Name: Gia Dao.

Student ID: 1001747062.

Class: CSE-3310-001-Fundamentals of Software Engineering.

HONORS CREDIT REPORT

-COCOMO Model-

Summary:

COCOMO (Constructive Cost Model) is a regression model based on SLOC, i.e., number of Source Lines of Code. It’s a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time, and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models.

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, SystemStar 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

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.

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 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:

* Precedentedness
* Development Flexibility
* Architecture / Risk Resolution
* Team Cohesion
* Process Maturity

### 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.

### 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)):  
  Effort = 2.94 \* EAF \* (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:  
  
    Effort = 2.94 \* (1.0) \* (8)1.0997 = 28.9 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.  
  
    Effort Adjustment Factor = EAF = 1.34 \* 1.09 = 1.46  
  
    Effort = 2.94 \* (1.46) \* (8)1.0997 = 42.3 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:  
  
    Duration = 3.67 \* (Effort)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:  
  
    Duration = 3.67 \* (42.3)0.3179 = 12.1 months  
  
    Average staffing = (42.3 Person-Months) / (12.1 Months) = 3.5 people

### The SCED Cost Driver

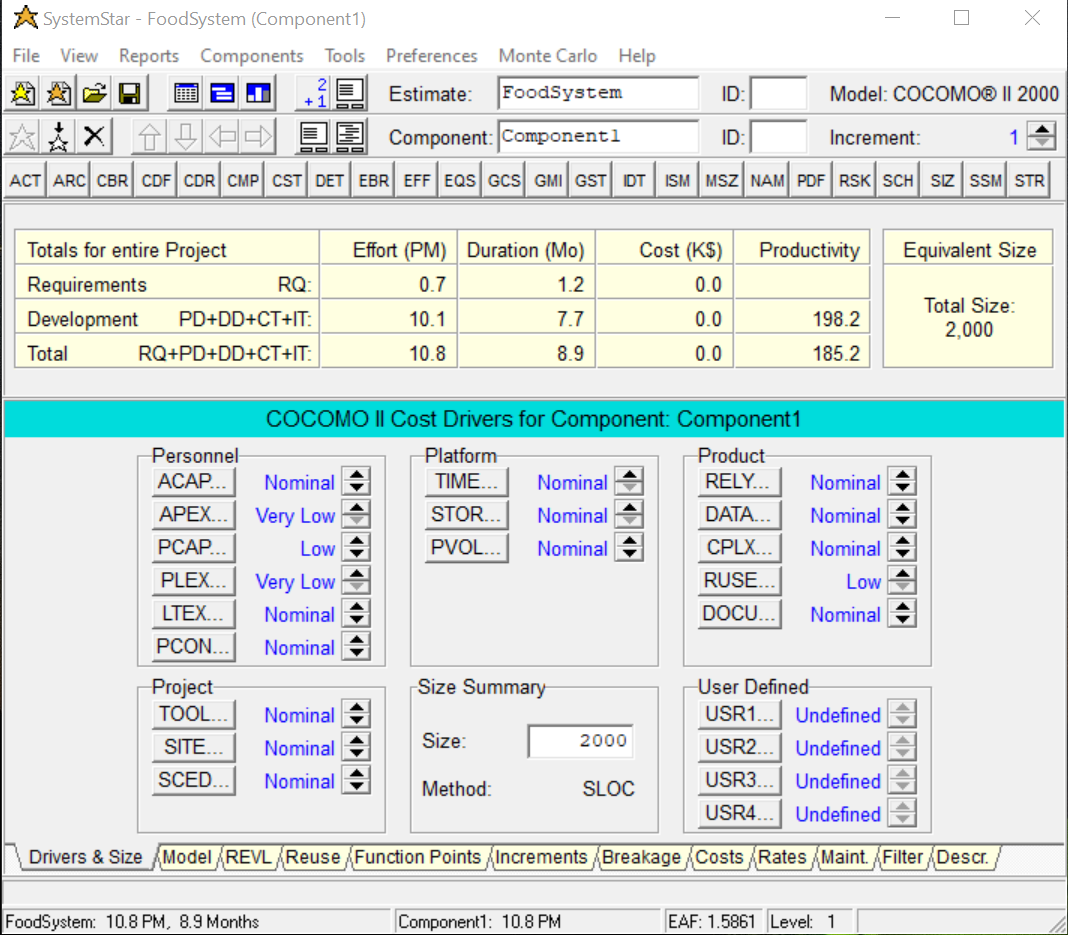
The COCOMO cost driver for Required Development Schedule (SCED) is unique, and requires a special explanation.  
  
The SCED cost driver is used to account for the observation that a project developed on an accelerated schedule will require more effort than a project developed on its optimum schedule. 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:  
  
    Duration = 75% \* 12.1 Months = 9.1 Months  
  
    Effort Adjustment Factor = EAF = 1.34 \* 1.09 \* 1.43 = 2.09  
  
    Effort = 2.94 \* (2.09) \* (8)1.0997= 60.4 Person-Months  
    Average staffing = (60.4 Person-Months) / (9.1 Months) = 6.7 people

Use COCOMO for Software Cost Estimation in Term Project “University Food System”:

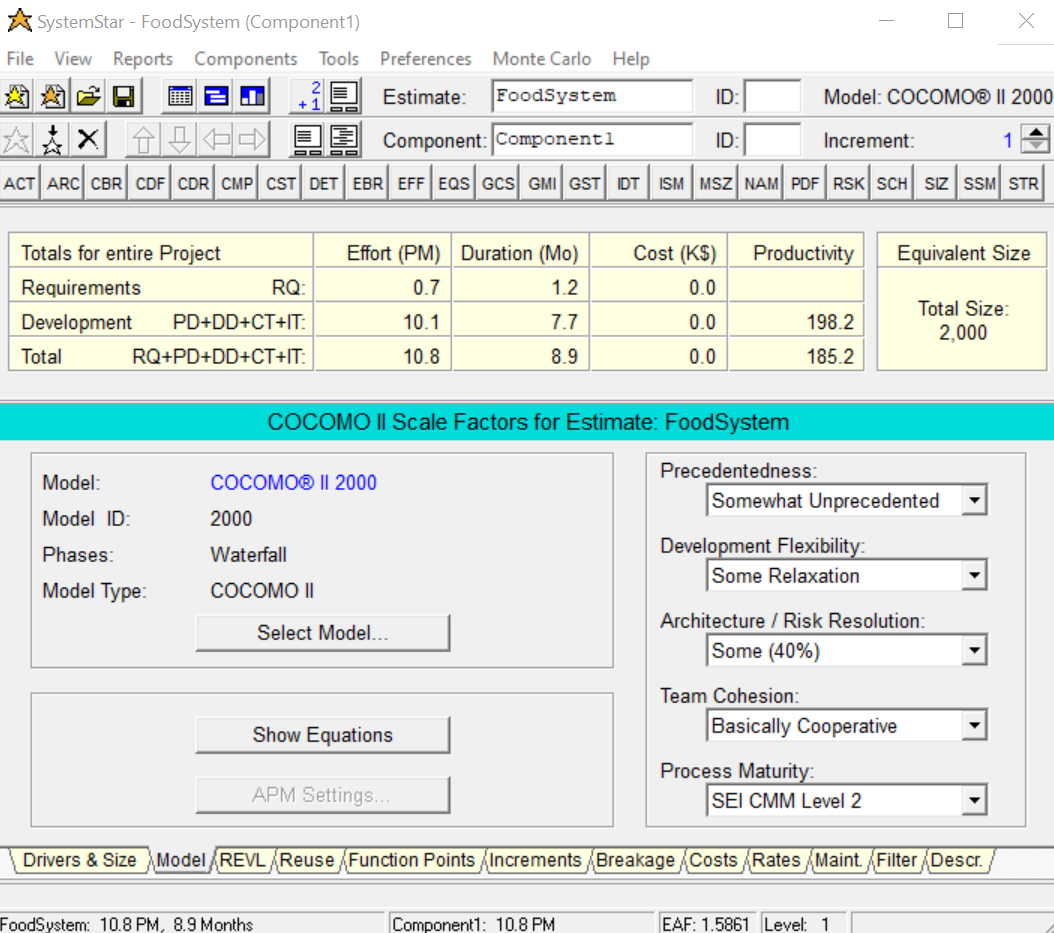
In term of the project, we have 4 members working together to develop the application. From the 17 assets of Cost Drivers, we had the following aspects:

* Considering Personnel, all members had no or low-experience with Android Studio, therefore, Programmer Capability Cost Driver and Platform Experience is set to Low. However, since we used Java to write the source code for our application and all members had some experienced with Object-oriented programming languages, so Language and Tool experience is set to Nominal
* Considering Product, we used Firebase to store all data that required in our application. This is a relatively small application and used within the university area, therefore the Database Size Cost Driver is set to Nominal. The Product Complexity is also set to Nominal
* Considering Size Summary, the source code includes approximately 20 Java files, each consists of nearly 80 lines of code on average, therefore, the source lines of code are 1,600. Also, includes the XML files to set up the GUI, the total is approximately 2,000.
* For this project, we used Conventional, or Heavyweight methodology to develop the application and all steps are calculated using COCOMO estimation.
* The project does not cost any fees for outside sources, materials, or any bills.

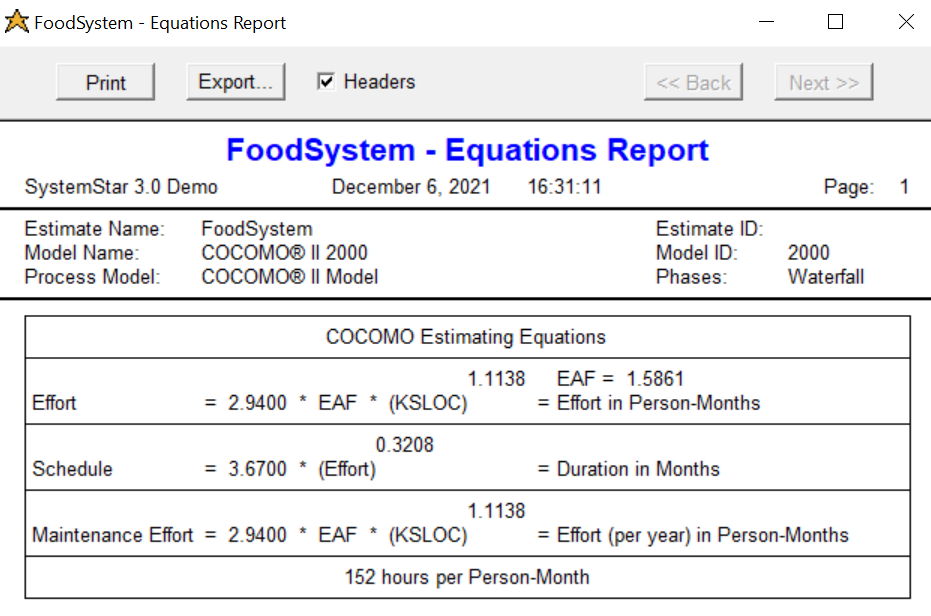
COCOMO Software Cost Estimation Report.



This is the main screen that shows the Cost Drivers set up for the project.



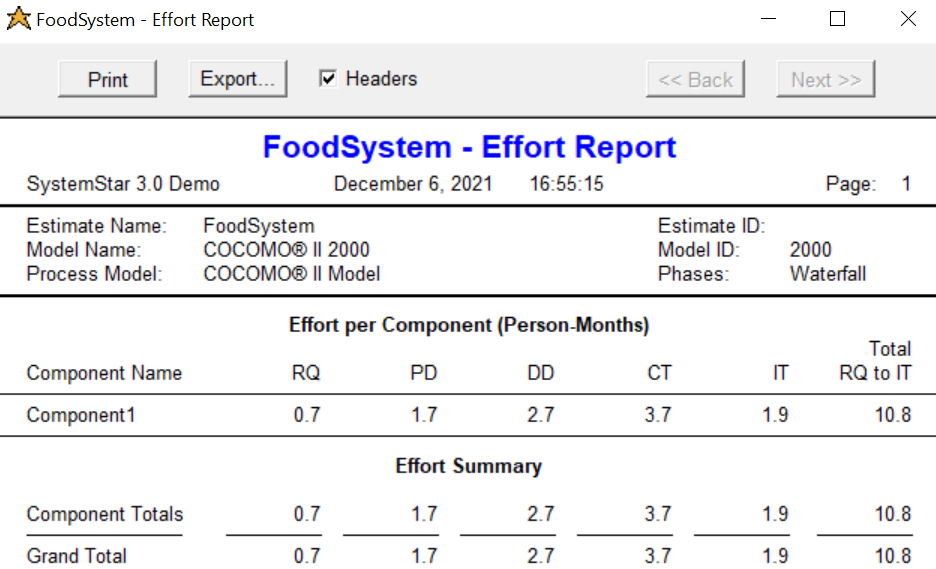
This is the main screen for Scale Drivers, including 5 factors that are set up for the project. Based on the group experience and cooperation, all factors are selected as the picture above.

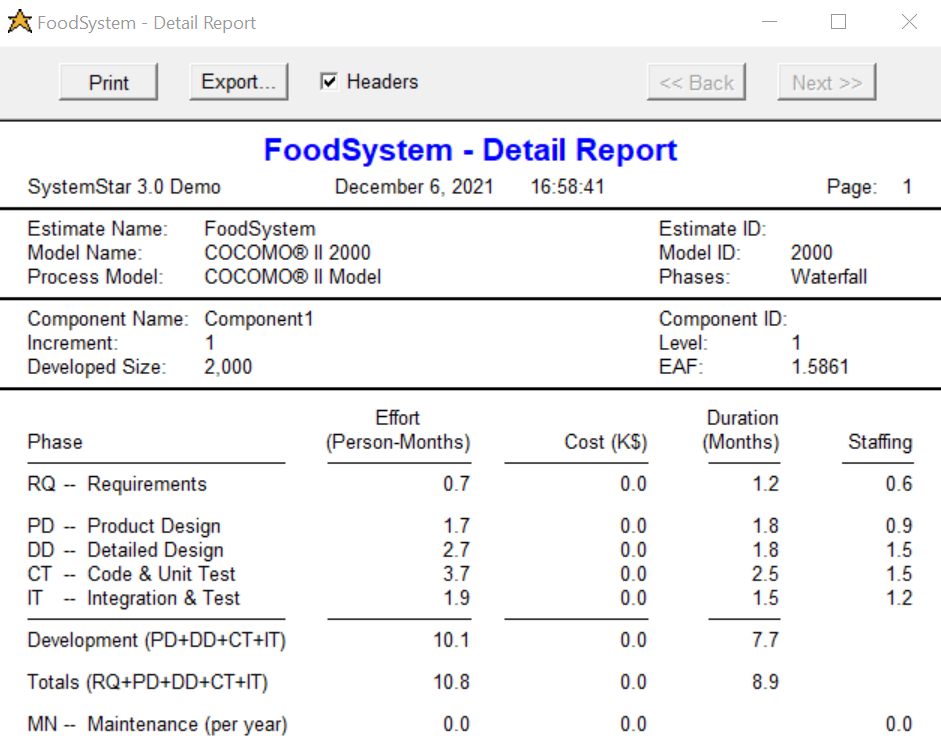


This is all the fundamental equations that needed to calculate the effort and duration. Here, KSLOC is 2, due to the fact that the total lines of source code are approximately 2,000. We have the following equations:

Effort = 2.9400 \* 1.5861 \* [2 ^ 1.1138] = 10.091 ~ 10.1 Person-Months, which correctly matched with the Development Effort displayed

Schedule = 3.6700 \* [10.1 ^ 0.3208] = 7.706, which matched with the Duration Development displayed in the image.





Conclusion:

Since this is the first time our group implemented this project, therefore, the estimation for effort and duration took longer than we expect to make the application fully satisfy all requirements. However, based on the unprecedent and the experience of each member of the group, this is a reasonable calculation.

# References

(n.d.). Retrieved from Softstar Systems: http://www.softstarsystems.com/index.html

GeeksforGeeks. (2020, June 8). *Software Engineering | COCOMO Model*. Retrieved from https://www.geeksforgeeks.org/software-engineering-cocomo-model/