EMSS Week 10

ESTIMATING EFFORT

Amol Shandilya, Minghui Jin, Yanyan Jiang

The Booch Group

1. COCOMO

COCOMO stands for constructive cost model, an algorithmic, empirical model based on project experience. It uses regression analysis technique, a technique can be used to find relationships between variables for predicting future values, to develop equations that best describe the data.

COCOMO 2 is an improved constructive cost model of the original one. It contains 4 sub-models, each one fits to the estimation of different types and stages of software development. As this model is an advanced one, so the discussion mainly focuses on COCOMO 2.

The effort estimating model is PM = A*SIZE^B*M, where A is an organization-dependent constant, B is the relative economies/ diseconomies of scale, M is project attributes and SIZE accounts for the size of project code which means the LOC, uniting by kLoc.

A is generated by analyzing previous projects, which indicated that the COCOMO 2 needs abundant data and information or very experienced experts to estimate its characters. B reflects the disproportionate effort for large projects, it can be adjusted according to the previous or the being designed project scale. So, this model can be used in the real-time system such as the EMSS solar power system, the observing system of the house devices.

Also, in the estimation by COCOMO 2, the calculation considers the influence of development environment such as the solar energy availability, which would add more accuracy to the estimates.

In the estimation of calendar time, it uses the formula of TDEV, reflecting the rate of progress, that can be used to monitor the progress of the whole progress, which is really helpful and cannot be implemented by other estimating methods.

However, even though the model has above pros, it contains some disadvantages. As it relies too much on the historical data, if there is not enough data or experienced managers and developers, the estimates may be far from accuracy. What’s more, this model is size dependent. But it is impossible to estimate the actual LOC at the early stage of development, also cannot be acquired from project documentations. And LOC estimates are also inaccurate, since different development languages, different algorithms, etc., would result in huge statistic discrepancy.

2. UCP

Pros:

1. UCPs only need the number of actors and use case to assign their weighting values, and technical complexity factor, environmental factor. It is easy to estimate and can be measured very early in EMSS system life cycle to save the team a great deal of estimating time. For example, in our EMSS project, we now have 5 use cases, and each use case has certain actors and certain defined steps. According these data, the UCP can be easily calculated.

2. It can establish an average implementation time per use case point, such as when considering the use case of excess power, it contains 2 actors, one is classified by average, the other is complex, resulting UAW as 5, and 6 steps, resulting UCCW as 10. According to the metrics of TCF, ECF, finally get the UCP of this use case. Multiplied by PF, we can estimate the effort of this use case. This would be very useful in planning future schedules.
3. Based on history actual usage data that will not change much and together with experienced managers, it provides estimates closely to actual data.
4. UCP is a very pure measure of size. Good estimation approaches allow us to separately estimate size from deriving duration. That means UCP will be independent of the size, skill, and experience of the team that implements EMSS system.

Cons:

1. UCP can be used only when requirements are written in the form of use cases and the estimate must be completed after all the use cases written. It will spend a lot of time and energy during early development period. If the documentation is not complete or accurate, the result will be inaccurate.
2. Use cases are large units of work to be used in planning a system. It is useful for initial estimate of whole EMSS system but they are much less useful for progressive team work assigned and carried out.
3. Dependent on goal-oriented, well-written use cases. If the rules for electricity power sell price are imprecise or the detail reflected in a use case varies tremendously by user, the resulting UCP may not be accurate.
4. Technical and environmental factors may be changeable or inaccurate, such as the actual supply of solar energy is unpredictable, and the prediction of the total electricity consumption usage is inaccurate. Additionally, some factors may be less useful or even useless for the project. However, these factors are also multiplied to UCCP when estimating the UCP, causing the influence of these environmental and technical factors being exaggerated.

UCP Estimate of Effort and Calendar Time for EMSS

Unadjusted Use Case Weight (UUCW)

Use Cases	UC Complexity
Day Time Power Supply	Simple
Sell Excess Power	Complex
Night Time Power Supply	Simple
User Away	Average
Get Usage Report	Average

Use Case Classification	Weight		Number of Use Cases	Result
Simple	5	x	2	10
Average	10	x	2	20
Complex	15	x	1	15
			Total:	45

Unadjusted Actor Weight (UAW)

Use Cases	UC Complexity
Solar Power System	Average
Electricity Provider	Average
The User	Complex
Payment System	Simple
Phone Application	Average

Actor Classification	Weight		Number of Use Cases	Result
Simple	1	x	1	1
Average	2	x	3	6
Complex	3	x	1	3
			Total:	10

Technical Complexity Factor (TCF)

Factor	Description	Weight	Rating
T1	Distributed System	2.0	○ 0 ● 1 ○ 2 ○ 3 ○ 4 ○ 5
T2	Response time/performance objectives	1.0	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ● 5
T3	End-user efficiency	1.0	○ 0 ○ 1 ○ 2 ○ 3 ● 4 ○ 5
T4	Internal processing complexity	1.0	○ 0 ○ 1 ● 2 ○ 3 ○ 4 ○ 5
T5	Code reusability	1.0	○ 0 ○ 1 ○ 2 ○ 3 ● 4 ○ 5
T6	Easy to install	0.5	○ 0 ○ 1 ○ 2 ○ 3 ● 4 ○ 5
T7	Easy to use	0.5	○ 0 ○ 1 ○ 2 ○ 3 ● 4 ○ 5
T8	Portability to other platforms	2.0	○ 0 ○ 1 ○ 2 ● 3 ○ 4 ○ 5
T9	System maintenance	1.0	○ 0 ○ 1 ○ 2 ○ 3 ● 4 ○ 5
T10	Concurrent/parallel processing	1.0	● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5
T11	Security features	1.0	○ 0 ○ 1 ● 2 ○ 3 ○ 4 ○ 5
T12	Access for third parties	1.0	○ 0 ○ 1 ○ 2 ○ 3 ● 4 ○ 5
T13	End user training	1.0	○ 0 ○ 1 ○ 2 ● 3 ○ 4 ○ 5

TF Total: 40.00

TCF = 0.6 +(TF/100)

TCF = 1.00

Environment Complexity Factor (ECF)

Factor	Description	Weight	Rating
E1	Familiarity with development process used	1.5	○ 0 ○ 1 ○ 2 ● 3 ○ 4 ○ 5
E2	Application experience	0.5	○ 0 ○ 1 ○ 2 ○ 3 ● 4 ○ 5
E3	Object-oriented experience of team	1.0	○ 0 ○ 1 ○ 2 ○ 3 ● 4 ○ 5
E4	Lead analyst capability	0.5	○ 0 ○ 1 ○ 2 ● 3 ○ 4 ○ 5
E5	Motivation of the team	1.0	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ● 5
E6	Stability of requirements	2.0	○ 0 ○ 1 ○ 2 ○ 3 ○ 4 ● 5
E7	Part-time staff	-1.0	○ 0 ○ 1 ○ 2 ● 3 ○ 4 ○ 5
E8	Difficult programming language	-1.0	○ 0 ○ 1 ○ 2 ○ 3 ● 4 ○ 5

EF Total: 20.00

ECF = 1.4 +(-0.03 x EF)

ECF = 0.80

UCP Calculation

The UCP is calculated based on the following formula:
UCP = (UUCW + UAW) x TCF x ECF

UCP = (45.00 + 10.00) x 1.00 x 0.80
UCP = 44.00

Our project contains **44.00** Use Case Points.

Calendar Time (TDEV)

TDEV = 3 x $(\text{Effort})^{1/3}$

Where, Effort Estimate is in terms of Staff Months

Staff Hours = UCP Efforts x PF

Assuming PF is 20 since the team is worked on smart home projects before

Staff Hours = 44 x 20 = 880

Staff Months = 880/150 = 5.86 Staff Months

TDEV = 3 x $(5.86)^{1/3}$

Calendar Time (TDEV) for EMSS = 5.4