Determining the Completed Effort of Adapting Existing Software

Ronald Cobb, Linda Smeraglinolo, and Dave Wood
Harris Corporation/GCSD

This paper describes a method for validating the estimated equivalent number of delivered source instructions determined using the Boehm adaptation adjustment factor.

THE METHOD FOR ESTIMATING effort to adapt existing software using the adaptation adjustment factor will become more prevalent as software developers turn increasingly to reuse as a way to cut schedule and development costs. To determine the accuracy of our estimates, we must develop a method to assess completed projects, which includes reused, modified, and new source code, and compare this to our original estimation. This paper proposes a simple method for such an assessment.

Due to the nature of modified code, the calculations in this paper are made in physical, rather than logical source, instructions.

Background

Determining the effort and schedule of proposed software work generally involves estimating the total source lines of code to be developed, then using a cost estimation tool such as Constructive Cost Mode (COCOMO) or REVIC. Estimating software development costs with reuse, as proposed by Barry Boehm [1], involves arriving at an equivalent number of delivered source instructions (EDSI). EDSI is the amount of source instructions which could be developed in the time required to adapt the reused software to meet current requirements. The formula uses the percent of design modification (DM), the percent of code modification (CM), and the percent of integration expected (IM). The EDSI is calculated by using an adaptation adjustment factor (AAF) as follows:

AAF = 0.4(DM) + 0.3(CM) + 0.3(IM)
EDSI = (reused code)*(AAF/100) + (new code)

For example: Reusing 1,000 lines of existing code with 50 percent design modification, 50 percent code modification, and 50 percent integration with 1,000 new lines of code would be equivalent to developing 1,500 new lines of code.

Assessment Method

The method we use for determining the EDSI when a project is completed requires four data points. The first two values are easily determined with a code counter. They are the amount of the original code (OC) and the amount of the delivered code (DC). The next two values are determined by running a UNIX diff command on each file in the original code with the same file in the delivered code. Both files must be processed to remove comments. The output from the diff command is filtered for lines removed and lines added and piped to a saved file. For example:

```
diff  ./old/main.c  ./new/main.c | grep '<'   > ./removed/main.c
diff  ./old/main.c  ./new/main.c | grep '>'   > ./added/main.c
```

The files in the 'removed' directory contain the lines of code not in the new files (either deleted or changed), and the files in the 'added' directory contain the lines of code not in the old files (either added or changed). The total number of lines in all the files in the 'removed' directory (REM) and the total number of lines in all the files in the 'added' directory (ADD) are used to determine the AAF as follows:

a) The actual new code is the difference between the delivered code and the original code: ACT_NEW = DC - OC
b) The actual modified code is the difference between the 'added' directory and actual new code: ACT_MOD = ADD - ACT_NEW
c) The actual modified code is the difference between the 'removed' directory and actual changed code: ACT_REMOVED = REM - ACT_MOD
d) The percent of code modification is the actual modified code plus the actual removed code, divided by the amount of the original code: CM = (ACT_MOD + ACT_REMOVED) / OC
e) Since we have no accurate method for determining the amount of design modification, we set it equal to the amount of code modification: DM = CM
f) The amount of integration is set to the percentage of original test cases rerun: IM = (rerun tests / total original tests).

The AAF is then applied to determine the EDSI:

g) The actual reused code is a product of the AAF and the original code: ACT_REUSED = AAF * OC
h) The EDSI is the sum of the actual new code and the actual reused code. EDSI = ACT_NEW + ACT_REUSED

For example: A software system contains 45,000 lines of code before enhancements (OC) and 63,000 after (DC). The total source lines in the 'removed' directory is 16,000 (REM), and the total source lines in the 'added' directory is 28,000 (ADD). If all of the original test cases are rerun (that is IM = 100 percent), then:

ACT_NEW = DC - OC = 63,000 - 45,000 = 18,000

ACT_MOD = 28,000 - 18,000 = 10,000

ACT_REMOVED = 16,000 - 10,000 = 6,000

CM = (10,000 + 6,000) / 45,000 = 35.5 percent

DM = 35.5 percent {same value as CM}

IM = 100 percent

AAF = 0.4(DM) + 0.3(CM) + 0.3(IM) = 0.4(35.5) + 0.3(35.5) + 0.3(100) = 54.85
{0.4,0.3,0.3 are the coefficents suggested by Boehm for DM, CM, and IM}

ACT_REUSED = (AAF /100) * OC = 24,682 = (54.85/100) * 45,000 = 24,682

EDSI = ACT_NEW + ACT_REUSED = 18,000 + 24,682 = 42,682

Conclusion

Using a method such as this allows software project engineers to verify the accuracy of their estimates by comparing the planned effort against actual performance. As stated in the introduction, these calculations are for physical source instructions. If metrics are kept in logical source instructions, a ratio of physical to logical source instructions can be provided by most code counters. This ratio can then be used as a multiplier on the physical source instructions to determine the logical. ◆

About the Authors

Ronald Cobb is a software engineer with the Harris Corporation Government Communication Systems Division (GCSD). He has more than 14 years experience in software development, software project leadership, and software process improvement. Cobb was formerly the manager of the GCSD software engineering process group, a position he held for three years. He has a bachelor's degree in computer science from the University of Colorado, Boulder, and a master's degree in business administration with a technical leadership focus from Florida Tech.

Harris Corp.
PO Box 9100 (MS: 5-5872)
Melbourne, Fla. 32902
Voice: 407-726-1239
Fax: 407-727-4555
E-mail: rcobb01@harris.com

Linda Smeraglinolo is a member of the engineering process group (EPG) at Harris Corp. She has 17 years of software experience, including three years in process improvement. She has held a variety of positions on numerous software projects prior to joining the EPG. She has a bachelor's degree in computer science from Rollins College and a master's degree in business administration from Florida Tech.

Harris Corp.
PO Box 9100 (MS: 5-5872)
Melbourne, Fla. 32902
Voice: 407-727-6926
Fax: 407-727-4555
E-mail: lsmeragl@harris.com

Dave Wood is a software engineer with the Harris Corporation Government Communication Systems Division (GCSD). He has 12 years experience as a software developer, software tester, and software instructor in object- oriented and C++ development. He has a bachelor of science degree in computer science from the University of North Florida, and a master of science degree in computer science with emphasis in Artificial Intelligence from Florida Tech.

Harris Corp.
P.O. Box 9100 (MS: 5-5872)
Melbourne, Fla. 32902
Voice: 407-729-7885
Fax: 407-729-7325
E-mail: dwood@harris.com

Reference

1. Boehm, B. *Software Engineering Economics*, Prentice-Hall, 1981.