

 You can download a Costar 7.0 demo now 

Overview of COCOMO

The COCOMO cost estimation model is used by thousands of software project managers, and is based on a study of hundreds of software projects. Unlike other cost estimation models, COCOMO is an open model, so all of the details are published, including:

- The underlying cost estimation equations
- Every assumption made in the model (e.g. "the project will enjoy good management")
- Every definition (e.g. the precise definition of the Product Design phase of a project)
- The costs included in an estimate are explicitly stated (e.g. project managers are included, secretaries aren't)

Because COCOMO is well defined, and because it doesn't rely upon proprietary estimation algorithms, Costar offers these advantages to its users:

- COCOMO estimates are more objective and repeatable than estimates made by methods relying on proprietary models
- COCOMO can be calibrated to reflect your software development environment, and to produce more accurate estimates

Costar is a faithful implementation of the COCOMO model that is easy to use on small projects, and yet powerful enough to plan and control large projects.

Typically, you'll start with only a rough description of the software system that you'll be developing, and you'll use Costar to give you early estimates about the proper schedule and staffing levels. As you refine your knowledge of the problem, and as you design more of the system, you can use Costar to produce more and more refined estimates.

Costar allows you to define a software structure to meet your needs. Your initial estimate might be made on the basis of a system containing 3,000 lines of code. Your second estimate might be more refined so that you now understand that your

system will consist of two subsystems (and you'll have a more accurate idea about how many lines of code will be in each of the subsystems). Your next estimate will continue the process -- you can use Costar to define the components of each subsystem. Costar permits you to continue this process until you arrive at the level of detail that suits your needs.

One word of warning: It is so easy to use Costar to make software cost estimates, that it's possible to misuse it -- every Costar user should spend the time to learn the underlying COCOMO assumptions and definitions from Software Engineering Economics and Software Cost Estimation with COCOMO II.

Introduction to the COCOMO Model

The most fundamental calculation in the COCOMO model is the use of the Effort Equation to estimate the number of Person-Months required to develop a project. Most of the other COCOMO results, including the estimates for Requirements and Maintenance, are derived from this quantity.

Source Lines of Code

The COCOMO calculations are based on your estimates of a project's size in Source Lines of Code (SLOC). SLOC is defined such that:

- Only Source lines that are DELIVERED as part of the product are included -- test drivers and other support software is excluded
- SOURCE lines are created by the project staff -- code created by applications generators is excluded
- One SLOC is one logical line of code
- Declarations are counted as SLOC
- Comments are not counted as SLOC

The original COCOMO 81 model was defined in terms of Delivered Source Instructions, which are very similar to SLOC. The major difference between DSIs and SLOC is that a single Source Line of Code may be several physical lines. For example, an "if-then-else" statement would be counted as one SLOC, but might be counted as several DSIs.

The Scale Drivers

In the COCOMO II model, some of the most important factors contributing to a project's duration and cost are the Scale Drivers. You set each Scale Driver to describe your project; these Scale Drivers determine the exponent used in the Effort Equation.

The 5 Scale Drivers are:

- Precedenedness
- Development Flexibility
- Architecture / Risk Resolution
- Team Cohesion
- Process Maturity

Note that the Scale Drivers have replaced the Development Mode of COCOMO 81. The first two Scale Drivers, Precedenedness and Development Flexibility actually describe much the same influences that the original Development Mode did.

Cost Drivers

COCOMO II has 17 cost drivers ♦ you assess your project, development environment, and team to set each cost driver. The cost drivers are multiplicative factors that determine the effort required to complete your software project. For example, if your project will develop software that controls an airplane's flight, you would set the Required Software Reliability (RELY) cost driver to Very High. That rating corresponds to an effort multiplier of 1.26, meaning that your project will require 26% more effort than a typical software project.

[Click here](#) to see which Cost Drivers are in which Costar models.

COCOMO II defines each of the cost drivers, and the Effort Multiplier associated with each rating. Check the Costar help for details about the definitions and how to set the cost drivers.

COCOMO II Effort Equation

The COCOMO II model makes its estimates of required effort (measured in Person-Months ♦ PM) based primarily on your estimate of the software project's size (as measured in thousands of SLOC, KSLOC)):

$$\text{Effort} = 2.94 * \text{EAF} * (\text{KSLOC})^E$$

Where

EAF Is the Effort Adjustment Factor derived from the Cost Drivers

E Is an exponent derived from the five Scale Drivers

As an example, a project with all Nominal Cost Drivers and Scale Drivers would have an EAF of 1.00 and exponent, E, of 1.0997. Assuming that the project is projected to consist of 8,000 source lines of code, COCOMO II estimates that 28.9 Person-Months of effort is required to complete it:

$$\text{Effort} = 2.94 * (1.0) * (8)^{1.0997} = 28.9 \text{ Person-Months}$$

Effort Adjustment Factor

The Effort Adjustment Factor in the effort equation is simply the product of the effort multipliers corresponding to each of the cost drivers for your project.

For example, if your project is rated Very High for Complexity (effort multiplier of 1.34), and Low for Language & Tools Experience (effort multiplier of 1.09), and all of the other cost drivers are rated to be Nominal (effort multiplier of 1.00), the EAF is the product of 1.34 and 1.09.

$$\text{Effort Adjustment Factor} = \text{EAF} = 1.34 * 1.09 = 1.46$$

$$\text{Effort} = 2.94 * (1.46) * (8)^{1.0997} = 42.3 \text{ Person-Months}$$

COCOMO II Schedule Equation

The COCOMO II schedule equation predicts the number of months required to complete your software project. The duration of a project is based on the effort predicted by the effort equation:

$$\text{Duration} = 3.67 * (\text{Effort})^{\text{SE}}$$

Where

Effort Is the effort from the COCOMO II effort equation

SE Is the schedule equation exponent derived from the five Scale Drivers

Continuing the example, and substituting the exponent of 0.3179 that is calculated from the scale drivers, yields an estimate of just over a year, and an average staffing of between 3 and 4 people:

$$\text{Duration} = 3.67 * (42.3)^{0.3179} = 12.1 \text{ months}$$

$$\text{Average staffing} = (42.3 \text{ Person-Months}) / (12.1 \text{ Months}) = 3.5 \text{ people}$$

The SCED Cost Driver

The COCOMO cost driver for Required Development Schedule (SCED) is unique, and requires a special explanation.

The SCED cost driver is used to account for the observation that a project developed on an accelerated schedule will require more effort than a project developed on its optimum schedule. A SCED rating of Very Low corresponds to an Effort Multiplier of 1.43 (in the COCOMO II.2000 model) and means that you intend to finish your project in 75% of the optimum schedule (as determined by a previous COCOMO estimate). Continuing the example used earlier, but assuming that SCED has a rating of Very Low, COCOMO produces these estimates:

$$\text{Duration} = 75\% * 12.1 \text{ Months} = 9.1 \text{ Months}$$

$$\text{Effort Adjustment Factor} = \text{EAF} = 1.34 * 1.09 * 1.43 = 2.09$$

$$\text{Effort} = 2.94 * (2.09) * (8)^{1.0997} = 60.4 \text{ Person-Months}$$

$$\text{Average staffing} = (60.4 \text{ Person-Months}) / (9.1 \text{ Months}) = 6.7 \text{ people}$$

Notice that the calculation of duration isn't based directly on the effort (number of Person-Months) ↘ instead it's based on the schedule that would have been required for the project assuming it had been developed on the nominal schedule. Remember that the SCED cost driver means "accelerated from the nominal schedule".

The Costar command Constraints | Constrain Project displays a dialog box that lets you trade off duration vs. effort (SCED is set for you automatically). You can use the dialog box to constrain your project to have a fixed duration, or a fixed cost.

 You can download a Costar 7.0 demo now 

[Home](#) | [Top of Page](#) | [Training](#) | [Great Links](#) | [FAQ](#)

Copyright ♦ 1986-2005 Softstar Systems All rights reserved.

Revised: April 18, 2005