) = 61.8
- large - L: exp(3.4622) = 31.9
- medium - M: exp(2.801) = 16.5
- small - S: exp(2.1398) = 8.5
- very small - VS: exp(1.4789) = 4.4

Repeat these calculations for every object type

C++ Object Size Ranges

Type	LOC per method 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

The Regression Parameters

Using *estimated* object LOC (x) and *actual* new and changed LOC (y):

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

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

Example

Historical data is (Estimate, Actual) pairs
Take: (30, 40), (40, 42), (50, 48)
Average is (40, 43.3)

B1 = (5280-5196) / (5000-4800) = 0.42
B0 = 43.3 – 0.42*40 = 26.5

So
Program size = 26.5 + (Estimate * 0.42)

The Prediction Interval - 1

The prediction interval provides a likely range around the estimate

- a 90% prediction interval gives the range within which 90% of the estimates will likely fall
- it is not a forecast, only an expectation
- it only applies if the estimate behaves like the historical data

It is calculated from the same data used to calculate the regression factors.

The Prediction Interval - 2

The lower prediction interval (LPI) and upper prediction interval (UPI) are calculated from the size estimate and the range where

- LPI = Estimate - Range
- UPI = Estimate + Range

$$Range = t(\alpha/2, n-2)\sigma\sqrt{1+\frac{1}{n}+\frac{(x_k - x_{avg})^2}{\sum_{i=1}^{n}(x_i - x_{avg})^2}}$$

The Prediction Interval - 3

The t distribution is for

- the two-sided distribution (alpha/2)
- n-2 degrees of freedom

Sigma is the standard deviation of the regression line from the data.

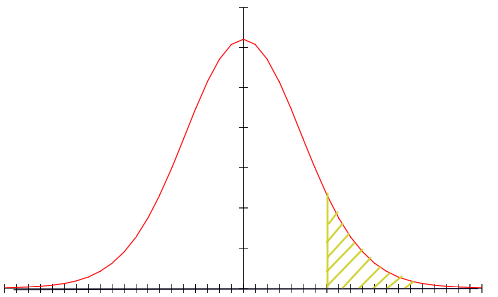
$$\sigma = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i - \beta_0 - \beta_1 x_i)^2}$$

The t Distribution

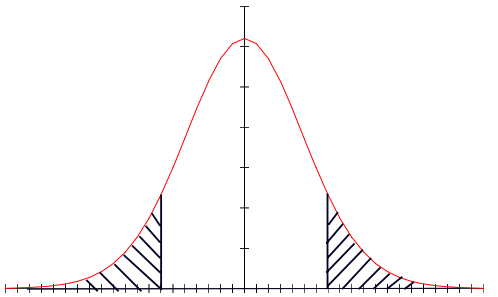
The t distribution
- is similar to the normal distribution
- has fatter tails
- is used in estimating statistical parameters from limited data

t distribution tables
- typically give single-sided probability ranges
- we use two-sided values in the prediction interval calculations

The Single-Sided t Distributio



The Double-Sided t Distributio



t Distribution Values

Statistical tables give the probability value p from minus infinity to x

For the single-sided value of the tail (the value of interest), take 1-p

For the double-sided value (with two tails), take 1 - 2*(1 - p) = 2p - 1
- look under p = 85% for a 70% interval
- look under p = 95% for a 90% interval

Prediction Interval Example

Calculate the range from historical data
Range = 235 LOC

Upper prediction interval (UPI)
UPI = N + range = 538 + 235 = 773 LOC

Lower prediction interval (LPI)
LPI = N - range = 538 - 235 = 303 LOC

Size Estimating Calculations

When completing a size estimate, you start with the following data
- new and changed LOC (N): estimate
- modified (M): estimated
- the base LOC (B): measured
- deleted (D): estimated
- the reused LOC (R): measured or estimated

And calculate
- added (A): N-M
- total (T): N+B-M-D+R

Actual Size Calculations

When determining actual program size, you start with the following data

- the total LOC (T): measured
- the base LOC (B): measured
- deleted (D): counted
- the reused LOC (R): measured or counted
- modified (M): counted

And calculate

- added (A): T-B+D-R
- new and changed (N): A+M

Messages to Remember

1 - The PROBE method is a structured way to make software size estimates.

2 - It uses your personal size data.

3 - It provides a statistically sound range within which the actual program size will most likely fall.